A visual search task to evaluate top-down and bottom-up control of the pre-attentive stage and the ACT-R/PM vision module

Thesis submitted to the
Institute for Graduate Studies
in Social Sciences
in partial satisfaction of the requirements for the degree of

Master of Arts
in
Cognitive Science

by
Hakan ÜNLÜ

Boğaziçi University
2006

The thesis of Hakan Ünlü
is approved by

Prof. Dr. Reşit Canbeyli ____________________________

Prof. Dr. A.C. Cem Say ____________________________

Assoc. Prof. Dr. Ali Tekcan ____________________________

Prof. Dr. Ethem Alpaydın ____________________________

Prof. Dr. Güven Güzeldere ____________________________

June 2006

ACKNOWLEDGEMENTS

I first would like to thank my thesis supervisors, Prof. Dr. Reşit Canbeyli and Prof. Dr. Cem Say, for their support during the preparation of this thesis. I also would like to thank Assoc. Prof. Dr. Ali Tekcan for his support throughout the project. Their guidance and comments improved the scope and quality of this thesis.

I would like to thank my advisor Prof. Dr. Sumru Özsoy for the support she had provided during my studies in the Cognitive Science Department. I also thank all faculty members contributing to the program. They made my four years in this program an exciting intellectual journey.

Many thanks to my dear friend Prof. Dr. Ethem Alpaydın, for telling me about this program in the first place and encouraging me to join.

During these four years, I always had continuous support from my business partners Nur Gökman, Dilek Ovacık and Özkan Dülger. I specially thank Nur, because her vision made Hitit a reality and I would not be able to complete this demanding project without the resources and flexibility of Hitit has provided. I also would like to thank everybody in Hitit for all their help during the preparation of this thesis. I know it was a boring experiment. Thank you for your patience.

I am always grateful to my mother, father and sister for their love and support throughout my whole life. But I would like to thank my father and my uncle specially for teaching me the value and joy of science in my childhood. I wish my father could have seen this late addition to our family library.

And finally, thank you Emel for all the love and happiness you brought into my life. Many thanks for the cake and the bike as well. It was the nicest reward I had in my life for being a hard-working ☺ student.

ABSTRACT

A visual search task to evaluate
top-down and bottom up control of pre-attentive stage
and the ACT-R/PM vision module

by

Hakan Ünlü

Visual Attention is deployed in two stages: The pre-attentive stage determines which areas of the visual field are relevant for the task and therefore need to be attended. The attentive stage processes the visual information available at the attended portion of the visual field. Two rival views suggest that the pre-attentive stage is controlled by physical properties of the visual field (bottom-up) or the goals and intentions of the observer (top-down). In support of the bottom-up approach, Theeuwes conducted an experiment to show an irrelevant singleton cannot be masked in a top-down fashion. However Bacon and Egeth (1994) suggested that the nature of the task dictates which method will be used. In this study, three experiments were conducted to test Theeuwes' Irrelevant Singleton hypothesis and Bacon and Egeth's Feature Search hypothesis. The results were not compatible with either claim. The experiment results are further analyzed. Data indicate that, (i) search times depend on the color, location, set size and the form (ii) the time spent per item is larger when there is no target in the display; (iii) in the presence of a target, the average search time per item is inversely proportional to the set size. Several possible explanations are discussed.

ACT-R/PM is a cognitive architecture that allows a cognitive task to be modeled in computer environment. One of our experiment setups was modeled in

ACT-R/PM to verify that ACT-R/PM can model our task. The results show that, when default parameters are used, ACT-R/PM is slower than human participants. Also, ACT-R models fail to show the inverse relation between average response time per item and the set size. These results were evaluated and a criticism of current ACT-R/PM constructs was provided.

KISA ÖZET

Dikkat-öncesi aşamada aşağıdan-yukarı ve yukarıdan-aşağı kontrolün etkileri ve
ACT-R/PM görsel modülünü sınamak için bir görsel görev

Hakan Ünlü tarafından

Görsel dikkat iki aşamada gerçekleşir: Dikkat-öncesi (pre-attentive) aşama, görsel algı alanında hangi bölgelerin o andaki görev için önemli olduğunu ve dolayısı ile dikkat edilmesi gerektiğini belirler. Daha sonraki dikkat (attentive) aşaması, o anda dikkat edilen bölgedeki öğeleri işler. Bu konuda yaygın olarak kabul gören iki görüş vardır. Bunlardan ilki (aşağıdan-yukarı (bottom-up)) dikkat öncesi aşamanın görüş alanının fisiksel özelliklerinden kaynaklandığını savunurken diğer görüş (yukarıdan-aşağı (top-down)), bakan kişinin amaç ve niyetinin bu konuda belirleyici olduğunu savunur. Aşağıdan-yukarı fikrini desteklemek için, Theeuwes bir deney yaparak eldeki görev ile ilgisiz bir tek öğenin (singleton) yoksayılamayacağını göstermiştir. Buna karşın Bacon ve Egeth (1994) eldeki görevin kullanılan stratejiyi belirlediğini öne sürmüştür. Bu çalışmada, Theeuwes'in İlgisiz Tekil Öğe fikri ile Bacon ve Egeth'in Özellik Arama hipotezini sınamak için üç deney yapılmıştır. Sonuçlar (i) görsel arama sürelerinin renk, yer, öğe sayısı ve şekli ile ilişkili olduğunu (ii) aranan hedef şeklin olmadığı durumlarda, öğe başına harcanan sürenin daha az olduğunu, (iii) hedef öğenin olduğu durumda öğe başına arama zamanının, öğe sayısı ile ters orantılı olduğunu göstermiştir. Pekçok alternative açıklama incelenmiştir.

ACT-R/PM, bilişsel görevlerin bilgisayar ortamında modellenmesini sağlayan bilişsel mimarilerden biridir. Deney kurulumlarımızdan biri ACT-R/PM ortamında modellenerek, ACT-R/PM'in deneyde kullanılan görevi modelleyip modelleyemeyeceği sınanmıştır. Sonuçlar göstermektedir ki, mevcut parametreleri ile ACT-R/PM bu görevi insanlardan daha yavaş yapmaktadır. Bunun yanında, ACT-R/PM öğe sayısı ile öğe başına arama zamanı arasındaki ters orantıyı da modellemekte başarısız olmuştur. Sonuçlar değerlendirilerek ACT-R/PM yapısının bir eleştirisi sunulmuştur.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1.

INTRODUCTION

We receive information about our surroundings through various sense modalities. We hear the sounds of music, smell flowers and feel the softness of a cloth. Vision is the most dominant of all senses. More brain area is devoted to vision than any of the other senses. If visual information conflicts with the data from other senses, we usually trust our vision. Using vision, we recognize the objects and forms around us (Reisberg, 2001).

Visual search is a cognitive task that also plays an important role in our everyday life. In broad terms visual search can be defined as locating and identifying a target item surrounded by distractor items. We start the day by trying to locate toothpaste in a crowded bathroom, sugar on the breakfast table and the weather report on the morning newspaper. Being such an important cognitive faculty, visual search is one of the main research topics in cognitive psychology.

Visual search involves deployment of visual attention to various parts of the visual field and looking for the target item at the location which in the current focus of attention.

Knowing more about visual attention and visual search, we can apply this knowledge to build better appliances and applications. However, this requires a solid theory of visual cognition, which enables us to make predictions on what might be a typical human behavior in certain contexts. When supported with theories about other cognitive processes, we can build a complete theory of cognition that can answer many questions about human cognitive processes. In his landmark work, Newell claimed that, such cognitive theories exist. Since then many researchers have been trying to build their own theories with the hope that it would be a more complete theory of human cognition. Newell himself developed Soar, Anderson developed ACT-R, Meyer and Kieras have developed EPIC. These theories are always supported by computer simulations that allow models of various cognitive processes to be built and observed.

Using a cognitive architecture like ACT-R, it is possible to build a model of a visual task such as visual search. For example, currently there are many research projects that center around building computer models of visual search, especially for modeling human interaction with computers. These projects usually make use of a cognitive architecture like EPIC or ACT-R and they try to model perceptual-motor tasks like searching items in a computer display and using keyboard and mouse to provide appropriate responses. The Human-Computer Interface tasks are naturally the first tasks to be modeled because the user interaction with the computers can easily be simulated by a program whereas a human being's interaction with the real world is too complex phenomenon to be captured in a computer model.

As the present research is related to visual search and cognitive modeling in ACT-R, in the next sections, we will summarize the current research in visual search, cognitive modeling and finally present our research project and goals.

Visual Search and Visual Attention

The amount of information transferred from the retina to the brain is estimated to be in the range of 10^8-10^9 bits per second and by far exceeds what the brain is capable of fully processing and assimilating into conscious experience. Because in spite of the parallel architecture of the brain, it appears that the brain employs a serial processing strategy (Deco et al, 2002). Therefore the processed information can only be a small portion of the available visual data which makes it necessary to select portions of visual input and ignore the rest. The postulated mechanism for selecting the subset of available visual data for further processing is visual attention. The concept of attention implies that the focus of attention will be deployed to the different parts of the visual field under the control of some sort of cognitive or physiological process. As a result of this, a part of the visual field is selected and attention is deployed to that field. This mechanism is commonly known as selective or focal attention (Broadbent, 1958; Kehneman, 1973; Neisser, 1967).

In order to explain how attention works, Helmholtz (1867) had introduced a metaphor in terms of a spotlight. Since then the spotlight metaphor is commonly used to describe the attentional process. (Crick, 1984; Treisman, 1982) This metaphor describes an imaginary spotlight of attention illuminating a portion of the visual field and only this illuminated portion is available for higher cognitive processing. The rest of the display is filtered out. If we want to see a part of the visual field, the spotlight should be moved that part of the display. Sperling and Weichelgartner (1995) demonstrated that, the spotlight's movement is discrete rather than continuous across the visual field. While it is fading away at one part of the display, it is increasing in strength at some place else. The movement of the attention can be done by eye movements (overt attention) or by focusing to a peripheral part of the visual field without moving the eyes (covert attention).

The original spotlight idea implies some sort of serial processing of the visual field and this is a widely accepted view. However, the mechanisms that decide where and when the spotlight will be directed to should also be explained. In order to accommodate the idea of a guided attention, rather than a random scan of the visual field, it is also postulated that visual attention has two stages. A *pre-attentive* stage comprises processes that are fast, parallel and involuntary, acting on the whole visual field. This stage is followed by an *attentive* stage in which the focus of attention is moved to the locations selected in the pre-attentive part. These pre-selected locations are then attended in a serial manner and there is a slower, serial and voluntary process of encoding and understanding the information at the attended location (Shaw 1978). The parallel pre-attentive processes guide the spotlight and the spotlight moves under the control of the pre-attentive processes.

Having a pre-attentive stage automatically calls for an explanation of the processes that take place at that stage. The fundamental question here is to determine if the cognitive processes that drive the pre-attentive stage of visual search tasks are controlled by the properties of the image or by the intentions or goals of the observer. Two different paradigms try to answer this question. The “top-down” or “goal-driven” view argues that the conscious goals of the observer take precedence over the physical properties of the image. The physical data collected at the pre-attentive level is processed based on the goals of the observer and the attention is driven as a result of this processing. On the other hand the “bottom-up” or “stimulus-driven” view argues that the specific physical properties of the image drive the visual search task independent of the observer’s conscious intentions. The observer has no control over the pre-attentive stage.

In most studies these two views are contrasted but it is also argued that they are two extreme ends of the spectrum and in most visual tasks, the observers’ performance is a combination of goal-driven and stimulus-driven approaches. Evidence indicates that both forms of processes occur in all stages of the visual search. The early stages are dominated by the stimulus-driven control, whereas the goal-driven control takes precedence in the later stages (van Zoest et.al. 2004).

The Feature Integration Theory (FIT) (Treisman and Gelade, 1980; Treisman and Sato, 1990) aims to explain what happens at the pre-attentive stage. According to FIT, in the first step of visual processing, several primary visual features are processed and represented with separate feature maps. A feature map is a matrix like representation of the visual field, identifying where certain features reside. A different feature map is formed for each different feature like form, color etc (Figure 1). For example a feature map for the color red is a matrix representation of the visual field where the red locations have some sort of value and other locations are empty. These maps are later integrated in saliency maps or priority maps. The aim of a priority map is to represent topographically the relevant parts of the visual field to guide the attention to these parts of the visual field where the likelihood of locating the target is higher (Deco 2002). So a priority map is in fact a deployment plan for attention. It is argued that the priority maps are formed using a stimulus-driven control (Koch and Ulman 1985) or goal-driven control (Van Dee Laar et. al. 1997).

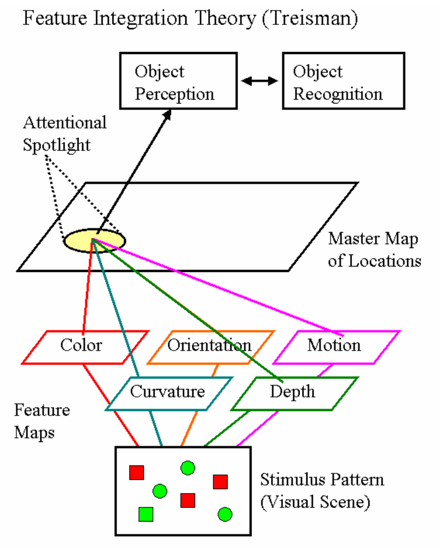


Figure 1. Treisman’s Feature Integration Theory. The feature maps for various features like color, orientation, motion, curvature, depth are automatically constructed. They come together to determine where the attention will be directed.

This theory is capable of explaining results of various experiments and also suggesting mechanisms for the binding problem. The binding problem is the question of mechanisms involved in the fusion of features that compose an object such as color, form and motion.

According to FIT, visual data are first processed in parallel across the complete visual field extracting single primitive features without integrating them. This is followed by a search of the items selected in the first pass in a serial manner. Therefore, FIT defines two kinds of visual search tasks, feature search and conjunction search. Feature search can be performed fast and pre-attentively for targets defined by primitive features. Conjunction search is the serial search for targets defined by a conjunction of primitive features. It is much slower and requires conscious attention. Treisman concluded from many experiments that color, orientation, and intensity are primitive features, for which feature search can be performed.

However, there is evidence that feature-search is not the only mechanism used in pre-attentive visual search. Pashler (1988) argued that subjects may employ two different strategies based on the nature of the task. They may monitor a specific feature and employ feature search, or look for an object that stands out. Theeuwes (1992) suggested that, the attention is first deployed to the salient items in the display, regardless of the observer's intentions and the relevance of the salient item for the task. Although a quantitative description of a salient item is not given, it can be vaguely defined as "an item that stands out in a display because it is different from the other items in one or more features".

Bacon and Egeth (1994) rejected this view. They suggested that the pre-attentive stage is always under conscious control. Based on the observer's understanding of the task, the observer can use ‘singleton-detection mode’ which favors for salient items. In the singleton-detection mode, subjects have a top-down predisposition to look for singletons in the display. However, the observer is able to override singleton-detection mode and concentrate on the relevant features, by employing ‘feature-search mode’. Theeuwes and Burger (1998) suggested that, it is only possible to ignore salient items if the features of the target and the salient item are known.

The following section gives a brief summary of studies by Theeuwes (1992), Theeuwes and Burger (1998) and Bacon and Egeth (1994) as they are closely related to the current study.

Relevant research

Theeuwes (1992): Perceptual selectivity for color and form

One of the seminal articles of the ‘bottom-up paradigm’ was written by Theeuwes in 1992. Theeuwes (1992) showed that the pre-attentive stage cannot selectively guide the attentive stage to the task relevant stimulus direction, when there are items relatively more discriminable from the others. According to Theeuwes, these findings indicate that the pre-attentive stage computes, for each stimulus dimension, the differences in features, resulting in an activation map, representing how different each item is from any other item in the display for a particular dimension like form or color. Then it is assumed that the attention is focused to the highest activation area on the map and then to the next highest and so on. This claim is different from the feature maps in FIT. A feature map assumes that a certain feature like color has been identified and a map for that particular color has been formed. However, an activation map shows only the differences. So activation maps are more ‘free’ from the meaning than feature maps, representing the stimulus in terms of some highs and lows in one physical dimension or the other.

In this experiment subjects were only told which dimension was relevant. So, subjects who were searching for a unique color received blocks of trials in which a red item was located among green non-target items or a green target item was located among red non-target items. Likewise, subjects looking for form were looking for a square among circles or a circle among squares. Therefore they did not exactly know the feature they were looking for. Only the target dimension is given to the subjects. The same holds for the distractor dimension. The subjects knew that there was another item present in the irrelevant dimension. But the exact features of this item had not been given.

Therefore, Theeuwes conducted a second experiment (Theeuwes 1992) to find out if the pre-attentive attentional capture can still be observed when the targets know the feature they are searching for (e.g. knowing the target is green). In this experiment, Theeuwes presented subjects with displays consisting colored circles or diamonds arranged in a circular layout. There were multi-element displays (Figure 2) with 5, 7 or 9 elements. In the no-distractor color condition a green circle was among red circles. In the distractor condition one of the red items was a square. In the no distractor form condition, the green circle was among green squares. In the distractor form condition one of the squares was red. In the color condition the subjects are asked to find a green item and in the form condition they are asked to find a circle.

Line segments with different orientations were presented in each item. Subjects are asked to determine the orientation of the line segment in the target item. The target item was always a green circle.

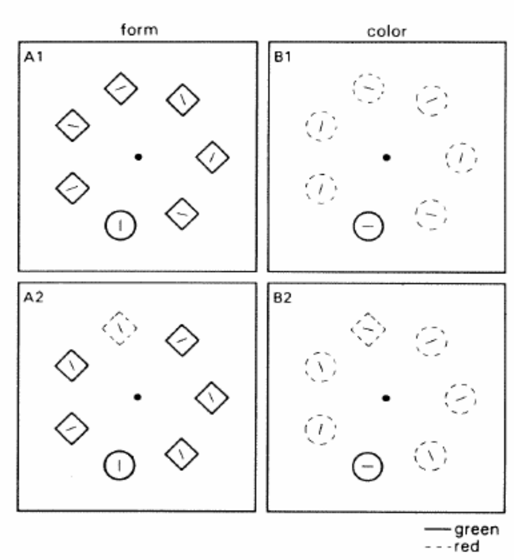


Figure 2. An example display from Theeuwes (1992) experiment.

The time to find the target increased if there was an object with a different color in the display. The color distractor singleton increased the response times when searching for form, whereas a form distractor when searching for color had no effect.

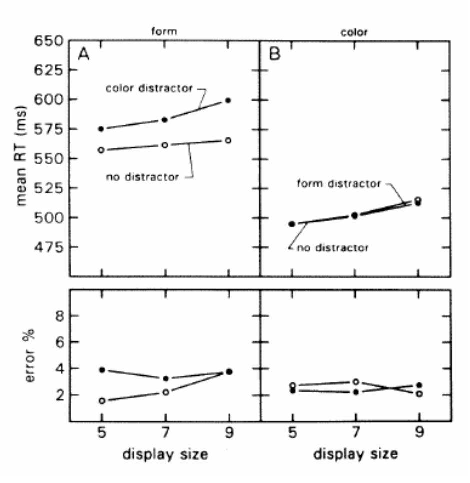


Figure 3. Experiment 1A: Mean reaction time and error percentages (percent of wrong responses given by the subjects) for search with or without a distractor for the from (A) and color (B) conditions.

Theeuwes concluded that the color becomes available earlier in time than the difference in form, suggesting in form condition attention is first captured by the uniquely colored distractor and then captured by the uniquely shaped target.

Theeuwes also tested if an increased number of trials may result in subjects inducing top-down control. Increasing the number of trials, he observed that even with extensive practice, subjects are not able to ignore an irrelevant color singleton.

In his second experiment, Theeuwes tested whether the asymmetry will reverse when the color discrimination becomes harder than the form discrimination. By using two very similar colors, he observed that this time the irrelevant form interferes with the search process (Figure 4).

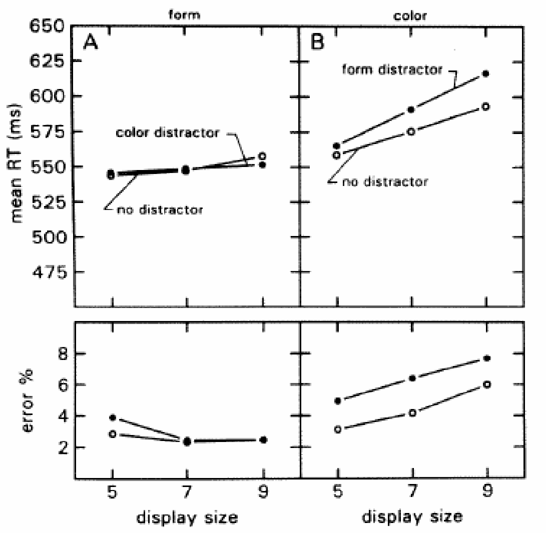


Figure 4. Experiment 2. Mean reaction time and error percentages (percent of wrong responses given by the subjects) for search with or without a distractor for the from (A) and color (B) conditions.

This is presented as supporting evidence that the attention is first captured by the most salient item. Theeuwes explains his results as follows:

"In short, the present study demonstrates the parallel [pre-attentive] stage cannot selectively guide the attentive stage to just the known-to-be-relevant target feature. Because selectivity depends on the relative discriminability of each of the dimensions, the findings can be explained by a model that assumes that the pre-attentive stage calculates automatically differences in features within stimulus dimensions, followed by an attentive stage that automatically shifts to the location of the features that pops out first." (Theeuwes, 1992, pp 605)

Theeuwes and Burger (1998) Attentional Control during Visual Search: The Effect of Irrelevant Singletons

In 1998, Theeuwes and Burger published a follow up study where using the letters of the alphabet. The task was to ignore one letter that was presented in a different color and look for the target letter among other letters (Figure 5).

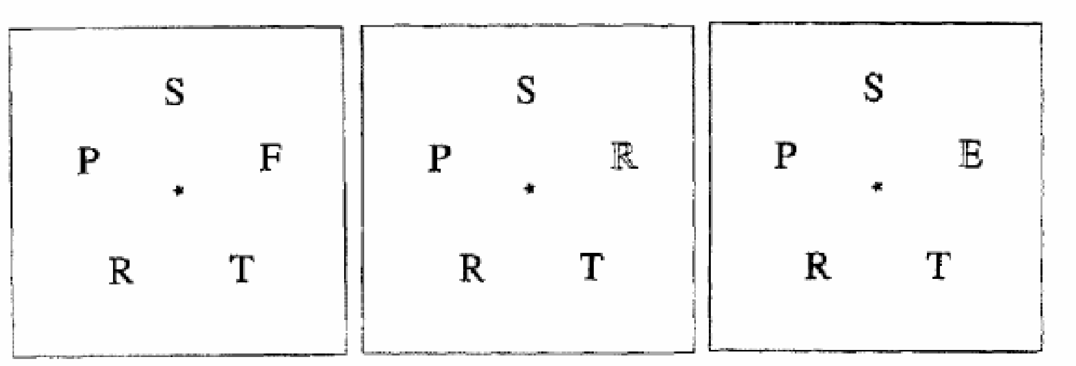
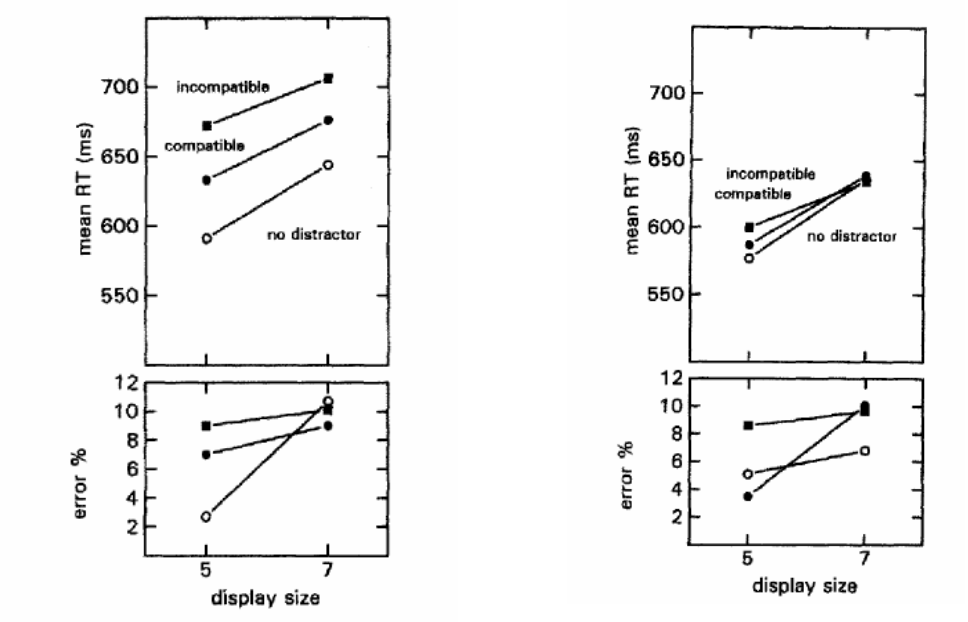


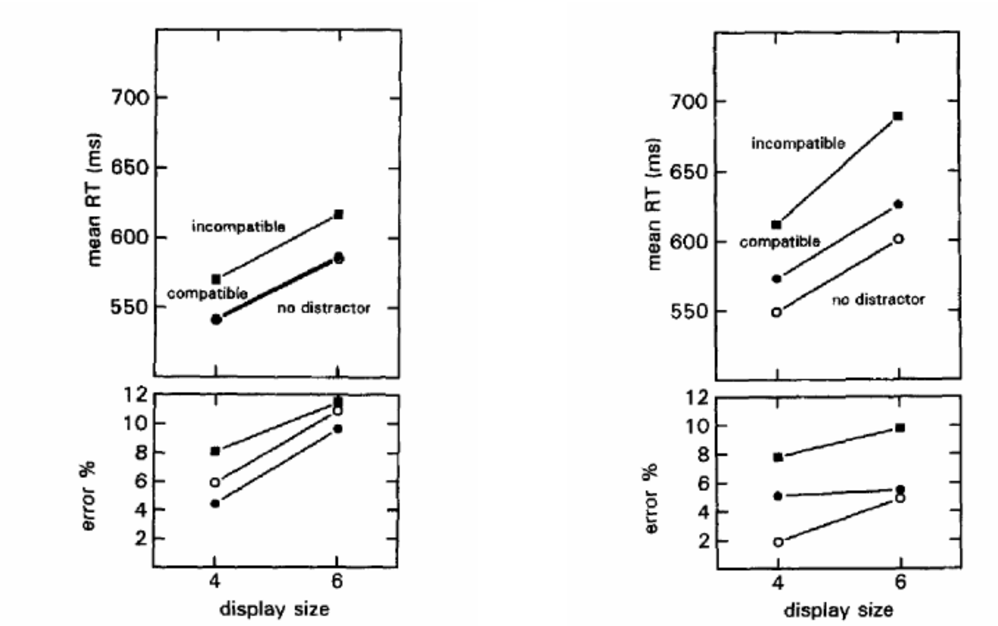
Figure 5. The task is to ignore the green singleton. In control condition (left) all letters are red. The target is ‘R’ (press right). In the compatible condition (middle) the singleton to be ignored is a green letter ‘R’ which is identical to target. In the incompatible condition (right) the singleton to be ignored is a green letter ‘E’, different from the target letter ‘R’.

In the first experiment the subjects were not told which color the singleton will appear and what color the target will be. In the second experiment, they were told the target and distractor colors which remained unchanged across the experiments. In the third experiment, the target color was fixed to gray and the distractor color was varied from one trial to the other. In the fourth experiment the non-target color was fixed to gray and the target color was varied.



Experiment 1 : Alternating target and non-target colors

Experiment 2: Subjects were told the target and non-target colors and they remain fixed over the trials.



Experiment 3: Subjects know the target color (gray) and the distractor color alternating.

Experiment 4: Subjects know the distractor color, target color was alternating.

Figure 6. The results of Theeuwes and Burger (1998)

The result of this study (Figure 6) also indicated that it is only possible to ignore an irrelevant singleton when both target and distractor features are known.

Bacon and Egeth (1994) Overriding stimulus driven attentional capture.

An objection to Theeuwes' ideas came from Bacon and Egeth. They claimed that the saliency determines attentional priority only if the subjects have already adopted a strategy that favors processing based on the saliency. If they adopt a strategy that looks for a salient item that stands out (singleton-detection mode) the singleton effects observed by Theeuwes are observed. However, if the subjects adopt a different strategy searching for a relevant feature (feature-search mode) then the bottom-up effects can be eliminated. They have claimed that, if the task involves a singleton with a known feature then the subject can use both strategies. An irrelevant singleton only affects the search if singleton-detection strategy is adopted by the subject. In order to support their hypothesis, they designed three experiments.

Their first experiment is an exact replication of the "form condition" in the first experiment done by Theeuwes (1992). The subjects were looking for a green circle among green squares. Sometimes, one of the squares is displayed in red. The presence of a red square increased the response time by 21 ms to 34 ms, which were very close to the effects reported by Theeuwes. However, Bacon and Egeth criticize this experiment by saying that the distractor factor was blocked rather than mixed. Therefore subjects were aware of whether there will be a distractor or not.

In their next experiment, Bacon and Egeth tried to overcome the singleton-detection mode by introducing more targets. In any display there were one, two or three targets. They assumed that, this will force the subjects to switch from singleton-detection mode to feature-search mode. Indeed the results indicate that the response time in these cases are affected by the number of targets but they are not affected by the presence of a distractor. It is worth noting that the displays where a single target was presented was in fact identical to the displays in the first experiment and the response times were very close to the first experiment's no-distractor case. This means that subjects are now able to mask the distractor effects.

In their third experiment, Bacon and Egeth introduced other non-target form singletons like triangles. They predicted that, as the target is no longer the only form singleton, the subjects will switch to feature-search mode. Indeed, the distractor effect of the color singleton again disappeared.

Based on these results, they concluded that if the subject is in a singleton-detection mode an irrelevant singleton would interfere with the task. However if the subject is using a feature-search strategy, then the irrelevant singletons can be ignored. This is taken as supporting evidence that even the early pre-attentive stages of visual search is under top-down control and therefore acts according to the strategies imposed on it in a top-down fashion.

Cognitive Modeling and ACT-R

One of the major goals of the research in cognitive psychology is to provide theories of cognitive processes and abstract models that represent them. Each different area of cognitive psychology has developed its own theories and models. For example, The Modal Model of human memory (Atkinson and Schifrin, 1968) gives an account of how memory works. Although the Modal Model is able to explain many things about human memory, it tells us little about how it interacts with other cognitive functions. In a sense, it is an isolated model. This is true for all theories that are developed as a result of research in specific areas of cognition. They are not able to provide a complete model that can explain the whole picture.

In his book titled Unified Theories of Cognition, Allen Newell (1990) claimed that psychology had arrived at the possibility of unifying these theories of cognition. This is a very strong statement meaning that we should be able to find a single cognitive theory that can model various cognitive functions acting together. A cognitive theory provides us with the building blocks or primitive operations of cognition. Using these, we are able to model various cognitive tasks. A model's success depends on how realistically it can model a wide range of cognitive tasks. Here being realistic is used in the sense that being close to human performance and uses for such models are limitless. In very simple terms, this will make it possible to build a 'brain simulator'. With sufficient sophistication, the model will be able to tell us what would be the typical behavior of a human subject performing a certain task like arithmetic, problem solving or visual search. Recently this approach has used to build working models that can evaluate a given computer display to find out what would be the performance of the subjects in processing it. For example, Byrne (2001) has developed a model that performs a menu selection task where the model serially searches a list of items and clicks the desired item with the mouse. With careful modeling of visual and motor tasks, Byrne successfully modeled the task with a perfect fit to the data captured from human subjects.

Following Newell's advice, various groups developed their cognitive theories which are usually accompanied with a computer modeling tool that can enable researchers to make simulations of their models in computer environment. Newell himself developed Soar (Laird et al. 1987). Among other popular systems are EPIC (Meyer and Kieras 1997), 3CAPS (Just and Carpenter, 1992) and ACT-R

(Anderson 1993). In the past decade, parallel research was conducted in all these frameworks. In this thesis we will concentrate on ACT-R and try to model a portion of our research experiment in ACT-R to see how well this cognitive architecture can model the visual search task.

ACT-R Basics

ACT-R was developed by John R. Anderson and his colleagues in Carnegie-Mellon University. ACT-R's official name is an acronym for 'the adaptive control of thought', as based upon the ACT Theory (Anderson, 1976). We have also seen it as another acronym based on the 1998 book's title, the Atomic Components of Thought - Rational.

Anderson (1998) defines ACT-R as: “ACT-R is a theory of the nature of human knowledge, a theory of how this knowledge is deployed and a theory of how this knowledge is acquired.”

In ACT-R, there are two types of knowledge or memory, *declarative* and *procedural.*

Declarative knowledge is the set of facts that we can reference with a direct recall from our memory. These are simple facts like ‘a dog is an animal’ or ‘2+3=5’ which most of us do not need any inferences to find out. We can directly recall these facts from declarative memory and use them in the thought process of the model.

Procedural knowledge is the set of things we know how to achieve; for example, how to perform addition, how to look for a certain item on a computer display etc. We can directly access the procedural knowledge, but the procedural knowledge does not give us any information by itself. It is just a description of how the declarative knowledge should be used to produce new pieces of declarative knowledge. In other words, it defines how we reach conclusions, how we solve problems or perform other tasks using the facts that we know.

It is worth mentioning that, we have described '2+3=5' as declarative knowledge and addition as procedural knowledge. The reason is, '2+3=5' is something we learn at primary school and know by heart. We do not resort to any calculation to answer this question. However in order to answer 127+456 we need to do a series of operations to find the result. We first need to add 7 and 6, care about the carry etc. It is also important to notice the fact that, declarative and procedural knowledge of each person is different. For some people '25+50=75' could be a fact known by heart, yet for other people it is an addition problem. The same is true for the models. Some models are provided with '2+3=5' as a part of their declarative knowledge, whereas other models need to do a finger calculation like a small kid to figure out '2+3=5'.

The declarative knowledge is represented in ACT-R as *chunk*s. A chunk is in fact a relation between various entities. The representation of '2+3=5' is given in Figure 7.

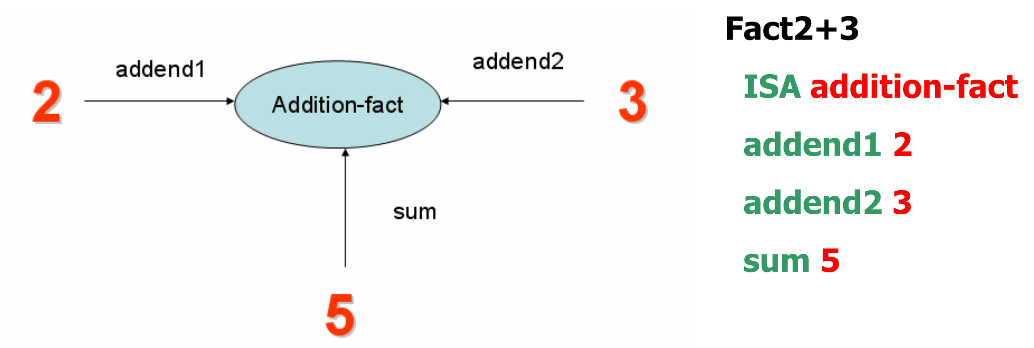


Figure 7. Visual representation of a chunk and the ACT-R code that corresponds to this chunk. In the ACT-R code, slot names are shown in green and the value in each slot is shown in red.

In this form it defines a piece of declarative knowledge or a fact. In plain words we can describe this fact as ‘It is a fact that when we add two and three, the sum is five’.

When we analyze in detail, a chunk is in fact a set of values each of which is given a special name. These are called *slots*. When we look at Figure 7, we see that there is a center node surrounded by a number of names and values. The chunk’s center node determines the type of the declarative knowledge and constitutes a special slot of each chunk: This is called the ‘*ISA*’ slot. Connected to the ISA slot, a chunk can have as many other slots as necessary. Our addition fact has four slots, the first one being an ‘ISA’ slot. Then we have three more slots, called addend1, addend2 and sum.

Procedural knowledge on the other hand, is defined as a set of conditional rules, called *production rules*. An example production rule is given in Figure 8.

ACT-R Production Rule

```
(P example-counting
  =goal>
    ISA count
    step counting
    number =num1
  =retrieval>
    isa count-order
    first =num1
    second =num2
==>
  =goal>
    number =num2
  +retrieval>
    isa count-order
    first =num2
)
```

English Description

```
If the goal is
to count and
the current step is counting and
there is a number which we will call
=num1
and a chunk has been retrieved
of type count-order
where the first number is =num1
and it is followed by another
number which we will call =num2
 Then
change the goal
to continue counting from =num2
and request a retrieval
of a count-order fact
for the number that follows =num2
```

Figure 8. An example production rule and its English description.

The English description, which is not a part of the actual model and shown here only for explaining the production rule, may seem a bit odd. But we should remember that, this is supposed to be the internal representation of procedural knowledge. It consists of two parts. The first part is called the left-hand-side, is the part before = => sign. It tells ACT-R what kind of chunks should be present in the current model so that this production rule can be used. If the model can find such pieces of declarative knowledge in its memory, then the right-hand-side, which is the part below = => is performed. This part has items prefixed with equal sign (=) or plus sign(+). The equal sign means that an existing buffer will be modified, whereas plus sign means that a new *buffer* will be added to the model. Buffers are working areas where ACT-R keeps chunks while it is running the model. We can think of them as analogous to the working memory.

So in plain English, the above production rule means: Look into the declarative memory. If the current goal is declared as counting numbers (slot ISA=count) and we are already counting (slot step=counting) and there is a current number, let us call this number as 'num1'. This means that we have a matching piece of declarative knowledge in our goal buffer and from it we learn the goal is counting, and we have already started counting and the current number. Now the second chunk tells us to go into the buffer area called retrieval and look for the chunk stored there. This time we have to find there, a count order where the first number is the number we have learned in the previous chunk. If we can find it, now we know num2, which is the number that follows our number. Then we tell the model to start counting from num2 and find another count order that give the number that follows num2. And this goes on like this, counting one by one, until the model encounters another production rule that tells the model to stop. The complete model for counting is provided in the next section.

In ACT-R, each production rule has an associated time value, which tells us how much time would elapse if this production rule is fired. Also, ACT-R production rules fire sequentially, a production rule needs to be completed for the next production rule to fire. Timings are an important component of each model. The default values that come with ACT-R software are based on data from several experiments with human subjects. Although these timings are parametric and can be changed, such a change would make ACT-R incompatible with the tasks that form the basis for this data. Therefore, this is a delicate issue that needs to be handled with care. Because, ad-hoc manipulation of the timings gives us the ability to fit a model's performance to human performance. However, the credibility of such a model is questionable. The common practice for improving our belief that the model correctly reflects human cognitive processes is to present the same task to ACT-R and human subjects and end up with similar performances, without modifying ACT-R default timings.

In addition to declarative and procedural memories, there is an additional memory called goal stack, which keeps track of what the model is trying to do. Given a goal (such as 'add 127 and 456') ACT-R tries to find the relevant production rules that may yield a valid answer. Usually there is more than one production rule that needs to be applied and various portions of declarative memory should be used. In order to satisfy a goal, usually a sub-goal should be satisfied and this may require a sub-sub goal. The goal stack keeps track of all these sub goals. At the lowest level, goals are satisfied usually by declarative knowledge or perceptual input. Then the system continues to satisfy the goals that caused this sub goal to be put on the goal stack.

A simple ACT-R Model: Counting

In this section we will examine the details of a simple ACT-R model to understand its inner working. Our model will be able to count, for the sake of simplicity, within the range of 1 to 6. This model is presented and discussed in more detail in ACT-R Tutorial Unit One, which can be downloaded from ACT-R Official Site (http://act-r.psy.cmu.edu/actr6/ has the version 6 which is newer than the version 5 used in our study).

Before we move on to the model, let us see how we would be teaching how to count to a 5 year old child. We would first teach him the sequence of numbers ("After one comes two, after two comes three"). The child has to memorize these facts or pieces of declarative knowledge. And then we would tell him what to do when someday asks him to count from a number to another number ("When somebody tells you to count from a number, start counting by telling the number that follows this number and continue telling the number that follows the last number you have told until you reach the target number"). The child has to learn this procedure well enough to be able to apply it. In building a cognitive model, we give our model each piece of declarative and procedural knowledge needed for the operation of the model, as if we are teaching something to a child.

Therefore, the first step in building a model is to give it the declarative chunks. This is the knowledge base of the model. In our counting model, we have to tell ACT-R which number follows which number. Here we have to keep one important thing in mind. ACT-R is not a computer programming language. Therefore it does not know the numbers, arithmetic or any other thing the computers are usually very good at. We have to tell it what is what starting from the very basics.

```
(b ISA count-order first 1 second 2)
(c ISA count-order first 2 second 3)
(d ISA count-order first 3 second 4)
(e ISA count-order first 4 second 5)
(f ISA count-order first 5 second 6)
(first-goal ISA count-from start 2 end 5 step start)
```

Here each line specifies one chunk. The first five define counting facts named b-f. The names are not important and there are only for the convenience of the modeler. These chunks are all of the type ‘count-order’. Each counting fact connects the number lower in the counting order (in slot ‘first’) to the number next in the counting order (in slot ‘second’). This is the knowledge that enables the system to count. The last chunk, ‘first-goal’, is of the type ‘count-from’ and it encodes the goal of counting from 2 (In slot ‘start’) to 5 (in slot ‘end’). In order to run the model, this chunk will be declared as the goal that needs to be satisfied. by the command (goal-focus first-goal). Note that the step slot of ‘first-goal’ is set to start at the beginning. It will be set to counting as the counting progresses and to stop when the counting is over. This use of a slot in the goal to maintain a current state (keeping track of what is happening now) is a common practice when writing ACT-R models. It provides a way to limit which productions are appropriate at any particular time. There are certain things we do when we start counting like setting our start and end points, while counting like incrementing our current count and comparing it with the end point, and at the end like stopping the counting. Therefore, within one task there are certain steps, where we perform certain sub-tasks.

Following our declarative knowledge, we define the procedural knowledge, called ‘production rules’. Our first Production rule is the ‘start’

```
(p start
     =goal>
          ISA count-from
          start =num1
          step start
     ==>
     =goal>
          step counting
     +retrieval>
          ISA count-order
          first =num1
)
```

The first part of this production rule, matches our first goal, which is a ‘count-from’ chunk where ‘step’ slot contains the value ‘start’. When ACT-R locates this match, it will look at our ‘count-from’ chunk, declared above to learn the value of num1, which is 2 in our case. When this is done successfully, ACT-R is ready to perform the actions specified after the ==> sign. The first step ‘=goal>’ tells ACT-R to replace the value in ‘step’ chunk to ‘counting’. The second part ‘+retrieval>’ tells ACT-R to go into the declarative memory, find a ‘count-order’ chunk with the ‘first’ slot containing num1, which is 2 in our current case. The next production rule will make use of this retrieved chunk. This retrieval puts our declarative chunk ‘c’ in the retrieval buffer.

When the ‘step’ is set to ‘counting’ and the retrieval from the previous step is successful, then the following production rule is selected by ACT-R.

```
(P increment
     =goal>
          ISA count-from
          start =num1
          - end =num1
          step counting
     =retrieval>
          ISA count-order
          first =num1
          second =num2
==>
     =goal>
          start =num2
```

```
      +retrieval>
           ISA count-order
           first =num2
      !output! (=num1)
)
```

The rule is called ‘increment’, again a name that means nothing for the model and there is only for the convenience of the modeler. This rule says that, if we retrieved a ‘count-order’ fact from the first step (we did and it was ‘c’) then put the value in the ‘first’ slot into num1 and the ‘second’ slot into num2. Now num1 contains 2 and num2 contains 3. If our goal is count-from and we are still counting and the num1 is not the same as the value (checked by the – sign in front of ‘end’) in the ‘end’ slot of our goal (which is still 5), then we should continue counting from num2 (which is now 3). We do this by setting the ‘start’ slot of our goal to num2 and retrieving a chunk where the ‘first’ slot contains num2. This will retrieve our declarative chunk ‘d’.

This rule will continue firing until we reach a stage there our number is the end point and we should stop counting. The other stopping condition is when we keep on counting but the declarative chunks do not contain the a chunk for the next number. In effect, like a human subject, if the model does not know enough it cannot fulfill its goal.

The last line ‘!output’ is again for the modeler’s convenience and shows the requested variable on the output screen. From a modeling point of view, it performs no function and takes no time. We will see later that, if we want the model to give us any responses we need to use the motor functions of ACT-R to tell our model to type the result out or speak. In such cases, the motor action will take some time.

Assuming that the model continues until counting and retrieves the chunk ‘e’, the above production rule cannot fire any more because of the line ‘- end =num1’. We need another production rule to cover this case.

```
(P stop
     =goal>
          ISA count-from
          start =num
          end =num
     step counting
==>
     =goal>
          step stop
          !output! (=num)
)
```

This production rule is fired then we have the same number as the start and end points of our count-from goal. In this case, there is no need to count any more, so we set the ‘step’ slot to ‘stop’. There are no other rules that declare any action when the ‘step’ is ‘stop’. Therefore the model terminates.

When we observe the output of the model, we see each step with an associated timing. Each production rule has a time period associated with it.

```
Time 0.000: Start Selected
Time 0.050: Start Fired
Time 0.100: C Retrieved
Time 0.100: Increment Selected
2
Time 0.150: Increment Fired
Time 0.200: D Retrieved
Time 0.200: Increment Selected
3
Time 0.250: Increment Fired
Time 0.300: E Retrieved
Time 0.300: Increment Selected
4
Time 0.350: Increment Fired
Time 0.350: Stop Selected
Time 0.400: F Retrieved
5
Time 0.400: Stop Fired
```

```
Time 0.400: Checking for silent events.
Time 0.400: * Nothing to run: No productions, no events.
```

These time values tell us when each rule is fired and on the whole, how long does it take for the model to accomplish a task. In this case, it took our model 400 ms to count from 2 to 5.

ACT-R/PM – Perceptual Motor functions

A cognitive task is usually a combination of mental cognitive tasks and perceptual and motor activity. For this reason, ACT-R/PM (Perceptual – Motor) was introduced. (Byrne and Anderson in Chapter 6 of Lebiere and Anderson 1998). ACT-R/PM is strongly influenced by EPIC.

The main perceptual task in ACT-R is vision. In the vision module the details of the computer display, which forms the visual field of ACT-R/PM is made available to the model in terms of declarative chunks. In addition to vision, an audition module is provided, in the same manner as the vision module. The sounds made by a computer application are represented as chunks in ACT-R’s declarative memory.

Two motor functions are also a part of ACT-R/PM. The speech module simulates a verbal response by the model and the motor module simulates mouse moves or keystrokes.

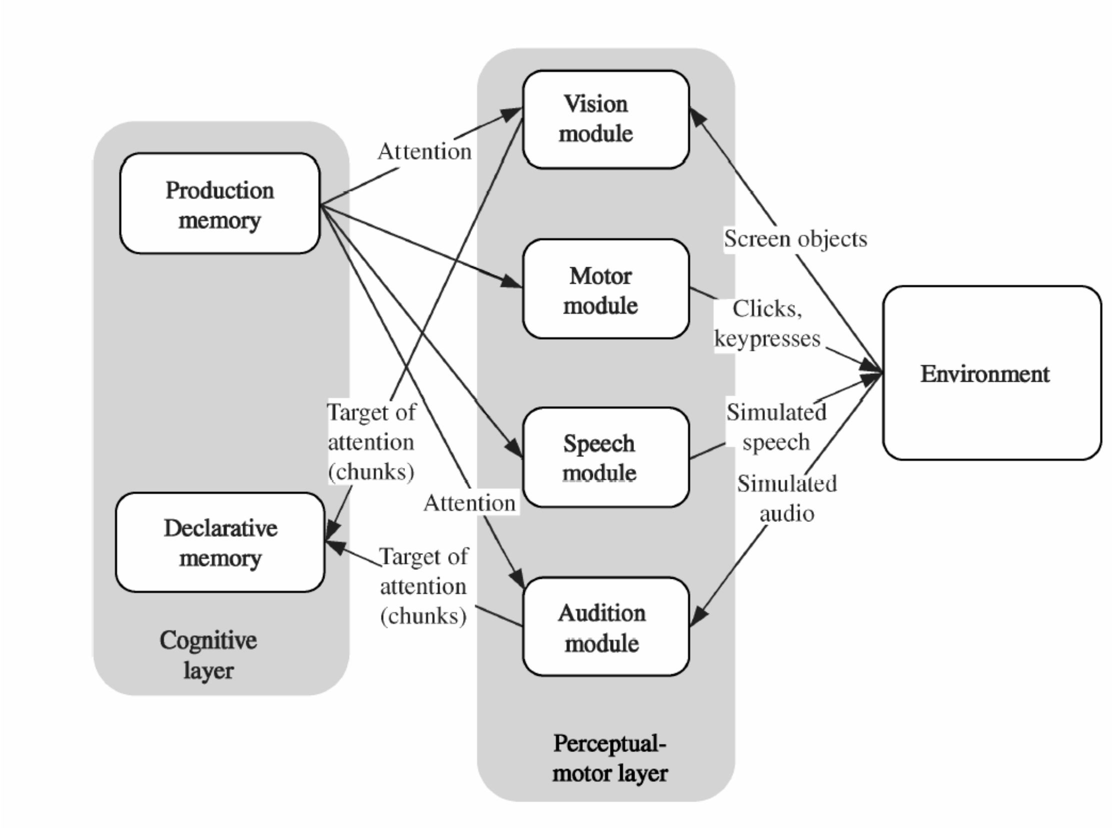


Figure 9. ACT-R/PM Components

ACT-R/PM has been designed and built to provide a more complete psychological theory of human performance. The perceptual-motor system is conceptualized as a layer between cognition and the external world.

ACT-R/PM - Visual Interface

Although it is easier to represent high-level cognitive processes as abstract knowledge in forms of declarative and procedural knowledge, it is much more difficult to find a relevant level of abstraction for lower-level processes. Most of the current theories assume that these processes can provide some sort of abstract description of the stimulus provided. However this creates some problems. First, by doing this, the theorists are introducing an unanalyzed level of freedom by assuming a processed representation of the input. As an ad-hoc abstraction mechanism can be used, it is never clear if the model's success depends on the representation chosen or the correctness of the theory. Another problem is, by doing this abstraction the theorists are ignoring the side factors affected. For example, if the visual input requires a number of attentional fixations, the attentional shifts may be an important but ignored part of processing.

In order to introduce a plausible theory of visual attention and perception, Anderson, Matnessa and Lebiere (Anderson, Matnessa and Lebiere in Chapter 5 of Anderson and Lebiere 1998) have introduced a synthesis of the spotlight metaphor (Posner 1980), Feature Integration Theory (Treisman and Sato, 1990) and the attentional theory (Wolfe, 1994) .

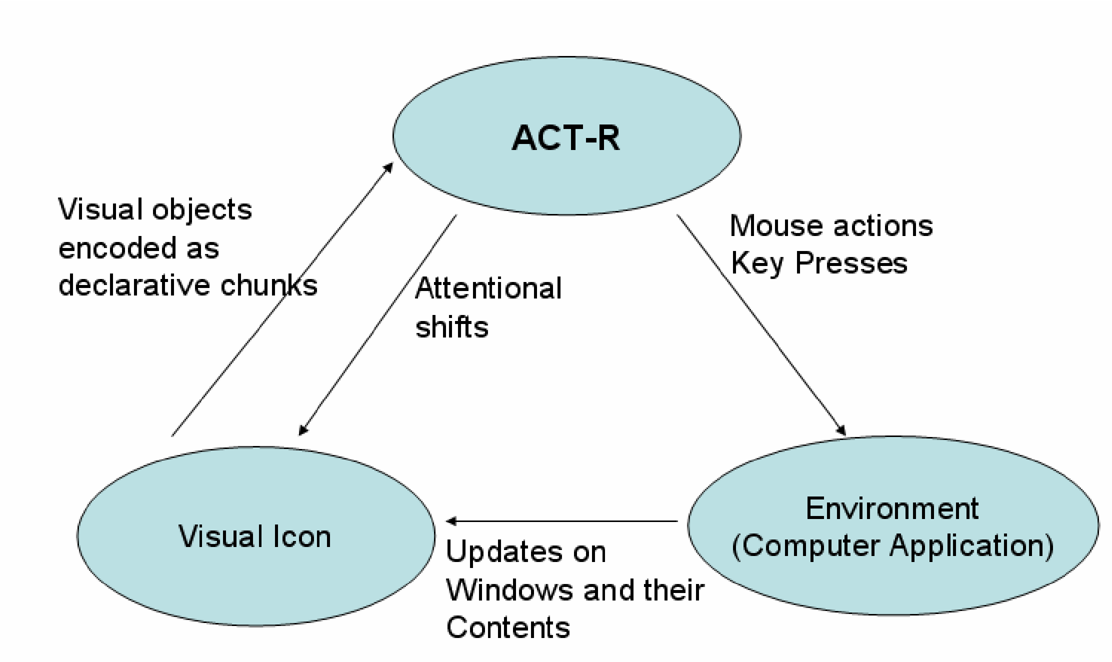


Figure 10. Relationship among ACT-R, the environment and iconic memory.

There are three components involved in the Visual Interface (Figure 10).

- ACT-R, the higher level cognition,
- The environment, which is simulated with a computer program in this case

- Iconic Memory, which represents the information displayed on the screen in terms of its features.

When ACT-R moves attention around the screen, it synthesizes the features on the attended location into a declarative chunk, which can be used by the production rules. The vision module in ACT-R models several visual processes like attending to a certain part of the display, recognizing the items – usually text strings or letters. It can keep track of attended portions of the display. The items 'seen' are also kept in a visio-spatial memory and they decay.

There are several constructs in ACT-R/PM that control the visual interface. For example to find a visual location that has not been attended before, we use

```
==>
   +visual-location>
     ISA   visual-location
     attended    nil
```

and to put the attended item into the visual buffer we use

```
   +visual>
      ISA          visual-object
      screen-pos  =visual-location
```

and to check that if it is a text item and the value is 'G' we use

```
   =visual>
      ISA          text
      value        "g"
```

In other words, with the use of simple constructs in ACT-R like buffers, chunks and production rules, we can move visual attention around the screen.

ACT-R/PM –Motor functions

Like the visual interface, ACT-R/PM can perform various hand movements, create speech or listen by using the buffers and chunks. Like vision, these are not real movements or real speech. Rather ACT-R/PM calculates the elapsed time for these actions and reports the time when these functions are performed.

To give an example, in order to press a key on the keyboard, we need to place a chunk into the ‘manual’ buffer.

```
+manual>
   ISA          press-key
   key          "a"
```

Likewise, sounds that need to be heard or speech that will be produced are always represented as chunks in various buffers. Using today’s technology, it is not hard to connect ACT-R/PM to robot arms, speech recognition modules or speech synthesizers. But, rather than creating a human like object, ACT-R/PM is more useful as a simulation model that acts on abstract objects. This makes it easier to follow its processes.

The Present Research

As discussed in previous sections, there is a large number of studies on how the visual search is affected in the presence of a salient object or a singleton. Experiments have been conducted to show that salient objects capture attention at the pre-attentive stage in a bottom up manner (Theeuwes 1992) or during goal driven processing in a top-down manner (Bacon and Egeth 1994). There are also studies that suggest that the type of processing is a function of time, and in the early stages of visual search bottom-up processes capture attention, whereas in the later stages top-down processes are more dominant. (Deco, Pollatos, Zihl 2002)

Studies by Theewes (1992), Turatto and Galfano (2000) and Theeuwes and Burger (1998) show that presence of a salient object in a display interferes with the visual search. They explain this by suggesting that the pre-attentive stage processes the salient objects first. Theeuwes and Burger take salience in a most common-sense way, assuming, for example that, an object with a different color than the others is more salient. Turatto and Galfano (2000) go a step further and assume that an object with a different color or shape or brightness is more salient than the others. The findings of both experiments support the assumption that a salient object attracts attention at pre-attentive stage.

On the other hand Bacon and Egeth (1994) claim that the nature of the task determines the role of the salient object. They claim subject can chose between feature-search and singleton-detection modes. The feature-search mode is not affected by the presence of a salient singleton, whereas the singleton-detection mode is susceptible to such effects.

In all the above experiments, a singleton in the irrelevant dimension has never been the target. Therefore, always negative effects of the distractors in other dimensions are measured. In our research project, we first start with the singleton case to verify if Theeuwes' (1992) bottom-up hypothesis is true.

Hyphothesis 1: Subjects can not ignore a salient singleton in an irrelevant dimension.

Theeuwes' experiments (1992) show that the subjects cannot ignore a singleton in an irrelevant dimension unless the properties of the target and distractor singletons are available to the subject and remains unchanged over the time.

However, these experiments make use of setups where the irrelevant singleton and the target are never the same. His experiments measured the degrading performance in locating the target in the presence of an irrelevant singleton.

In case there is a possibility that the irrelevant dimension singleton can also be the target, based on Theeuwes' claims we can predict that this item will attract attention first even if the subjects are instructed to ignore any differences in the irrelevant dimension. This would result in a performance increase for the cases where the irrelevant singleton and the target are the same item over the cases where they are not the same item. Such a result provides support for the bottom-up hypothesis. On the other hand, if no such effects exist, this means that the subjects are clearly able to mask the singleton effect in an irrelevant dimension under top-down control. Such a result would contradict Theeuwes' claims.

Hyphothesis 2: Subjects cannot ignore the irrelevant dimension.

Going further, if the visual search is really under top-down control, it would be easier for the subjects to ignore the effects of the irrelevant dimension, even if the items in the irrelevant dimension are not a singleton. In such a case the feature-search would definitely be used instead of singleton-detection mode. According to Bacon and Egeth(1994), in feature search mode, subjects will be totally capable of ignoring the irrelevant dimension. However, if there is a bottom-up control, more salient items would be processed before the other items and therefore we would expect dissociation based on the target's features in the irrelevant dimension.

In order to test these claims, we designed experiments where items differ in two dimensions (form and chromatic features). By investigating the responses given by the subjects looking for a target form, as the chromatic features of the target and distractors change, we can measure the effects of chromatic features on subject performance. The chromatic features will be totally irrelevant for the task and the subjects will be told specifically to ignore this dimension. If the chromatic features cannot be totally ignored, we can see this as a support to the bottom-up hypothesis. On the other hand, if these features can be ignored, this would support Bacon and Egeth’s claims and the top-down hypothesis would be corroborated.

CHAPTER 2

METHOD

Theeuwes (1992) and later Bacon and Egeth (1994) studies show the effects of a singleton in the display. However, they both do this in a certain experimental setup, using some geometric shapes. If these results can be generalized, then we should be able to observe the same effects when we use a different form. For this reason, in our experiment, we have chosen the capital letters of the alphabet. In fact, Theeuwes (1998) uses capital letters to produce same effects and our experiment is similar to this experiment in some respects.

Two experiments were conducted. Both experiments used letters of the alphabet in varying numbers and layouts and the target was always the letter ‘G’. The subjects were instructed that the colors have no relevance to the task, they were there to distract them and they should try not to get distracted by colors as they tried to locate the target as quickly and accurately as possible.

In the first experiment, there was only one item with a unique color to test the assumption that the attention is really captured by a singleton. We expected that, according to Theeuwes, the subjects would react much faster if this singleton is also the target item.

In the second experiment, the colors on the screen were evenly distributed over the trials and targets. As there were no more singletons in most of the displays, according to Bacon and Egeth, we expected feature-search would be forced which result in top-down masking of any effects of the chromatic dimension. This would result all distractor effects to disappear and the response times would be the same regardless of the chromatic features of the target. Any difference in the subjects’ performance in locating the targets in different colors would be taken as supporting evidence for bottom-up approach, where the chromatic dimension is playing a role in spite of the clear instructions that it should be ignored.

Experiment Task

The task was to search a certain letter in a frame size of n letters. In each trial there were m letters of one color and $n-m$ letters of other color.

In Experiment 1 (Singleton-case) only one of the letters was in a different color. (m was always 1 or $(n-1)$) We have tested the conditions where the target was the singleton, target was not the singleton and there was no target.

In Experiment 2 (Mixed-Case) the number of letters in the first color (m) changed from 0 to n. For each m, there were three trials, where the target letter was in the first color, was in the second color and there was no target letter.

Experiment 3 was a variation of Experiment 2, in order to verify the effects observed in Experiment 2 can be generalized to other forms. For this purpose Windows' Wingdings font was used instead of the capital letters of the alphabet.

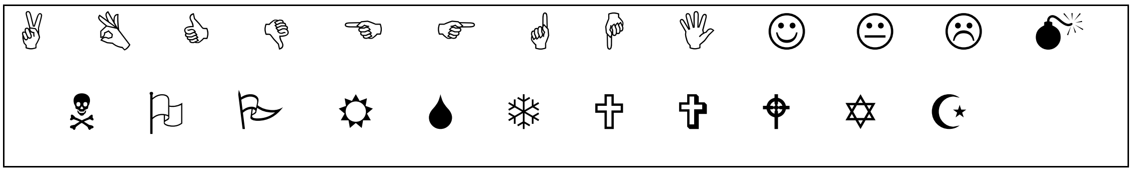


Figure 11. Windows Wingdings font. The target is the victory sign (✌). Note that a lot of 'hand' figures are present in this font, which makes the task even more difficult for the subjects.

The same experiments were also presented over the Internet, as a separate study. The experiment results from Internet are not considered a part of this research. This was done as a small side study to measure if similar experiments can be conducted over the Internet and the results were briefly discussed in a separate section.

Experiment Setup

A laptop with 14” LCD Screen was used for the experiment (Fujitsu-Siemens Lifebook). The screen resolution was 1024x768.

Five or twenty letters (n=5 or n=20) from the English Alphabet were presented in three different color combinations and two different layouts. Each letter was presented in Tahoma 24pts Bold Font.

In the *circular* case the letters were placed on an imaginary circle around the center point with a radius of 300 pixels. In the *random* case the letters were presented in the same angular locations but at different distances from the center in such a way that the average distance is 300 pixels.

In the *two color* (TC) condition letters were presented in red and blue. Red had the RGB values of R=0xFF, G=0x00, B=0x00; and blue had the RGB value of R=0x00, G= 0x5A, B= 0xFF. These colors were chosen because they have the same gray level equivalents and therefore the same luminance. (Calculated using the formulation given in http://www.compuphase.com/cmetric.htm).

In the *brightness* (BR) condition letters were presented in red and dark red. Red had the RGB values of R=0xFF, G=0x00, B=0x00; and dark red had the RGB value of R=0x70, G= 0x00, B= 0x00. These colors were chosen because they are different shades of the same color, therefore they have a different luminance.

In the *multi-dimension* (KY, short for Kırmızı-Yeşil) condition letters were presented in bright green and dark red. Green had the RGB values of R=0xFF, G=0xFF, B=0x00; and dark red had the RGB value of R=0x40, G= 0x10, B= 0x10.

These colors were chosen because they differ in both color and luminosity dimensions.

So effectively there were 12 setups for each experiment where there were two choices in two variables and three choices in the color dimension.

20 letters / Two Colors / Circular
20 letters / Brightness / Circular
20 letters / Multi-dimension / Circular
20 letters / Two Colors / Random
20 letters / Brightness / Random
20 letters / Multi-dimension / Random
5 letters / Two Colors / Circular
5 letters / Brightness / Circular
5 letters / Multi-dimension / Circular
5 letters / Two Colors / Random
5 letters / Brightness / Random
5 letters / Multi-dimension / Random

The subjects started the experiment by entering their names, age, gender and whether they are using eye-glasses or not. Then they choose one of 12 experimental setups. In controlled experiments, the experimenter instructed the subjects on which setups should be done. For the experiments done over the Internet, the program randomly presented one of the setups and the subjects had the freedom to change it if they like. The target letter (always G for Experiment 1 and 2 and the victory sign (✌) for Experiment 3) was introduced to the subjects before the experiment begins. The victory sign was especially chosen because the Wingdings font has many hand signs and the subjects are not able to distinguish the target from the distractor only by looking at its superficial features. In Experiment 1 and 2, when the target was present, the letter C was not on the display. But in the no-target conditions there was always a capital C in the display. This was done to make sure that the subjects really look for G. The subjects, who were replying before identifying the target, would be making a lot of errors due to C and their data would be disqualified. The same consideration is not observed for Wingdings font because there are a lot of hand symbols. Leaving them out in the target case reduces the number of available shapes. So in Experiment 3 there were a number of hand shapes in both target and no-target displays.

The subjects are specifically instructed that the colors were not important and their only task was to find the target letter as quickly and as accurately as possible. In the singleton case, they were specifically instructed that there was only one letter with a different color, their task was to find the target letter, the singleton had the same probability of being the target as any other letter in the display therefore the color difference had no relevance to the task and should have been ignored. In the first experiment set of each subject, the experimenter showed the first few displays of a sample experiment and instructed the subjects while answering a few displays himself.

During the experiment, before each display, a fixation point was presented (like Theeuwes 1998, Bacon and Egeth 1994) followed by a set of capital letters which constitute the items in the experiment display. The fixation point was a grey diamond shape (Windows Wingdings Bold 14pts: ◆) and was shown for 1500 ms followed by a grey dot shape (Windows Wingdings Bold 14pts: ●) presented for 750 ms. Then the letters are shown. The fixation point was not removed when the letters were shown. The display remained for five seconds or until the subject pressed a key on the keyboard. At this point the screen was cleared and the diamond fixation point for the next set was presented. At all times, at the top left corner, there was a counter, in dark grey, telling the subjects how many displays are shown and what the total number of displays are.

When the letters were presented, the subjects were asked to press ‘A’ if the target was in the display and press ‘L’ if the target was not present. The letters ‘A’ and ‘L’ were chosen because they are on the opposite ends of the keyboard and the subjects were able to keep their hands on them to minimize the motor activity time. This is in line with Bacon and Egeth (1994) experiment. Bacon and Egeth had used ‘Z’ and ‘/’ instead but ‘A’ and ‘L’ keys were preferred in our study because of the Internet experiment. The letters ‘Z’ and ‘/’ are not always in the same place in different keyboard layouts like Turkish or German.

The letters in the display were randomly chosen. However, the letter C, which closely resembles to G was not presented in target-present condition and always presented in no-target condition. This reduced the possibility of subjects mistakenly giving a positive answer for C in no-target condition and made sure that they have seen C and eliminated it as a non-target object in no-target conditions. The time between the presentation of the display and the subject’s response time was recorded to the nearest millisecond.

For Experiment 2 and 3, in 20 letter displays there were 61 different cases. In each display there were 20-m elements of color1 and m elements of color2. The target was in color1, color2 or no target was on display. (For m=0 and m=20 all the letters on the display were in the same color so there are only 2 possibilities instead of 3. Therefore we had 61 cases instead of 63.) For 5 letters there were 16 displays. In Experiment 1, the same number of displays presented although the setup of this experiment had not forced any such consideration. The subjects had to reply to each display in 10 seconds.

To warm the subjects up, before the experiment sets were presented, 10 random displays which were similar but not the same as the ones used in the experiment were shown to the subjects.

At the end of the experiment the percentage of correct responses and average response time were presented to the subjects. This was done only for getting the subjects more involved in the process. The subjects were told that this was only for them to evaluate their performance and the results of the experiment would not be individually graded.

Each experiment was held in three different sessions in different days for each subject. So, for three experiments, each subject had the possibility of eight sessions, four sets in each session. Subjects had a short break between the sets. The order of the sessions was intermixed so that no two subjects did the experiment sets in the same order. This was done for evenly distributing any possible effects due to learning or being bored.

At any one session the subjects completed one of the following sets:

20 letters / Experiment 2 / Two Colors and Brightness
5 letters / Experiment 2 / Two Colors and Brightness
20 letters / Experiment 1 / Two Colors and Brightness
5 letters / Experiment 1 / Two Colors and Brightness
20 and 5 letters / Experiment 1 / Multi-dimension
20 and 5 letters / Experiment 2 / Multi-dimension
20 letters / Experiment 3 / Two Colors and Brightness
5 letters / Experiment 3 / Two Colors and Brightness

All the experiments were done in the meeting room of a software company under the same light conditions and using the same computer. The subjects were asked to sit in the most comfortable position and adjust the screen for best viewing. The experimenter left after giving the instructions and subjects were alone in the meeting room during the experiment. Experiments were conducted over a period of three months where each subject was tested in different times on different sets.

Not all the subjects did all the experiments. The experiments were randomly distributed among the subjects. At least 12 subjects did each setup, except for experiment 3, where the minimum subject number was 8.

The same experiments were also presented over the Internet, as a separate study. The experiment results from Internet are not considered a part of this research.

Elimination of the elapsed time for motor activity

The reaction time measured in all experiments can be divided into three components: (i) time elapsed for the target to realize a change in the display, i.e. the appearance of the letters on the screen. (ii) visual search and decision, (iii) motor activity to press the correct key.

A separate study was conducted to isolate the elapsed time for the visual search and decision component from the rest of the activities involved in the task.

In this study, nine subjects were asked to press letter ‘A’ as soon as they see the letters appear on the screen. The results of this measurement are provided in Appendix 1.

The average reaction time of the subjects for this task was 257 ms (std=28.35). This figure was subtracted from the reaction time measurements for the experiment, yielding a more precise figure for the relevant portion of the whole task.

Experiment 1. Singleton Case

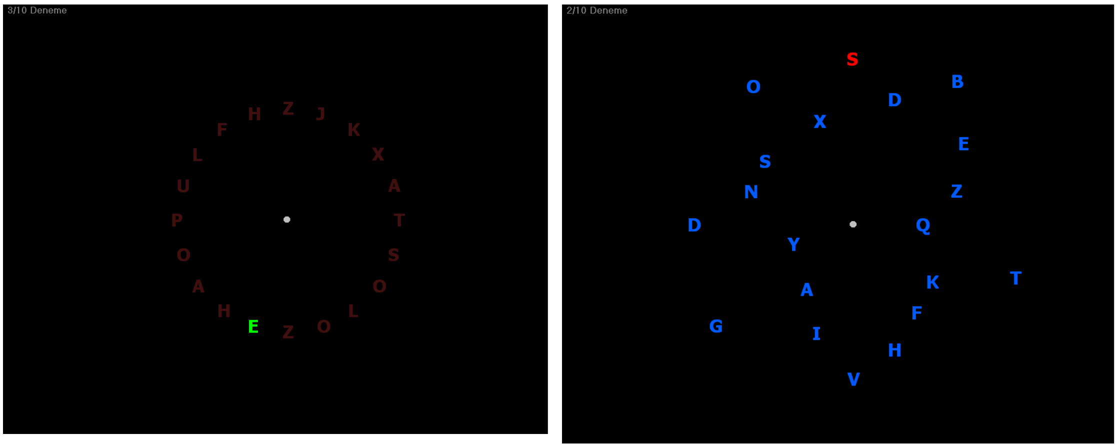
Subjects

A total of 33 subjects were tested (25 male and 8 female). 135 experiment sets were done by male subjects and 27 by female subjects. The subjects were the employees of a computer company, most of them using computers daily for their jobs. Their ages varied between 18 and 56. Only two subjects (1 male and 1 female) had little or no interaction with computers before. Most of the subjects had university degrees. One subject has a high school degree and two subjects had primary school education.

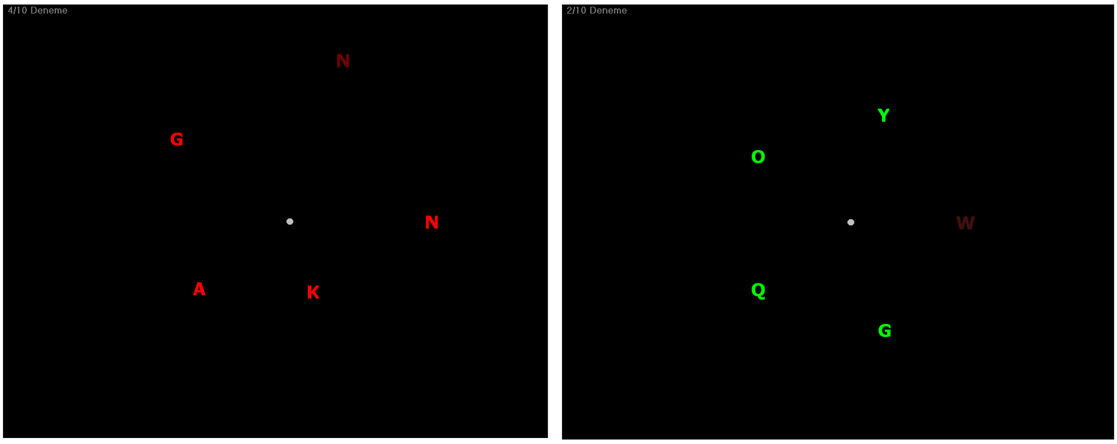
Not all the subjects did all the setups. At least 12 subjects did each setup successfully. The number of subjects successfully completed each set are provided in Appendix 6.

Design and Stimuli

The design of the experiment was as described above. In each display there was only one letter in a different color. All other letters were of the same color except this singleton which is our ‘irrelevant dimension singleton’. The target was the letter ‘G’, with the equal likelihood of being at any position and in any color, including the ‘irrelevant dimension singleton’. The position of the target and the position of the irrelevant singleton were randomly chosen. The target was absent in one third of the trials. These trials were used as verifications that the subjects were actually looking for the target letter before they replied. The subjects were specially briefed to ignore the color differences and the fact that there was one element with a different color which does not give them any hint to find the target.



20 Letters, Multi Dimension, Circular   20 Letters, 2 colors, Random



5 letters, Brightness, Random   5 letters, Multi Dimension, Circular

Figure 12 Sample displays from experiment 1. Note that, there is always a single letter in a different color.

Results

For any single display the incorrect replies or no replies (cases where subject was failed to press a key before the display is removed automatically) were classified as error. For any setup, if a subject's error rate was more than 20%, then the answers given by the subject for that particular set were completely excluded from the data. For the cases where the subject's replies contained less than 20% errors, only the error replies were excluded from the data.

The average search time for 5-letter displays was 734 ms (std= 353, median=657) for no-target condition and 466 ms (std=237, median=411) for the target condition, which is very close to the numbers reported in Theeuwes (1998), as this experiment is very similar to the Incompatible condition for 5 letters in the first experiment of Theeuwes.(1998). The Theeuwes' result is around 675 ms, but this also contains the motor time for the keystroke. (466+257) = 723 ms is our response time and this is close to Theeuwes' result.

The average search time for 20-letter displays was 2580 ms (std=1034, median = 2394) for no-target and 988 ms (std=660, median=822) for target conditions.

In this experiment, we expected to find faster visual search times when the irrelevant dimension singleton is also the target. Because if the color singleton is attracting the attention first, the search times should be much lower when the target is the color singleton i.e. attended first. Out of our six different experiment setups, we obtained no clear indication that this wss the case. The results of search times are given in Appendix 2. We had expected m=1 and target in color1 (cond=1) cases to be faster than m=4 or 19 cases. Because when m=1 there was only one item in color1 (singleton target case), whereas in the other case there are 4 or 19 items in color1 (non-sinleton target case), the other item being in color2. Likewise, we had expected that for the targets in color2, m=4 or 19 cases would be faster than m=1. The number of times our assumption holds is summarized in Table 1.

Number of Letters & Singleton Color	**Number of Setups**	**Number of setups where it is faster to locate target if the target is in irrelevant singleton color**
5 letters – color1	6	3
5 letters – color2	6	5
20 letters – color1	6	5
20 Letters – color2	6	2

Table 1. Number of cases where the expected results are observed.

In 24 different setups only 15 setups support our assumptions and this can be regarded as chance. The two 5-out-of-6 cases also cannot be trusted because they occurred for different colors. It was also not possible to get expected results by evaluating on the basis of each color combination (i.e. two colors case for 5 and 50 letter conditions) or layouts (Random layouts vs. circular layouts). In order to verify our result an ANOVA analysis was conducted for each color to see if there was any difference in response times when the target was a singleton and. Of 64 different ANOVA analysis's (32 setups and two colors) only six showed $p<0.05$. Therefore we concluded that singleton effects were not observed.

By evaluating the results in the chromatic dimension, the color of the target, without considering the fact that the target was the color singleton or just one of the many items in this particular color, we saw that except one case (20-Letter, BR, Random) the subjects were always faster to find bright red targets than dark red targets, bright green targets faster than dark red targets and blue targets faster than red targets. However, an ANOVA analysis for the color dimension also showed only one setup out of twelve has $p<0.05$. The results are reported in Table 7. This shows there is no effect of the color dimension except in 20 letter KY Circle layout. .

When we compared the results from the 5 and 20 letter cases (Appendix 5), the results indicated that, it took on the average 3.54 times longer to decide that there was no target letter in the 20-letter display compared to 5-letter displays. However, if the target was present, then the visual search time for 20-letter displays took only 2.04 times longer than 5-letter displays. This result shows that, if there is a target in the display, the time spent per item is considerably lower in the 20-letter case whereas in the no-target case the difference is much lower. Ideally, if the search was a linear process, this ratio should have been 4.00 for both cases, whatever the search strategy is.

The relation between the search time and the location of the target was also analyzed. For 20-letter displays, the visual field is divided into four quadrants and the visual search time is examined for each quadrant. The quadrants are numbered as follows (Figure 13)

1 – Lower Right

2 – Lower Left

3 – Upper Left

4 – Upper Right

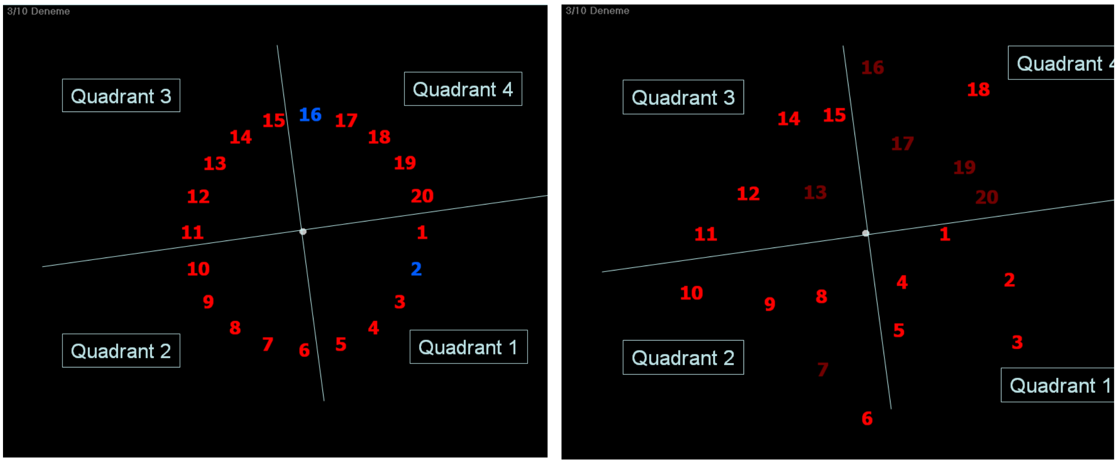
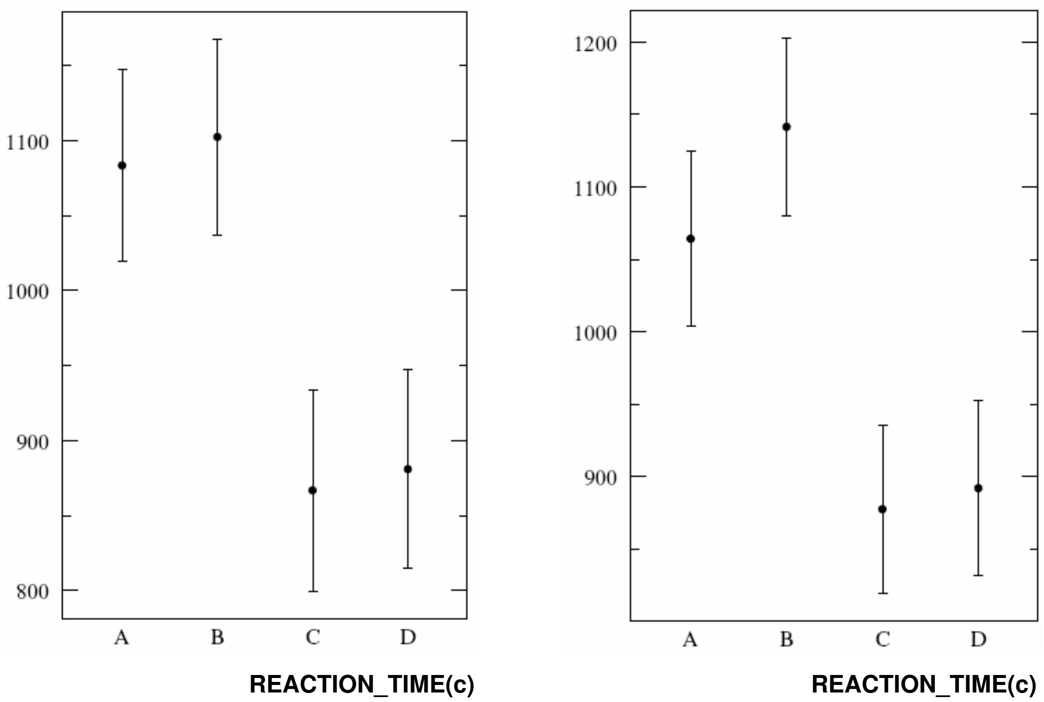


Figure 13. The position of quadrants and items in the display. Item position 1 is considered to be in the lower quadrant whereas position 11 is in the upper quadrant.

The detailed results are presented in Appendix 8. Results show that was easier to find the target when the target was in one of the upper quadrants. The difference between upper and lower quadrants was around 200 ms which is quite significant considering the average reaction time is around 988 ms. This means the search time in upper quadrants is %20 shorter than that in the lower quadrants.

An ANOVA analysis is also conducted on the reaction times based on the quadrants. There is an effect of location on reaction time for each color. For color1 $F(1,4) = 14.32$, $p<0.0001$ and for color2 $F(1,4)=17.99$, $p<=0.0001$. Also 95% confidence level graphs (Figure 15) show that the upper and lower halves were generating this effect. Also a post hoc Tukey test was conducted. This test was done on all samples for this experiment fro colors 1 and 2 and shows that quadrants in the upper and lower halves of the screen are grouped together.



Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
3,00	389	866,71	
4,00	396	880,87	
1,00	424		1083,43
2,00	413		1102,41
Sig.		,991	,978

Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
3,00	457	877,68	
4,00	413	892,24	
1,00	417		1064,20
2,00	407		1141,41
Sig.		,987	,281

Color1 Color2

Figure 14. 95% confidence level graph and post hoc Tukey analysis for experiment 1. (A=Quadrant1, B=Quadrant2, C=Quadrant3, D=Quadrant4). This graph shows that the upper (3&4) and lower (1&2) quadrants have an effect on the reaction time.

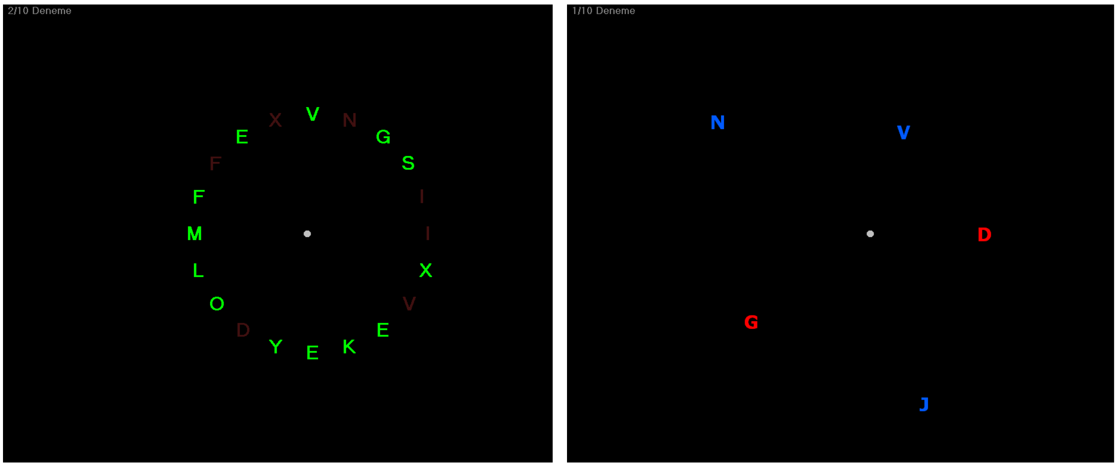
Experiment 2. Mixed Case

Subjects

The same subjects were used for Experiment 2. The procedure was also the same. The minimum number of subjects who successfully completed each set was 13.

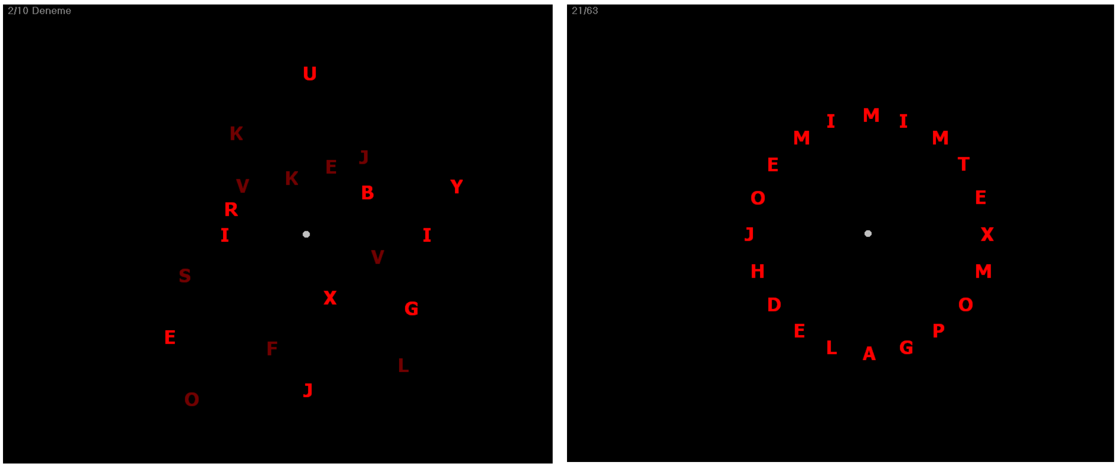
Design and stimuli

The design of the experiment was similar to Experiment 1 where each experiment set presented a series of displays in a color pair (BR, KY or TC), 5 or 20 letters in a random or circular layout. But in this experiment, each display had varying number of letters of each color setting. For example, for the two color case with 20 letters, there were cases where the number of red items was varied from 0 to 20. For each case the target was either red or blue or there were no target. Subjects were specially briefed to ignore the color differences.



20 letters, Multi Dimension, Circular

5 letters, 2 colors, Random



20 letters, Brightness, Random

20 letters, Brightness, Circular. (This display contains 20 bright red letters. Same display is also a part of two color case.)

Figure 15. Sample displays from Experiment 2. The number of letters in each color is varied from 0 to n.

Results

The same error exclusion conditions applied as Experiment 1.

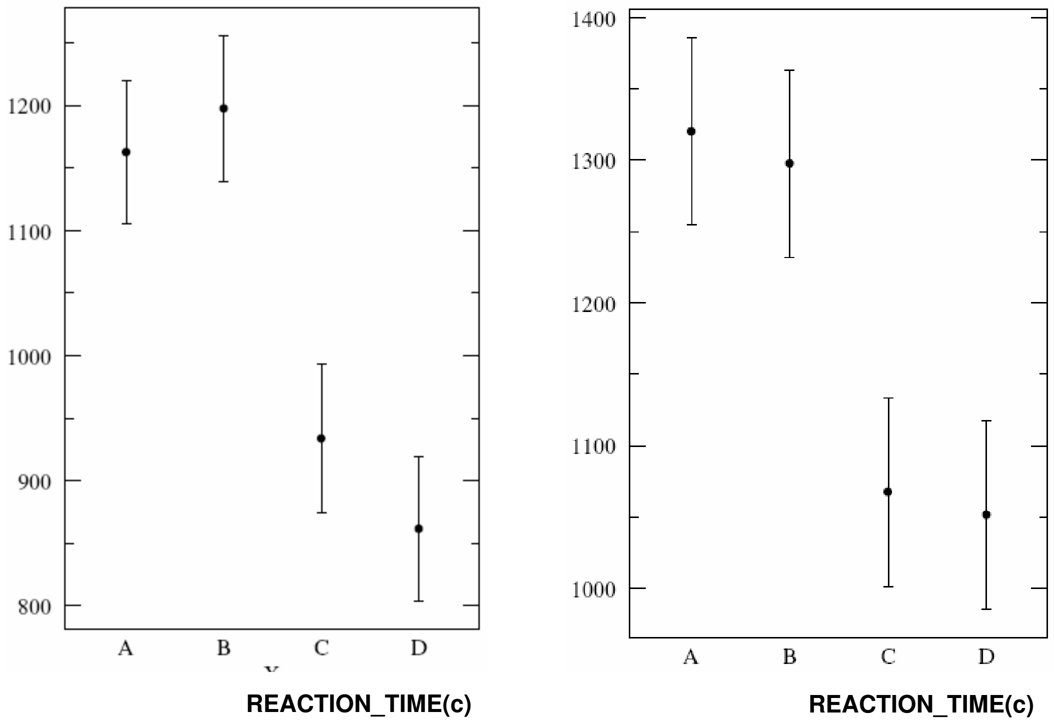
The search time results are presented in Appendix 2. The average search time for 5-letter displays was 789 ms (std = 299, median = 732) for no-target condition and 513 ms (std=239, median=451) for the target condition, which is 10% higher than the values reported for the first experiment.

The average search time for 20-letter displays was 2639 ms (std=759, median =2564) for no-target and 1111 ms (std=678, median=931) for target conditions. This is 11% higher than the values reported for Experiment 1.

When we evaluate the results for the color dimension, we see that except in one case (5-Letter, TC, Random) the subjects were always faster to find bright red targets than dark red targets, bright green targets faster than dark red targets and blue targets faster than red targets. The results are reported in Table 7. An ANOVA analysis was conducted on the different setups to investigate the variance of reaction time based on color. ANOVA analysis shows no consistent correlation between these two dimensions, except 20-letter KY circle and random cases. .

Comparing the results from 5 and 20 letter cases (Appendix 5), the results indicate that, it took on the average 3.38 times longer to decide that there was no target letter in 20-letter display compared to 5-letter displays. However, if the target was present, then the visual search time for 20-letter displays was only 2.08 times longer than that for 5-letter displays. This result is very consistent across different layouts.

The relation between the search time and the location of the target was also analyzed. Like the first experiment, the quadrant analysis indicate that for 20-letter cases, there was on the average 250 ms. difference in reaction time between the top half and bottom half of the display. It is faster to find the target if the target is in the upper half of the screen. An ANOVA analysis is also conducted on the reaction times based on the quadrants. There is an effect of location on reaction time for each color. For color1 F(1,4) = 31.98, p<0.0001 and for color2 F(1,4)=18.48, p<=0.0001. Also 95% confidence level graphs (Figure 16) showed that the upper and lower halves are generating this effect. The post hoc Tukey test grouped the upper and lower quadrants, verifying our findings.



Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
4,00	454	861,55	
3,00	430	933,79	
1,00	464		1162,72
2,00	445		1197,57
Sig.		,307	,837

Color1

Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
4,00	431	1051,63	
3,00	426	1067,54	
2,00	436		1297,78
1,00	435		1320,06
Sig.		,987	,966

Color2

Figure 16. 95% confidence level graph and post hoc Tukey analysis for Experiment 2. The pattern is quite similar to Experiment1.

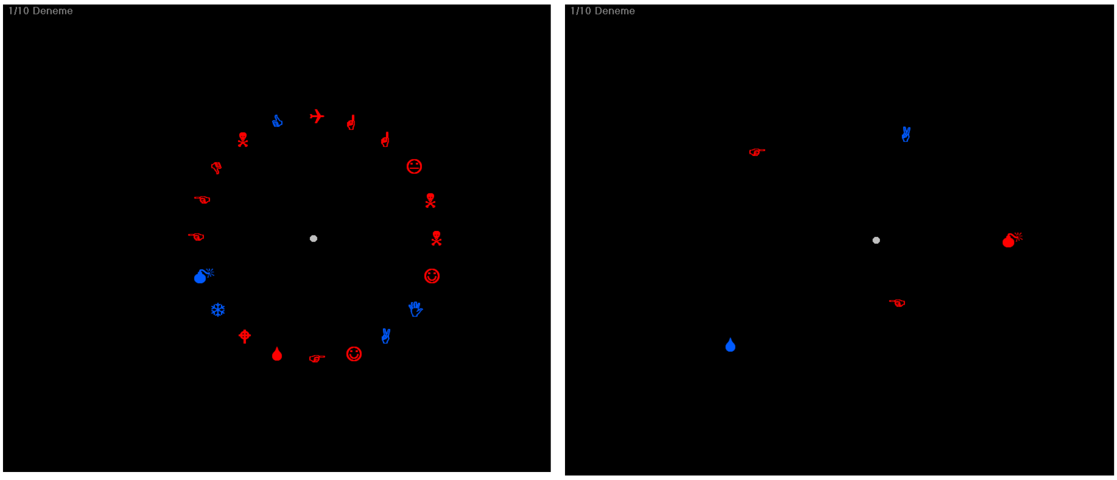
Experiment 3. Mixed Case Wingdings Font

Subjects

The same subjects were used for Experiment 1 and 2. The procedure was also the same. The minimum number of subjects who successfully completed each set was 8.

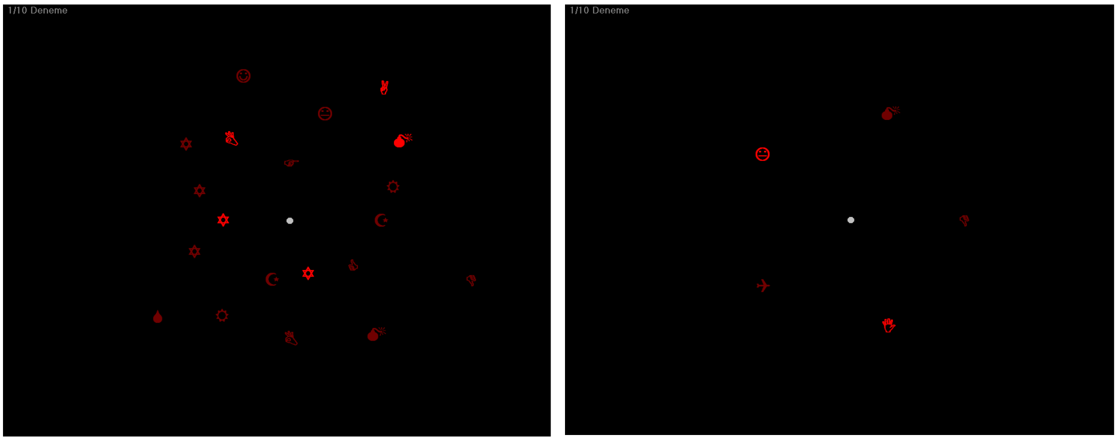
Design and stimuli

The design of the experiment was similar to experiment 2. The KY case was not included in this experiment because of time constraints.



20 letters, 2 colors, Circular

5 letters, 2 colors, Random



20 letters, Brightness, Random

5 letters, Brightness, Random

Figure 17. Sample displays from Experiment 3. Wingdings font was used.

Results

The same error exclusion conditions applied as Experiment 1 and 2.

The search time results are presented in Appendix 2. The average search time for 5-letter displays was 1373 ms (std = 490, median = 1323) for the no-target condition and 799 ms (std=428, median=711) for the target condition.

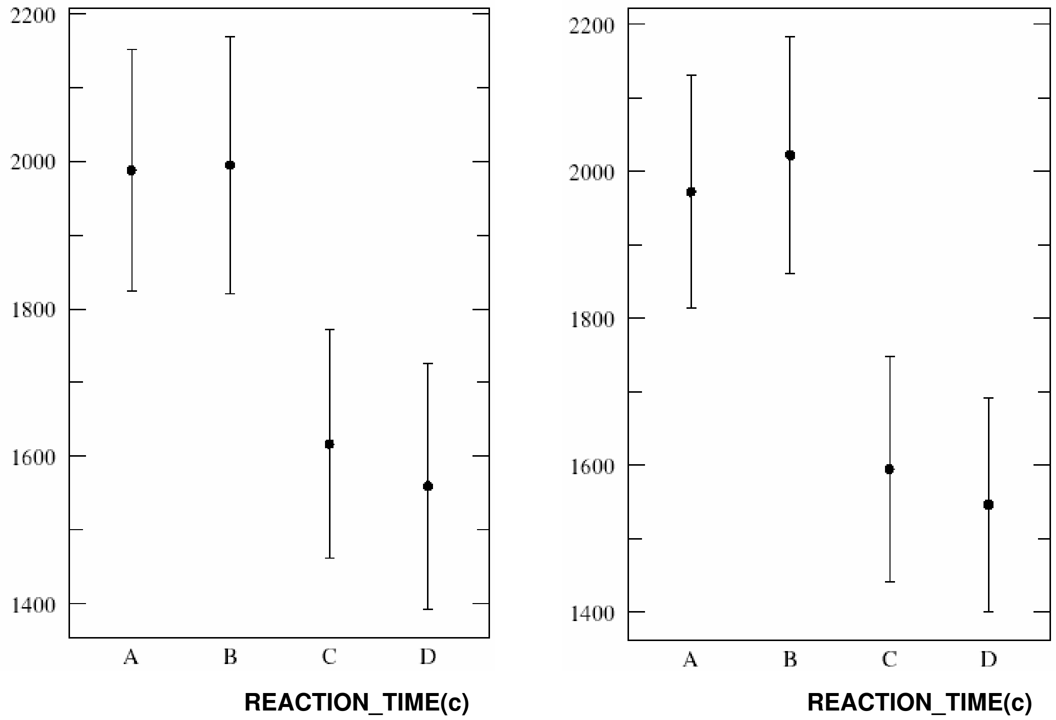
The average search time for 20-letter displays was 3877 ms (std=1050, median =3697) for no-target and 1774 ms(std=1106, median=1553) for target conditions.

Both for the 5 and 20 letter cases, the search time results were higher than the results of Experiment 2 which suggests that this task was more difficult than the other two experiments.

When we evaluate the results for the color dimension, the results do not show a noticeable difference. The differences in averages, which are smaller compared to Experiment 1 and 2, are in the opposite direction of the other two experiments. It is faster to find a dark red target among bright red distractors. These results are reported in Appendix 4. However, the ANOVA analysis shows no color effects for any setup. But, we have to note the fact that the KY condition was not done in Experiment 3.

When we compare the results from the 5 and 20 letter cases (Appendix 5), the results indicate that, it took on the average 2.84 times longer to decide that there was no target letter in the 20-letter display compared to 5-letter displays. However, if the target was present, then the visual search time for 20-letter displays was only 2.21 times more than that for the 5-letter displays. This result is very consistent across different layouts. This result is also compatible with the results from the first and second experiments but here the difference is not as striking as the other experiments.

The relation between the search time and the location of the target was also analyzed. Like the first two experiments, the results indicate that the upper half of the display was processed much faster than the lower half. An ANOVA analysis was also conducted on the reaction times based on the quadrants. For color1 $F(1,4) = 7.69$, $p<0.0001$ and for color2 $F(1,4)=9.96$, $p<=0.0001$. 95% confidence level graph is presented in Figure 18. Like the other two experiments, this experiment also showed strong location effects.



Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
4,00	176	1559,81	
3,00	201	1616,83	
1,00	181		1987,93
2,00	161		1994,81
Sig.		,964	1,000

Color1

Tukey HSD

QUADRANT	N	Subset for alpha = .05	
		1	2
4,00	204	1546,62	
3,00	181	1595,30	
1,00	171		1972,15
2,00	166		2021,99
Sig.		,972	,970

Color2

Figure 18. 95% confidence level graphs and post hoc Tukey analysis for Experiment 3.

The presence of the location effect indicates that similar processing patterns of the visual field is utilized by the subjects. Therefore the increase in response times may be due to the familiarity of the letters of alphabet or due to the fact that there are more characters in Wingdings font that share the characteristics of the target, compared to G and other letters of alphabet.

The Internet Results

A total of 195 valid experiment sets were received from Internet. The minimum age was 8 and the maximum age was 62, the average age of the subjects was 32. The distribution of experiment in different setups is provided in Table 2.

Set Size	**Experiment**	**INTERNET**	**LAB**
5	Exp 1	23	81
20	Exp 1	30	81
5	Exp 2	33	97
20	Exp 2	78	92
5	Exp 3	11	35
20	Exp 3	20	39

Table 2. Number of valid experiments.

When we examine the average response times we see that the response times from Internet are slower except for the 20-letter Experiment 2, where we have a more or less perfect match, suggesting that if more sets were received, the Internet experiment would yield similar average times. However, the difference in average times might also be due to the various distracting factors subjects have experienced while doing the experiment in their homes or workplaces.

Experiment	**Set Size**	**No Target Avg and stddev**	**No Target Median**	**Target Avg and stddev**	**Target Median**
1	5	954 ± 577	849	620 ± 403	521
1	20	3008 ± 1182	2818	1211 ± 812	982
2	5	992 ± 497	872	624 ± 332	553
2	20	2716 ± 809	2714	1172 ± 708	1003
3	5	1636 ± 687	1428	1043 ± 587	912
3	20	4560 ± 1253	4337	2055 ± 1265	1811

Table 3. The average search times.

The quadrant effects are very similar. In Experiment 1 and 2, the 200 ms speed difference between upper and lower half of the display is still present and this difference disappears for Experiment 3.

For the multi-dimension case (KY) much bigger effects are observed. The effects in other two cases (BR and TC) are not consistent or significant.

The increase in the search times from 5 letter sets to 20 letter sets show similarities, but the slope of increase is lower for no-target cases (Table 4).

	Increase rate for No-Target	Increase rate for Target
Exp1	3.35	2.18
Exp2	2.93	1.95
Exp3	2.87	2.00

Table 4. Increase in search times from 5 letter sets to 20 letter.

No ANOVA analysis is conducted on the Internet data.

On the whole, the Internet experiment also shows patterns quite similar to our experiment in laboratory conditions. The only problem looks like attracting more subjects. There should be more rewards or the experiment should be more interesting to reach a larger set of subjects over the Internet.

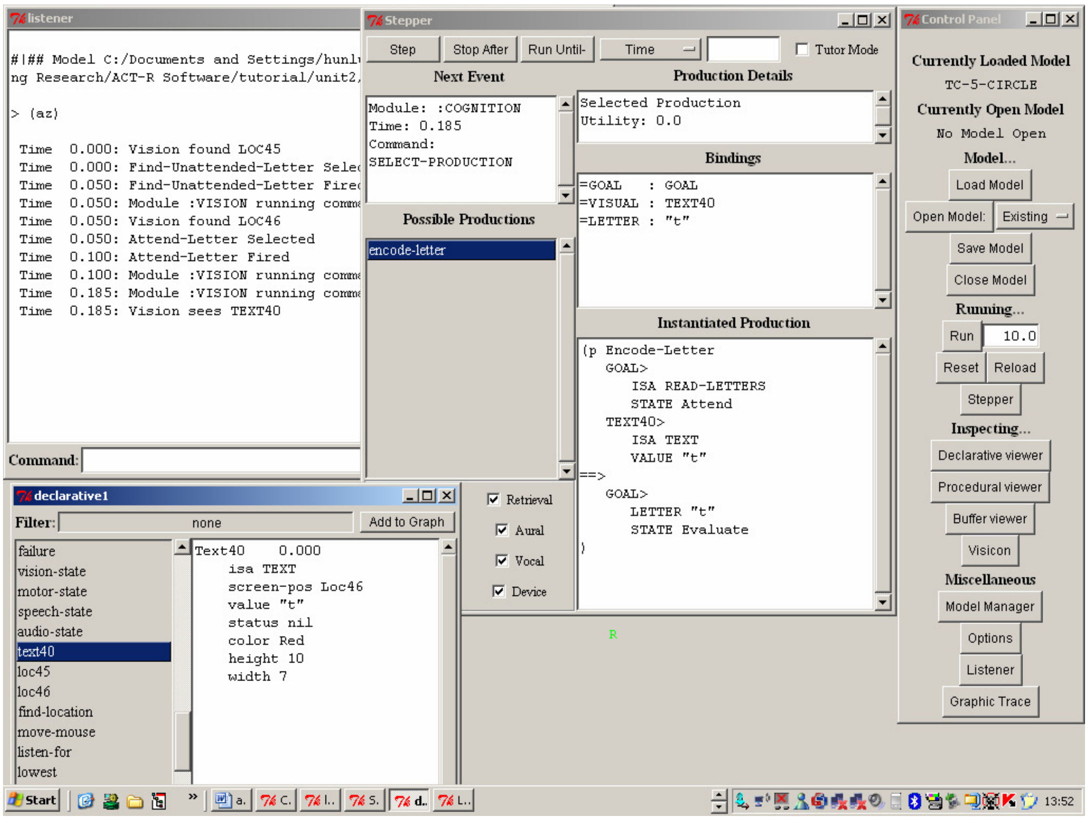
CHAPTER 3

ACT-R MODELLING

In the second phase of our research project, we built an ACT-R/PM model and compared its results with the results from our model.

We have modeled only two experiments in ACT-R. The subset chosen is Experiment 2, Two-Colors, Circle, 5 and 20 letter combinations. However, the results obtained and our discussion that follows is valid for all cases. We will first provide an explanation of the model before evaluating the results.

For the experiment, the ACT-R/PM Standalone PC version 5 is used (Figure 19). This software can be downloaded from the ACT-R site. (http://act-r.psy.cmu.edu/).



```
#|## Model C:/Documents and Settings/hunlu
ng Research/ACT-R Software/tutorial/unit2/

> (az)

 Time  0.000: Vision found LOC45
 Time  0.000: Find-Unattended-Letter Selec
 Time  0.050: Find-Unattended-Letter Firec
 Time  0.050: Module :VISION running comma
 Time  0.050: Vision found LOC46
 Time  0.050: Attend-Letter Selected
 Time  0.100: Attend-Letter Fired
 Time  0.100: Module :VISION running comma
 Time  0.185: Module :VISION running comma
 Time  0.185: Vision sees TEXT40
```

Command:

```
Module: :COGNITION
Time: 0.185
Command:
SELECT-PRODUCTION
```

encode-letter

```
Selected Production
Utility: 0.0
```

```
=GOAL    : GOAL
=VISUAL : TEXT40
=LETTER : "t"
```

```
(p Encode-Letter
   GOAL>
      ISA READ-LETTERS
      STATE Attend
   TEXT40>
      ISA TEXT
      VALUE "t"
==>
   GOAL>
      LETTER "t"
      STATE Evaluate
)
```

Currently Loaded Model

TC-5-CIRCLE

Currently Open Model

No Model Open

Run 10.0

Filter: none

failure
vision-state
motor-state
speech-state
audio-state
text40
loc45
loc46
find-location
move-mouse
listen-for
lowest

```
Text40    0.000
     isa TEXT
     screen-pos Loc46
     value "t"
     status nil
     color Red
     height 10
     width 7
```

Figure 19. ACT-R/PM Version 5 running on PC. The control panel allows models to be loaded and various debug/trace windows to be opened. The listener is where the user communicates with ACT/R giving commands to run the model to do an experiment. Stepper shows each production rule processed. The declarative window shows the contents of the declarative memory.

The Detailed Explanation of the Model and the Experiment

The experiment has some differences from the one that was presented to human subjects. ACT-R/PM's current version supports only one font and size. Therefore the letters are not exactly the same size as the other experiments. Also all colors are not supported by ACT-R/PM, therefore the basic colors of Windows are used.

The Experiment

The ACT-R/PM code consists of two parts. The first part is the experiment. Just like the experiment we had developed for human subjects, we need to develop an experiment for the ACT-R/PM model to act on. The experiment is in fact a computer program, written in LISP, a programming language commonly used in Artificial Intelligence applications. The common practice in ACT-R is to write the experiment twice, once for human subjects and once for the model and compare the results. However, based on the correspondence with the developers of the vision module in ACT-R, we were advised not to use ACT-R for precise reaction time measurements. Therefore the ACT-R experiment was only given to the model, where human subjects performed the experiment explained in the previous chapter.

```
THE EXPERIMENT

(defvar *response* nil)

{ do-experiment starts the experiment. The person or model can do the experiment
based on the value of actr-enabled-p* }
(defun do-experiment ()
  (if *actr-enabled-p*
      (do-experiment-model)
    (do-experiment-person)))

{ Person doing the experiment. This part is not used and most of the code is repeated
in the model so please refer to the do-experiment-model }
```

```
(defun do-experiment-person ()

 (let* ((lis (permute-list '("B" "A" "D" "F" "O" "H"
                              "J" "K" "L" "M" "N" "P"
                              "Q" "R" "S" "T" "V" "W"
                              "X" "Y" "Z")))
         (text1 (first lis))
         (lis2 (permute-list lis))
         (text2 (first lis2))
         (lis3 (permute-list lis))
         (text3 (first lis3))
         (lis4 (permute-list lis))
         (text4 (first lis4))
        (lis5 (permute-list lis))
         (text5 (first lis5))
         (lis6 (permute-list '("C" "G")))
         (target (first lis6))



         (window (open-exp-window "Letter Recognition" :x -5 :y -5 :width 1500
:height 900 )))
      (case (random 5)
               (0 (setf text1 target))
               (1 (setf text2 target))
               (2 (setf text3 target))
               (3 (setf text4 target))
               (4 (setf text5 target)))
    (add-text-to-exp-window :text text1 :x 350 :y 267 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text2 :x 574 :y 196 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text3 :x 711 :y 386 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text4 :x 575 :y 570 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text5 :x 350 :y 503 :color (first(permute-list
'(green red))))


    (setf *response* nil)

    (while (null *response*)
      (allow-event-manager window))

    *response*))

{ Model doing the experiment }

(defun do-experiment-model ()

{ first select 5 random letters. One of these is either a "C" (no-target or "G"
(target) }
 (let* ((lis (permute-list '("B" "A" "D" "F" "O" "H"
                              "J" "K" "L" "M" "N" "P"
                              "Q" "R" "S" "T" "V" "W"
                              "X" "Y" "Z")))
         (text1 (first lis))
         (lis2 (permute-list lis))
         (text2 (first lis2))
         (lis3 (permute-list lis))
         (text3 (first lis3))
         (lis4 (permute-list lis))
         (text4 (first lis4))
        (lis5 (permute-list lis))
         (text5 (first lis5))
{ Initially, our five letters are neither G nor C. These two letters are not in the
first list. Then we set another variable is either a G (target case) or C (no-target-
case). }

         (lis6 (permute-list '("C" "G")))
         (target (first lis6))
```

```
{ Open a blank window that covers the whole screen }

         (window (open-exp-window "Letter Recognition" :x -5 :y -5 :width 1500
:height 900 )))

{ We replace one of our five letters with G or C. At this stage we have randomly
chosen if this will be a target or no-tagret setup and located the target at a random
element of our five letter list. }

      (case (random 5)
               (0 (setf text1 target))
               (1 (setf text2 target))
               (2 (setf text3 target))
               (3 (setf text4 target))
               (4 (setf text5 target)))
{Now we show these letters on the screen on an imaginary circle. Each letter is
either blue or red. }

    (add-text-to-exp-window :text text1 :x 350 :y 267 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text2 :x 574 :y 196 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text3 :x 711 :y 386 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text4 :x 575 :y 570 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text5 :x 350 :y 503 :color (first(permute-list
'(blue red))))

        (reset)
    (pm-install-device window)
    (pm-proc-display)
{ :visual-num-finst is a parameter that tells ACT-R how many items the model can keep
track of. After that many items are attended, the model forgets if it attended the
oldest item in its list, in order to be able to attend a new item. This value is
assigned a value larger than 5 to make sure that the model can keep track of all
items on the display. This may not be a good assumption and might be improved.
However even with this assumption, which may give the model an advantage over a human
subject especially in the 20-letter case, the model is still much slower than a human
counterpart. This issue is discussed in detail in the 'discussion' chapter of the
thesis. }
    (pm-set-params :real-time t
               :visual-num-finsts 6 :visual-finst-span 10)

    (setf *response* nil)

    (pm-run 10)

    *response*))

(defmethod rpm-window-key-event-handler ((win rpm-window) key)
  (setf *response* (string key))
  (clear-exp-window)
  (when *actr-enabled-p* (pm-proc-display))) (clear-all)
(pm-reset)

(chunk-type read-letters letter state)
```

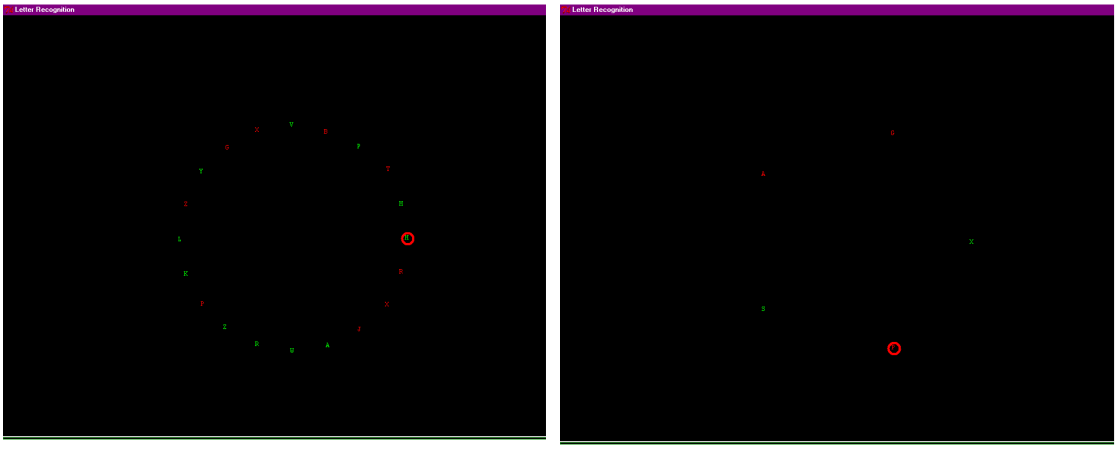


Figure 20. Example screens from ACT-R/Experiment. The model is doing the experiment and the red circle shows the current location being attended by the model. The letters are much smaller than the letters in human subjects experiment but it is not possible to control the font in current ACT-R software.

The Model

The second part of the ACT-R code consists of the model. Our model contains only production rules as no declarative knowledge is needed. Each production rule in the model is explained below. In plain English, the models tells ACT-R to find a letter that has not been attended before, attend to it and encode the attended letter to identify if it is the target or not. If it is the target the model presses ‘A’ and terminates. If it is not the target, the model shifts attention to another letter, as long as there are still unattended letters in the display. We have left it to ACT-R to decide which letter should be selected first and which letters will be attended next. It is possible to instruct ACT-R to start from a certain location (e.g. top of the screen), and shift attention to an object by giving its relative location (e.g. nearest, left, right, under etc.).

```
{ THE MODEL : OUR GOAL IS READING LETTERS }

(add-dm
 (goal isa read-letters state start))

{ IF the goal is reading letters and the state is start then find an unattended
object and set state to find location }

(P find-unattended-letter
```

```
   =goal>
      ISA         read-letters
      state       start
 ==>
   +visual-location>
      ISA         visual-location
      attended    nil
   =goal>
      state       find-location
)
{ IF the goal is reading letters and the state is find location, and the visual
module is not busy, then attend to the location (which was found in the previous step
but it was not attended...) }

(P attend-letter
   =goal>
      ISA         read-letters
      state       find-location
   =visual-location>
      ISA         visual-location
   =visual-state>
      ISA         module-state
      modality    free
==>
   +visual>
      ISA         visual-object
      screen-pos  =visual-location
   =goal>
      state       attend
)

{ IF the goal is reading letters and a letter is attended, then read this letter and
set state to evaluate if it is a target or not. }

(P encode-letter
   =goal>
      ISA         read-letters
      state       attend
   =visual>
      ISA         text
      value       =letter
==>
   =goal>
      letter      =letter
      state       evaluate
)

{ If the goal is reading letters and the state is evaluate and the letter is G, then
set state to give a 'yes' response. }

(P evaluate-letter-G
   =goal>
      ISA         read-letters
      state       evaluate
   =visual>
      ISA         text
      value       "g"
==>
   =goal>
      state       respond-yes
)

{ If the goal is reading letters and the state is evaluate and the letter is NOT G,
then set the state to decide. }

(P evaluate-letter-NotG
   =goal>
      ISA         read-letters
      state       evaluate
   =visual>
      ISA         text
     -  value       "g"
==>
   =goal>
      letter      =letter
```

```
      state       decide-on-no-g
)
{ If the goal is reading letters and the state is decide, then try to check if there
is  an unattended location and set the state to find that location.}

(P decide1
   =goal>
     ISA    read-letters
     state  decide-on-no-g
==>
   +visual-location>
     ISA   visual-location
     attended    nil
  =goal>
    state  find-location

)
{ If the goal is reading letters and we are in a state to find a location, but if
there is no such location, then respond with 'no' }

(P decide2
  =goal>
    ISA    read-letters
    state  find-location
   =visual-location>
    ISA    error
==>
   =goal>
    state respond-no
)
{ If the goal is reading letters and the state is responding with 'yes', then
activate the motor functions to press key 'A' and terminate the task. }

(P respond-found
   =goal>
      ISA         read-letters
      state       respond-yes
   =manual-state>
      ISA         module-state
      modality    free
==>
   +manual>
      ISA         press-key
      key         "a"
   =goal>
      state       stop
)
{ If the goal is reading letters and the state is responding with 'no', then activate
the motor functions to press key 'L' and terminate the task. }

(P respond-notfound
   =goal>
      ISA         read-letters
      letter      =letter
      state       respond-no
   =manual-state>
      ISA         module-state
      modality    free
==>
   +manual>
      ISA         press-key
      key         "l"
   =goal>
      state       stop
)

(sgp :v t)

(pm-set-params :real-time t :show-focus t)

(goal-focus goal)
{ Model doing the experiment. If this value is set to nil, then a human subject does
the experiment.}

(setf *actr-enabled-p* t)
```

Results and Discussion

When we run the experiment, ACT-R/PM produces the following output

```
Time  0.000: Vision found LOC45
 Time  0.000: Find-Unattended-Letter Selected
 Time  0.050: Find-Unattended-Letter Fired
 Time  0.050: Module :VISION running command FIND-LOCATION
 Time  0.050: Vision found LOC46
 Time  0.050: Attend-Letter Selected
 Time  0.100: Attend-Letter Fired
 Time  0.100: Module :VISION running command MOVE-ATTENTION
 Time  0.185: Module :VISION running command ENCODING-COMPLETE
 Time  0.185: Vision sees TEXT40
 Time  0.185: Encode-Letter Selected
 Time  0.235: Encode-Letter Fired
 Time  0.235: Evaluate-Letter-Notg Selected
 Time  0.285: Evaluate-Letter-Notg Fired
 Time  0.285: Decide1 Selected
 Time  0.335: Decide1 Fired
 Time  0.335: Module :VISION running command FIND-LOCATION
 Time  0.335: Vision found LOC48
 Time  0.335: Attend-Letter Selected
 Time  0.385: Attend-Letter Fired
 Time  0.385: Module :VISION running command MOVE-ATTENTION
 Time  0.470: Module :VISION running command ENCODING-COMPLETE
 Time  0.470: Vision sees TEXT44
 Time  0.470: Encode-Letter Selected
 Time  0.520: Encode-Letter Fired
 Time  0.520: Evaluate-Letter-Notg Selected
 Time  0.570: Evaluate-Letter-Notg Fired
......
```

Listing 1. Output from ACT-R/PM while performing the experiment. The start and end times for finding, attending and processing one letter are highlighted.

Attend-letter	50 ms
MOVE-ATTENTION	85 ms
Encode-letter	50 ms
Evaluate-letter-not-g	50 ms
Decide1	50 ms

Table 5. ACT-R/PM timings for processing one non-target item.

The time it takes to process one non-target item is 285 ms. It takes 50 ms to attend a letter, 85 ms to see attended letter and encode it into a chunk in the visual buffer, 50 ms to encode the chunk to determine what the letter actually is, 50 ms to understand it is not a target and 50 ms to decide if the model has to stop or continue.

So the time it takes for the visual search (excluding motor time) is

ST(no-target, 5-letter) = 5 ∗ 285 = 1425 ms.

ST(target, 5-letter) = 2.5 * 285 = 713 ms.

ST(no-target, 20-letter) = 20 * 285 = 5700 ms.

ST(target, 20-letter) = 10 * 285 = 2850 ms.

The motor times are not considered in the search time calculations. ACT-R/PM spends 210 ms to press a key. This number is compatible with our finding of 257 ms because our motor time delay includes attending to a certain location on the display without encoding it and then pressing a key. In ACT-R terms this is 50ms + 210 ms = 260 ms which is almost a perfect fit (50 ms is the time elapsed for the *find-unattended-letter* production rule to be fired).

The average results for target and no target conditions are compared with the results from human subjects.

	ACT/R 5-letter	**Experiment 5-letter**	**ACT-R 20 letter**	**Experiment 20-letter**
Target	713	513	2850	1111
No-target	1785	789	5700	2639

Table 6. Target and non-target search times for 5 and 20 letter displays for ACT-R and human subjects.

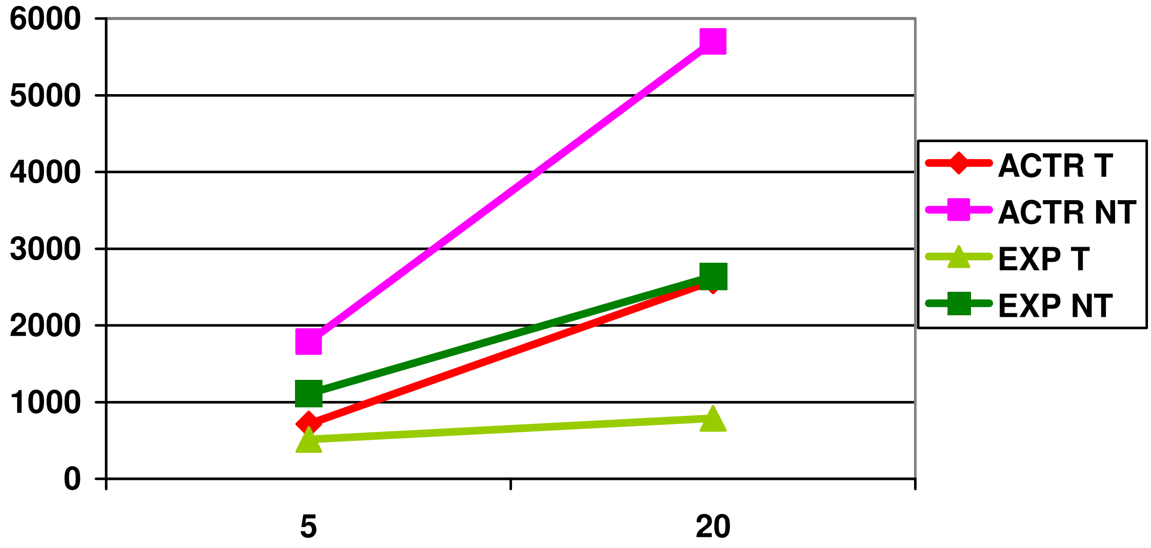


Figure 21. The comparison of experiment results and ACT-R for target (T) and non taget (NT) cases. ACT-R is much slower and the slope of increase for target cases is very different in both cases. In ACT-R the slope for target and non-target cases (increase in reaction time from 5 to 20 letter displays) are the same whereas for the experiments, the slope of the target condition is lower than the no-target case.

As it can be seen from Table 6 and Figure 21, our ACT-R/PM model is much slower than the human subjects where the real difference lies in the processing time of each letter.

Here we have used the default values provided with ACT-R/PM and did not attempt to fit it to the experiment data. It is possible to set the times for attending the letters and shifting the attention, but the default values in ACT-R are obtained from experimental data and they are expected form a good fit to human data if the model is right. Also, such manipulation would not alter the slope for target cases. In fact this is a more important question that will be addressed in the discussion.

We have not attempted to model the differences due to color and location in our ACT-R/PM model and concentrated in average search times. We can conclude that, ACT-R/PM, with this serial search approach and current parameters cannot properly model this task properly as its performance is quite different from the human subjects.

CHAPTER 4

DISCUSSION

Absence of the singleton effects

According to Theeuwes (1992) a singleton in the display attracts attention before the other items in the display and this cannot be overridden by the top-down processing, in other words by the instructions given to the subjects. In our experiments, before each experiment, subjects were briefed to ignore the color differences and look for the target letter. In the light of the Theeuwes' hypothesis, we were expecting faster response times when the target is also the singleton in the display Also when the target was not the singleton; the response times should have been negatively effected.

In the first experiment the subjects were able to ignore the singletons even if the singleton is in a more dominant color. We had also expected, based on the bottom-up hypothesis, faster response times in Experiment 2 when the target is in the color with a few (one or two) items on the display. In other words, when we examine the graphs in Appendix 2, we have expected a minimum for color1 for m=1 or 2, and likewise another minimum for color2 when m=3 or 4 (for 5 letters) and 18 or 19 (for 20 letters). However, we fail to observe these effects in a consistent manner.

Based on averages, for the 5-letter sets, the desired effects were observed in 8 out of 12 cases and this result can be due to chance factors. For 20-letter condition the results show a similar pattern. Only 7 out of 12 test conditions showed the expected results and this can also be regarded as a chance factor. Based on the

ANOVA analysis and the average response times for singleton and non-singleton cases, we cannot observe any clear effects for both 5 and 20 letter sets. It looks like the subjects are able to process the displays without any consistent distraction by the singleton. Appendix 2 presents all ANOVA values.

This finding indicates that the singleton hypothesis by Theeuwes is not valid for our experiment setup. However, these results should not be taken as evidence that the bottom-up hypothesis is totally false because the results from the color dimension support the view that items with salient features have shorter search times. Some colors yield faster search times even though the subjects are told to ignore colors. Regardless whether an item is a color singleton or not, certain colors make it faster to locate and identify the target. This finding will be discussed in a separate section.

In the first experiment, why have we failed to find the singleton effects observed in many other experiments (Bacon and Egeth, 1994; Theeuwes, 1991, 1992, 1994)? We can say that a singleton is not always a salient object. Bacon and Egeth have already shown that this might be the case. If the subject is not employing a singleton-search mode, then the singleton effects are not observed. However, like the Theeuwes (1992) and Bacon and Egeth (1994) experiments, we have told the subjects that there would be singletons, so we were expecting them to perform a singleton-search strategy due to bottom-up activation even though they were told to ignore them. From the results we can conclude that they had successfully ignored the singleton, by employing another strategy. These results might be due to feature-search mode as suggested by Bacon and Egeth. But very strong location effects observed consistently make such a suggestion very questionable.

Failure to find the singleton effects in both experiments shows that we cannot generalize Theeuwes’ results. Also Bacon and Egeth’s claims of feature-search mode vs. singleton-search mode cannot be generalized. Because if their position is correct we would have observed a singleton effect in the first experiment where the subjects knew about the irrelevant singleton.

There are other studies that may explain the results of our experiment. Gibson and Jiang (1998) pointed out that the attention capture habituates quickly to the repeated presentation of task irrelevant singletons. In our case, subjects completed a 10 display warm up routine before going into the experiment and the experiment itself is quite long, which may give enough opportunities for the subjects to habituate to the singleton effect. However, then the question of habituation should be answered. Does habituation take place in a bottom-up manner or does it occur under conscious control? So the habituation argument, although it may be used to explain the results of this experiment, does not give us any clues about which hypothesis to support.

Hortsmann (2002) claims that if a singleton is present in the display, it captures attention only if its selection is intended. In our case, as we have told the subjects that the presence of the singleton is irrelevant for their task, they have successfully masked the singleton effects.

Another option is to question the eligibility of the bottom-up / top-down paradigm. We can do this by suggesting the attention is driven by a more complex mechanism that cannot be modeled in terms of a pre-attentive and attentive stage and more or less serial deployment of attention to various parts of the display. It is deployed as a result of a complex interaction of the features on the display and the goals. This brings us to a more holistic position in examining the visual cognition.

Feature Integration Theory assumes that we are able to form feature maps for various features in parallel and then bring them together to build a deployment plan for attention. However, it does not consider that the difficulty of forming a feature map in one dimension eats up from available processing resources and leaves fewer resources for forming the maps of the other features resulting in these features to be ignored or processed to a lesser extent. The effects of one dimension on the other dimensions is discussed by Wolfe et al. (2003) .

Another possibility is suggested by Deco et al (2002). They question the current paradigm altogether and suggest that two stage mechanism with a pre-attentive and attentive stage may not be the only explanation. They provide a mathematical model of how attention is deployed. According to them the deployment planning of attention is an ongoing process where the items on the visual field are processed in parallel and in time we decide where to attend. If we have to form an analogy, it is like slowly increasing the light in a room as more features of each item become more evident over time and a constant race is going on about which part of the display will get the attention next, rather than a plan that is decided and carried out in a separate phase.

Color Effects

The results listed in Appendix 4 show that it was faster to find a target if the target is in a brighter color. Also between red and blue, blue yielded faster search times. However, an ANOVA analysis showed that, only for KY displays with 20 letters there are color effects. KY case was where the color difference was the greatest. For BR and TC cases there were no color effects.

	Exp1	**Exp2**	**Exp3**
BR-CIRCLE-5	F=2.95, p<0.88	F=0.244, p<0.622	F=1.749, p<0.190
BR-CIRCLE-20	F=0.693, p<0.405	F=3.525, p<0.061	F=0.014, p<0.90
BR-RANDOM-5	F=2.949, p<0.088	F=1.758, p<0.187	F=0.365, p<0.547
BR-RANDOM-20	F=1.148, p<0.284	**F=17.939, p<0.000**	F=3.011, p<0.084
KY-CIRCLE-5	F=0.022, p<0.882	**F=3.925, p<0.049**	N/A
KY-CIRCLE-20	**F=7.979, p<0.005**	**F=40.678, p<0.000**	N/A
KY-RANDOM-5	F=0.015, p<0.902	F=0.072, p<0.788	N/A
KY-RANDOM-20	F=0.941, p<0.332	**F=33.949, p<0.000**	N/A
TC-CIRCLE-5	F=1.629, p<0.204	F=0.014, p<0.905	F=0.053, p<0.819
TC-CIRCLE-20	F=0.830, p<0.363	F=1.605, p<0.206	F=0.001, p<0.975
TC-RANDOM-5	F=0.171, p<0.680	F=0.364, p<0.548	F=0.214, p<0.645
TC-RANDOM-20	F=0.316, p<0.575	F=0.349, p<0.555	F=1.755, p<0.186

Table 7. The ANOVA analysis for all different experiement setups for reaction time based on color.

When this result is interpreted together with the absence of a singleton-detection effect, even for the bright colors, we can conclude that it is faster and easier to process items in some colors than others. (Here one is tempted to say it is faster and easier to process salient items but this would be a circular definition as saliency is defined as “easier to spot”. Therefore we will avoid using this word.)

Item Processing Time

‘Item Process Time’ (p) can be defined as the time it takes for us to process an item well enough to be able to determine if it is a target or non-target item. It is very common to explain the differences in reaction times with the visual search strategy used. Top-down, bottom-up, FIT and other theories concentrate on how the mechanism works. However, the very same effects can be due to the difference in processing (recognition) time of various elements. It may take less time to process a bright green G than a dark red G.

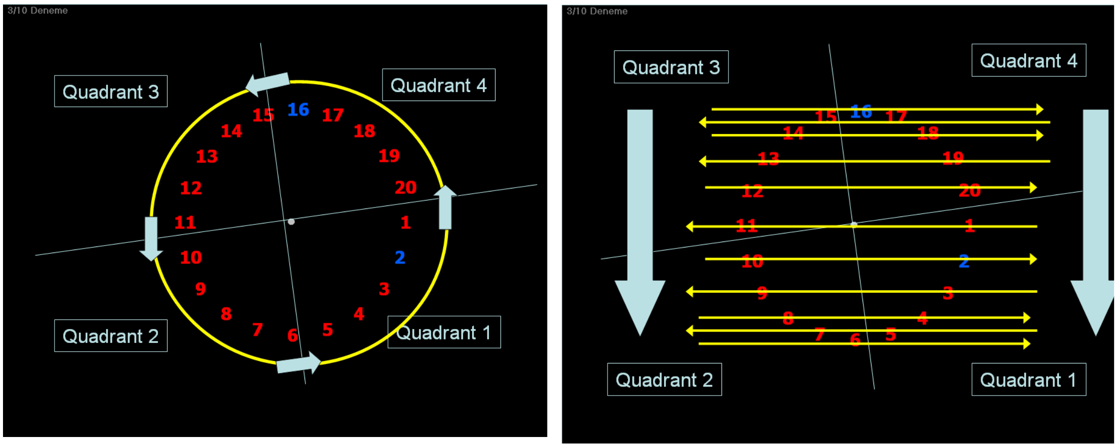
Location of the Target

The analysis of our results shows that there is a strong location effect and the subjects have a tendency to process the display from top to bottom.

The quadrant analysis in Appendix 8 shows that in all three experiments, it is faster to find the target if the target is in the upper half of the screen. This means that the subjects are first processing the upper half of the screen before the lower half. There is not much difference between the left and right hand sides of the display. The ANOVA and 95% confidence graph analysis also supports this finding. A more detailed ANOVA analysis based on each different experiment setup is provided in Appendix 8. From 32 different combinations, only 9 have $p > 0.05$. This indicates that our findings are not due to chance factors. However a closer look at the distribution of these 9 cases, we can make further observations. When the target is bright green on KY cases, our ANOVA analysis shows that all cases are subject to location effects. However, the location effects cannot be observed for none of the 'dark red target' cases for KY setups. So out of nine setups where location effects are not observed, four of them are KY/color2 (dark red) setups. The remaining five is distributed over BR and TC cases for both colors. There is a clear difference in processing bright green and dark red in KY cases. In fact, in the same display while the brighter items are being processed from top to bottom, the darker items are processed in a more or less random fashion. Considering the fact that the search times were much faster for bright green targets, we can conclude that subjects had a tendency to process bright green items before dark red items.

Based on the observed location effects, we can assume that one of the following search strategies is used.

- The subjects start from left or right of the screen and process the upper part serially continuing down to the lower part. Subjects that start from right go anti-clockwise and subjects that start from left go clockwise. This alternative seems to be too arbitrary and not very likely. Also the circular scan idea has problems in explaining the results for random layout where it is harder for the subjects to engage in a serial search.
- The subjects scan the display from top to bottom (Figure 22). If there are two or more items in the same horizontal position, then the subjects scan it either from left to right or right to left. Because when the subjects scan a horizontal line from left to right, their attention is already directed to the right hand side of the display and they process the display from right to left. Since they scan each horizontal level in a different direction, we are not able to observe any difference between the response times of right and left but since the vertical scan is from top to bottom, it is faster to find the objects on the upper half of the display.



Subjects search clockwise or anti-clockwise

Subjects search from top to bottom

Figure 22. Two different search strategies that might be employed by the subjects. The alternative on the right seems like a more likely alternative.

Both explanations have difficulties in explaining the other findings, like the faster visual search times for some colors and the absence of location effects in KY/Color2 cases. Apparently the processing order is a function of color and location.

But a general analysis of response times for each position on the screen indicate that the later hypothesis is more likely (Figure 23).

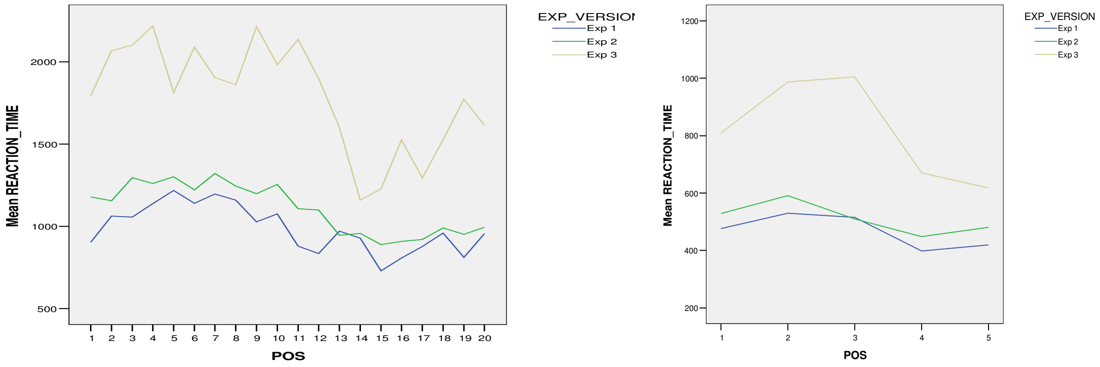


Figure 23. Average reaction times per position. The top positions (14, 15, 16) have a minimum and the positions in the upper half (11-20) have lower response times than the lower half (1-10). The results from 5 letter case, show that positions 4 and 5, which are on the upper half of the screen have lower reaction times, providing more support for the second hypothesis.

The Relation of Set Size and Search Time

In the no-target condition, in order to decide that a target is not present, the subject needs to process all elements on the display regardless of their various features. The average reaction times for all different experiment setups are presented in Appendix 2 and Appendix 3. As expected, in all experiments the reaction times for the no-target condition are higher than the target condition. In the presence of the target letter, the search terminates as soon as the target is found. The early termination of the search results in faster reaction times in the ‘target’ condition.

But an important observation can be made when we compare the reaction times in 20 letter and 5 letter set sizes for target and no-target cases. When we examine Appendix 5, we see that the reaction times for no-target cases in 20-letter layout is 3.45 times higher than the 5-letter layout. However the reaction times for target cases only are only 2.06 times higher. The rate of increase in the reaction times as the set size increases is lower if there is a target in the display. This result is very consistent over different experiment layouts. This consistency suggests that it is not a result of the different colors and layouts used in the experiment and most probably we would have obtained a similar result if we had performed the experiment with a single color.

The difference in the visual search times of 5 and 20 letter displays can be regarded as an indication of serial processing of the items in the display. Nilsen (1991) reported relation between the serial position of the target and response times (Figure 24). His research shows a linear correlation between the target position relative to the start point of the search and reaction time. If we assume serial processing or some sort of unguided search strategy, then the average serial position of the target will be n/2.

In this case the search time for the no-target condition is

$$ST(\text{No-Target}) = a + n * p$$

and the search time for target condition is

$$ST(\text{Target}) = a + n*p / 2$$

where a is a constant amount of time spent in the beginning or at the end of processing and p is the processing time for each item.

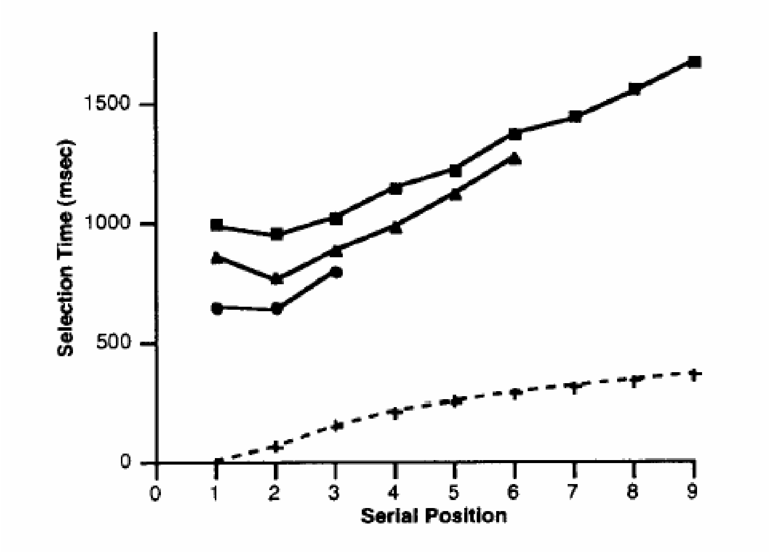


Figure 24. Nilsen's results. The search time is linearly correlated to the serial position. Here *a* is the time offset in the stating point of the lines and *p* determines the slope. The lines indicate the selection times for 3, 6 and 9 item sets.

Table 8 gives the solution for a and p values which provide a good fit for 5 and 20 letter setups in Experiment 1. There are differences in random/circular and color1/color2 conditions but they do not affect the pattern significantly.

Set Size	**a**	**p**
5-letters	199 ms	107 ms
20-letters	-602 ms	159 ms

Table 8. *a* and *p* values for Experiment 1.

As a negative value for 'a' is not meaningful, we can conclude that serial search strategy that Nilsen has described, was not used by the subjects. Either *a* or *p* or both should be different in both equations in order to fit these equations to our results.

As our model in ACT-R is also based on Nilsen's approach, like similar models in ACT-R (Byrne, Anderson, Douglas, Matessa 1999) (Anderson, Matessa, Lebiere, 1998, Chapter 5 in Atomic Components of Thought and Chapter 2 in ACT-R Tutorial), the experiment data and ACT-R data do not match.

This anomaly is caused by the assumption that we encode and recognize an item before proceeding to the next item, in other words we process items fully and serially. However, this might not be the case. For each item the subjects had to determine if it was a target or non-target. As each item has a higher probability of being a non-target item, a cost efficient search should first answer the easier question of “Could this be a non-target?”. This question is easier because many of the non-target letters have features that do not exist in our target letter G. For example the A, E, F, I, N, M letters do not have the curvature feature. The subjects can easily ignore these letters because if there is no curvature, this item has no possibility of being a target. Some subjects reported that they have specifically looked for an arrow shape as shown in Figure 25. This can be taken as evidence that the subjects find strategies to evaluate the likelihood of an element before engaging in a full encoding of the item.

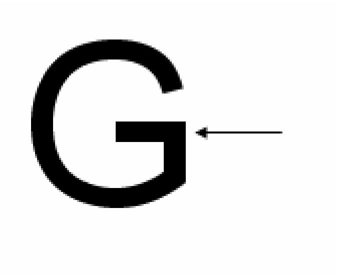


Figure 25. Some subjects reported that they were looking for the arrow like end of the letter G, in an effort to disqualify non-targets.

These considerations force us to revise the equations for visual search.

Let us now define *p* as the average time a subject spends to determine target-likelihood of an item and redefine our equations by introducing two new values. First, we will assume that when all the items are processed for likelihood and subjects fail to find the target, they might try to do one more pass over the items. They may or may not process all items once again but we assume that the time spent in this “make sure there are no-targets” phase is correlated to the set size where d_{nt} is an additional time spent for each item. d_{nt} may involve reprocessing time for some items or a non-visual decision task. Our second assumption is, once we have found an object that is likely to be the target, we process it further to make sure that it really is the target. Again, this process may occur more than once as subjects may mistakenly process letters like Q or C as possible targets. But we assume that on the average the overhead of this decision process is d_t.

$$ST(\text{No-Target}) = a + n*p + d_{nt}*n$$
$$ST(\text{Target}) = a + n*p/2 + d_t$$

With the introduction of these two variables there are an infinite number of possible solutions to these equations. But when we consider both 5 and 20 letter cases together only the following set gives a good approximation.

Variable	**Exp1 Random**	**Exp1 Circle**	**Exp2 Random**	**Exp2 Circle**	**Exp3 Random**	**Exp3 Circle**
a	121	228	168	174	709	580
p	77	62	92	67	123	129
d_{nt}	46	61	32	54	28	40
d_t	170	174	120	162	-169	-113

Table 9. Solutions for four variables in different setups.

This second set of equations, describe a totally different cognitive set of tasks involved in visual search. Our assumptions might be questionable, but our general claim that suggests that a visual search task is an elimination process rather than a brute force search holds a better chance to explain our results. ‘d_{nt}’ and ‘d_t’ factors should be further tested with different set sizes, like 10 to test the validity of these equations. Also this formulation assumes that ‘p’ is the same for all letters regardless of its features. The discussion on color differences, suggests otherwise. Also we can safely assume that processing time might be different depending on form (deciding ‘S’ is not ‘G’ might take a longer time than deciding ‘I’ is not ‘G’).

If this is the case, then how can we explain the contradiction with Nilsen’s results? Nilsen keeps the order of processing strictly under control. In this case, ‘d_{nt}‘ should be 0. However in our case, as the subjects are free to deploy their attention to any part of the screen and a sequential processing is harder due to the nature of the display, it seems plausible for them to engage in a verification phase.

Do these equations represent a more realistic model for visual search? Since the answer to this question is beyond the scope of this research project, no further tests are done to verify the claims presented above. However, it is an interesting question that deserves further research.

The anomaly of negative ‘d_t‘ values in Experiment 3 indicate that even our new equations are too simple to represent the task in this experiment. We may need to further improve them by taking into consideration that, in case there are many items that share many features with the target, the total encoding might be done more than once. So if we define P_T as the “Probability of any given item on the display has a sufficient number of common features with the target to force a full encoding” we can rewrite our equations as

$$ST(\text{No-Target}) = a + n*p + (1 - P_T)*d_{nt}*n + (P_T*n)*d_t$$
$$ST(\text{Target}) = a + n*p/2 + d_t * P_T*(n/2)$$

These equations take into consideration the items that are not the target but still encoded fully because this is the only way to determine that they are not a target.

Difference between circular and random layouts

Another interesting finding is that in the 20-letter setup, the reaction times for the target case in the random layout were slightly faster than then circular case. However this effect was not visible in non-target conditions. The results can be found in Appendix 7.

For 20-letter cases, the search times in random layout were about 10% faster than those in the circle layout. This ratio was around 12% in Experiment 2. However this effect could not be observed in Experiment 3 where the target conditions were very close and this time the circle condition is 6% faster.

A similar effect was also visible in the 5-letter case but to a lesser extent. Also the effect seems to be lower in the Multi-dimension (KY) cases.

When an ANOVA analysis is conducted on reaction time based on the display layout, for 20-letter displays we can observe a clear pattern where Experiment 3 and KY case do not show any correlation. For 5-letter displays there are no correlations.

	Exp 1	**Exp 2**	**Exp 3**
BR	**F(1,2) = 5.36** **p< 0.02**	**F(1,2) = 11.69** **p< 0.001**	F(1,2) = 1.86 p< 0.172
KY	F(1,2) = 0.80 p< 0.37	F(1,2) = 2.86 p< 0.91	N/A
TC	**F(1,2) = 14.02** **p< 0.001**	**F(1,2) = 18.49** **p< 0.001**	F(1,2) = 0.82 p< 0.775

Table 10. ANOVA results for random and circular designs fro 20-letters.

There are two important observations here

- Experiment 3 does not show any effects of display layout on the reaction time. The difference between this experiment and others is the font of the items. We have previously seen that, the font used in Experiment 3 yield longer search times. Therefore the effect of the font masks any effects that might be due to the layout.

- KY condition shows no effects as well. When we evaluate this result together with the results from the color dimension, where KY is the only setup showing some level of correlation between color and reaction time, we can conclude that the effects of color is masking the effects of the layout.

Another interesting observation is the presence of layout effects in BR and TC cases, where we had not observed any color effects. Because, the expectation was, in the absence of other effects the circular design facilitates an orderly (serial or not) search of the display where each item is processed once. However, in a random display subjects should keep track of which items are previously attended. Furthermore, in order to be faster than the circular layout, they have to do this very efficiently so that they do not process the same item twice (or do not process each item more than they do in the random case). This means they have to find a better strategy to process all the items in the display.

It is also difficult to explain why the response times for non-target condition are very close. As all the elements need to be attended in the non-target condition, the average time spent for each element is the same. This is logical but directly entails that in the presence of the target; subjects can direct their attention to the target without fully processing all items.

We can offer the following explanations for this result:

- The circular display forced the subjects to process the items serially whereas in the random display they were able to divide the display into chunks and a meta-cognitive process determined which chunk was more likely to contain the target. When there was a target, this process yielded earlier processing of the chunk with the target or faster elimination of the chunk that were not likely to contain a target. In the no-target condition, on the other hand, all letters sooner or later must be individually attended and this takes more or less the same amount of time. But, this hypothesis also has its own problems. First it assumes that the meta-cognitive stage takes a very short time. Second, it assumes that in the random case, the subject has strategies or devices to overcome double processing of the items in the display.
- For 20 letter set sizes, the subjects use the same strategy, i.e. processing in chunks in both cases. But in the random layout it is slightly easier to divide the display into chunks. In 5-letter case the letters are sparse, so instead of using the chunks, the subjects have to use a serial search. Even in the random display for 5 letters one can easily keep track of which letters are attended.

A discussion of the experiment setup and possible pitfalls in the current experimental paradigm

The experiment setup is based on a commonly used method, i.e. presenting items on a computer display and subjects used keyboard to give their replies.

Variations are used in Theeuwes (1992), Bacon and Egeth (1994), Turrato and Galfano (2000). The main difference is the usage of the capital letters of the alphabet as display objects. Theeuwes and Burger (1998) also used capital letters and some displays are very similar to the displays they used in their experiments.

In order to evaluate the other alternative, a similar version of the second experiment is conducted using the Windows Wingdings font, which is assumed to be a collection of unfamiliar and harder to process shapes. Although a much smaller subject group is used in this experiment, the results show that there is a negative impact in response times and a high degree of incompatibility in the various effects observed. Due to time and resource limitations, a full version of this experiment was not done. Therefore it remains an open question to evaluate the effects of using familiar and simple shapes like capital letters. But it is obvious that the experiment design has a great impact on the results. This makes it very questionable to design one experiment and draw general conclusions based on this experiment.

Another possible problem with the experiment setup was to use the reaction time to judge about the visual search performance, but the reaction time includes visual search time and the motor response time to press the key. There might be some interference and distortion of the results due to this factor and we can attribute the failure of the experiment to replicate the Theeuwes and Bacon-Egeth claims to this fact. However the same approach has successfully produced consistent dissociations in response time for different dimensions (like the number of letters in the display or experiment layout). This increases our confidence level in the experiment.

Number of Subjects in the Experiment

Wolfe(2000) gives a very good account of the common pitfalls in designing visual search experiments based on his experience. In his article he suggests that at least 10 subjects should be tested if each subject is having 300-400 trials or one subject should be performing thousands of trials. Based on this our current number of subjects might be considered low for judging on individual cases like the reaction time reports for individual case (Appendix 2 and 3). However, for cumulative measures the total number of trials exceeds the numbers suggested by Wolfe and might be considered more reliable. Considering the time and effort required for each subject to complete the whole set (around 2 – 2.5 hours per subject) it was very difficult to obtain more data for individual cases.

Subject Reports about the Experiment

Although no systematic interviews were conducted with the subjects, some of them voluntarily provided the following information.

Looking for an arrow shape: Two subjects, independently and without being asked, said that, they have tried to mask all the distractor effects (i.e. color differences) by looking only for the arrow-like shape in G. The possible consequences of such a strategy have been discussed above.

The location of the target and the correct key: One subject reported that he felt that he had to press the key depending on the position of the subject. If the target was in the left half of the screen he was more likely to press “A” and if the target was on the right hand side, he was more likely to press “L”, which would be an incorrect response. He said he had to think more if the target is on the right. This subject was one of two left handed subjects in the group. However, as the general results do not show any difference between the right and left hand sides of the display, we classified this as an individual case that do not effect the overall response times.

Fixation Point: One of the subjects told that the fixation point was a little distracting. As the fixation point was also on the display, it is normal that it became a part of the visual search process as well. But the fixation point was also a part of the search display in other experiments (e.g. Bacon and Egeth 1994). It is assumed that, since it is there all the time, it is not reprocessed over and over. But the fixation point is an irrelevant color singleton as well. Keeping it on the display might have been an error but no effort is done to test its effects due to time limitations. Any follow up research should address this issue as well.

CHAPTER 5

CONCLUSION

Our results showed us that there are effects of color, location, form, layout and set size on subjects' search performance. This leaves us again with a very complicated picture about the cognitive processes guiding visual attention.

We have seen that, when there are color effects, layout effects are not visible. Location effects are also influenced by the presence of color effects. But in the absence of color effects, location and layout effects can be observed. Furthermore the color effects depend on the number of items in the display. If there are more items on the display, then the color effects are more important.

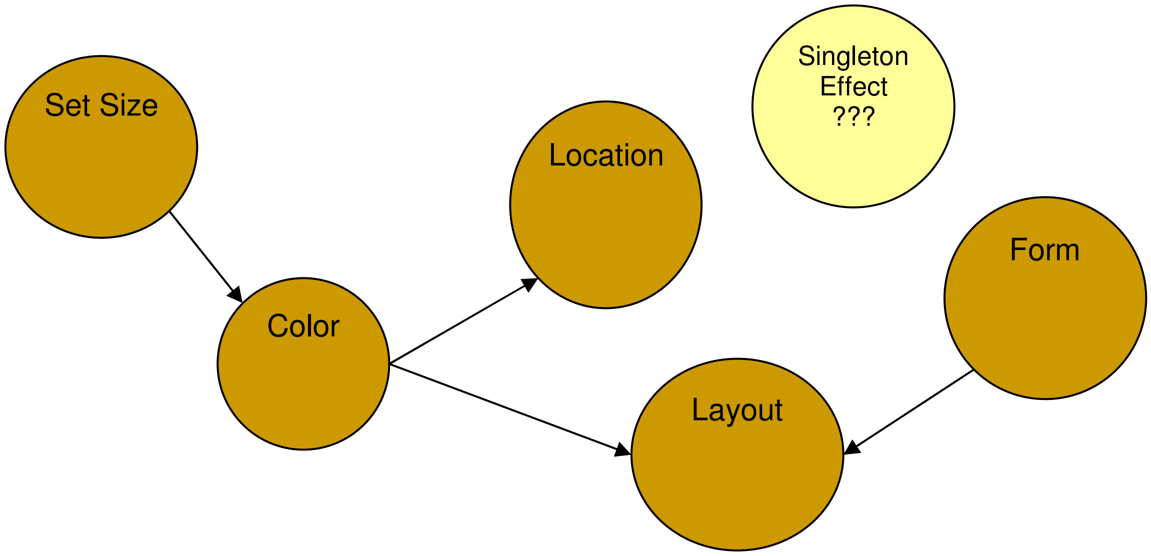


Figure 26. Which factor is influenced by which other factor? Is the presence of a singleton really a factor. Which one of these factors are under bottom-up control and which are under top*down control.

Figure 26 shows which factors are influenced by which other factors, based on the results from our experiments. We cannot say that this the complete set of factors. For example is the presence of a singleton in the display is really a factor?

Does the color have an effect on subjects’ performance for different forms? There may be color combinations, that we have not used in our experiment, but may produce such effects.

Our results indicate that the visual search is a cognitive task affected by many parameters. Therefore, the complex interaction of the different features of the display should be taken into consideration for a complete theory of visual cognition. Unless we can come up with such a cognitive model any result we may arrive from the experiments that contain only limited features will have only a limited capacity to explain visual cognition.

Bibliography

Anderson J.R. (1976). Language, memory, and thought. *Lawrence Erlbaum Associates.* Hillsdale, NJ

Anderson J.R., Lebiere, C (1998). Atomic Components of Thought. *Lawrence Erlbaum Associates,* Mahwah, NJ

Bacon, W. F. and Egeth, H. E. (1994). Overriding stimulus driven attentional capture. *Perception and Psychophysics*, 55, pp 485-496.

Broadbent, D. (1958). Perception and communication. *Oxford Pergamon.*

Byrne, M. D., Anderson J.R., Douglas S., Matessa, M. (1999). Eye tracing of the visual search of pull-down menus. *Human Factors in Computing Systems Proceedings of CHI 99*, pp 402-409.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55, pp 41-84.

Crick, F. (1984). Function of thalamic reticular complex: The Searchlight Hypothesis. *Proceedings of the National Academy of Sciences*, 81, pp 4586-4590.

Deco, G., Pollatos, O., Zihl, J. (2002). The time course of selective visual attention: theory and experiments. *Vision Research* 42, pp 2925-2945.

Helmholtz, H.v. (1867). Handbuch der psysiologischen Optik.*Voss* Leipzig.

Horstmann, G. (2002). Evidence for Attentional capture by a surprising color singleton in Visual Search. *Psychological Science*, 13, pp 499-505.

Gibson, B.S., Jiang Y. (1998). Surprise! An unexpected color singleton does not capture attention in visual search. *Psychological Science*, 9, pp 176-182.

Just M.A. Carpenter P.N. (1992). A capacity theory of comprehension: Individual differences in working-memory. *Psychological Review*, 99, pp 122-149.

Kehneman, D. (1973). Attention and Effort, *Prentice-Hall.* Englewood Cliffs, NJ

Koch C., Ullman, S. (1985). Shifts in the selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4, pp 219-227.

Laird, J.E., Newell, A., and Rosenbloom, P.S. (1987). SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, vol. 33 (1987), pp. 1-64.

Meyer, D.E., Kieras D.E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic Mechanisms. *Psychological Review*, 104, pp 3-65.

Neiser, U. (1967). Cognitive Psychology, *Appleton-Century-Crofts.* New York.

Newell, A . (1990). Unified Theories of Cognition. *Harvard University Press*. Cambridge, MA

Nilsen, E.L. (1991). Perceptual Motor control in human-computer interaction (technical report no 37) *Ann Arbor, MI : The Cognitive Science and Machine Intelligence Laboratory,* The University of Michigan.

Nothdruft, H. (2002). Attention Shift to Salient Targets. *Vision Research* 42, pp 1287-1306.

Pashler, H. (1988). Cross-dimensional interaction and texture segregation. *Perception and Psychophysics*, 43, pp 307-318.

Posner, M. I. (1980). Orienting of Attention. *Quarterly Journal of Experimental Psychology*, 32, pp 3-25.

Reisberg, D. (2001). Cognition, 2nd Ed. *W.W.Norton & Company*, New York.

Shaw, M (1978) A capacity allocation of cognitive resources to spatial locations. *Journal of Experimental Psychology: Human Perception and Performance*, 4, pp 586—598.

Sperling, G. (1980). Information Available in brief visual presentations. *Psychological Monographs* 74, pp 498.

Sperling, G. and Weichshelgartner J. (1995). Episodic theory of the dynamics of spatial attention. *Psychological Review*, 102, pp 503-532.

Treisman, A.M. and Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology,* 12, 97-136.

Treisman, A.M. (1982). Perceptual grouping and attention in visual search for features and objects. *Journal of Experimental Psychology: Human Perception and Performance*, 8, pp 194-214.

Treisman, A.M. and Sato, S (1990). Conjunction Search Revisited. *Journal of Experimental Psychology: Human Perception and Performance*, 16, pp 459-478.

Theeuwes, J. (1991) Cross-Dimensional Perceptual Selectivity. *Perception and Psychophysics*, 50, pp 184-193.

Theeuwes, J, (1992). Perceptual selectivity for color and form. *Perception and Psychophysics*, 51, pp 599-606.

Theeuwes, J. (1994). Stimulus-driven capture and attentional set: Selective search for color and visual abrupt onsets. *Journal of Experimental Psychology: Human Perception and Performance* 20, pp 799-806.

Theeuwes, J. and Burger, R. (1998). Attentional control during visual search: The effect of irrelevant singletons. *Journal of Experimental Psychology: Human Perception and Performance* 24, pp 1342-1353.

Turatto, Massimo and Galfano, Giovanni (2000). Color, form and luminance capture attention in visual search. *Vision Research* 40, pp 1639-1643.

Van Zoest, Wieske and Monk, Mieke and Theeuwes, Jan (2004). The role of stimulus-driven and goal-driven control in saccadic visual selection. *Journal of Experimental Psychology: Human Perception and Performance* 30, pp 746-759.

Van de Laar,P.,Heskes, T., and Gielen S. (1997). Task dependent learning of attention. *Neural Networks*, 10, pp 981-992.

Vierck, Esther and Miller, Jeff (2005). Direct selection by color for visual encoding. *Perception and Psychophysics*; 67, pp 483.

Wolfe, J. (1994). Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review,* 1, pp 202-238.

Wolfe, J. (2000). Visual Attention. In De Valois KK, editor, Seing 2nd ed. San Diego CA, *Academic Press*, pp 335-386.

Wolfe, J. M, Buther, S., Lee, C., and Hyle, M. (2003). Changing your mind: On the contributions of Top-down and Bottom-up Guidance in Visual Search for Feature Singletons, *Journal of Experimental Psychology: Human Perception and Performance* 29, pp 483.

Appendices

Appendix 1. The measurement for the elimination of elapsed time for motor activity

This report shows the average time spend for motor activity. The average or the averages and the standard deviation is shown at the bottom. This average is subtracted from the reaction time of each response.

Appendix 1

Motor Overhead Time Calculation Data

REACTION TIME	EXPERIMENT	TYPE	LAYOUT
211	Exp 1	BR	RANDOM
253	Exp 1	BR	RANDOM
289	Exp 1	TC	CIRCLE
249	Exp 1	TC	CIRCLE
287	Exp 2	KY	RANDOM
220	Exp 2	TC	CIRCLE
287	Exp 2	TC	CIRCLE
238	Exp 3	BR	RANDOM
279	Exp 3	TC	RANDOM

Average	**257**	Miliseconds
Std. Dev	**28,35**	

Appendix 2. Reaction times per experiment type

This shows the average reaction time for each different layout. M is the number of objects in Color1. N is the number of letters in the display Therefore there are N-M letters in Color2.

Condition 0 is no-target condition. Condition 1 means target in color1 and condition 2 means that the target was in color 2.
For each different color combination and condition, the average response times for different color/layout combinations are shown.

ANOVA analysis of each setup is provided in part 2.

Appendix 2

Visual Search Time Averages

Exp 1 **N= 5**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
1	0	668	779	829	695	667	738
4	0	779	693	764	680	694	839
1	1	453	414	393	494	452	583
4	1	491	514	386	439	456	470
1	2	482	456	429	535	462	494
4	2	361	446	666	415	337	461

Exp 1 **N= 20**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
1	0	2377	2537	2443	2427	2754	2856
19	0	2356	2553	2596	2515	2777	2628
1	1	791	885	692	977	1367	877
19	1	950	1100	884	1022	894	1054
1	2	880	1016	825	988	978	1175
19	2	1286	1274	689	841	1017	1684

Exp 2 **N= 5**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
0	0	693	764	833	730	795	896
1	0	795	763	789	711	877	906
2	0	792	745	857	689	906	807
3	0	732	690	746	715	793	891
4	0	648	739	725	777	796	958
5	0	738	806	720	809	843	839
1	1	437	500	490	508	631	511
2	1	381	573	406	470	525	520
3	1	457	461	498	462	503	469
4	1	485	526	471	471	623	491
5	1	482	594	411	530	557	483
0	2	442	500	437	488	570	498
1	2	517	517	487	485	614	595
2	2	442	519	439	518	536	603
3	2	502	552	612	485	605	574
4	2	444	584	525	556	584	525

Exp 2 **N= 20**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
0	0	2538	2831	2243	2599	2862	2575
1	0	2835	2766	2495	2410	2668	2889
2	0	2781	2860	2643	2640	2630	2780
3	0	2661	2548	2612	2489	2622	2740
4	0	2400	2813	2563	2426	2724	2903
5	0	2741	2646	2565	2450	2769	2755
6	0	2823	2739	2622	2602	2709	2648
7	0	2687	2588	2402	2528	2646	2820

Appendix 2

Visual Search Time Averages

8	0	2553	2870	2748	2292	2612	2639
9	0	2683	2930	2481	2389	2613	2772
10	0	2477	2623	2761	2361	2631	2444
11	0	2725	2609	2629	2568	2728	2505
12	0	2404	3001	2447	2748	2915	2709
13	0	2422	2801	2516	2543	2818	2702
14	0	2501	2825	2606	2537	2649	2748
15	0	2702	2718	2408	2796	2484	2669
16	0	2758	2551	2506	2568	2436	2725
17	0	2875	2699	2421	2667	2645	2466
18	0	2548	2549	2518	2615	2674	2866
19	0	2925	2775	2445	2730	2478	2617
20	0	2678	2677	2373	2412	2444	2717
1	1	893	1448	716	1135	1100	1016
2	1	1123	1516	846	1066	965	883
3	1	885	1533	710	1144	740	842
4	1	1019	1131	854	863	1008	1119
5	1	1271	1258	945	817	900	1111
6	1	1032	1508	844	1109	709	959
7	1	1046	1625	851	921	771	1114
8	1	1139	1112	918	1294	875	762
9	1	1026	1271	776	979	723	844
10	1	775	1514	986	1459	847	1144
11	1	1010	1267	841	1472	979	946
12	1	992	1296	871	1346	763	1035
13	1	997	1115	677	1247	1066	891
14	1	1048	1238	794	1000	796	978
15	1	1070	1088	981	1197	884	1086
16	1	1193	1289	734	1422	1008	716
17	1	1040	1321	913	991	1221	953
18	1	940	1282	797	1227	965	1263
19	1	1287	1493	845	957	1039	1023
20	1	1125	1273	840	942	766	1161
0	2	984	1205	829	1081	1064	1003
1	2	727	1424	863	1186	1117	1401
2	2	936	1171	1074	1294	1670	1332
3	2	1003	1021	1022	1129	1007	1228
4	2	883	1094	1012	1174	1025	1202
5	2	733	1338	841	1128	1041	1386
6	2	1113	1068	959	1103	1193	1286
7	2	844	1181	1148	1391	841	1055
8	2	1218	1445	1337	1464	1060	1395
9	2	904	1412	1151	1254	1136	1106
10	2	1333	1245	789	1075	1018	1434
11	2	912	1181	1080	1237	1110	1213
12	2	1018	1137	1527	1083	1384	1512
13	2	1121	1173	1005	1412	1253	1745
14	2	829	1461	999	1316	1341	1288
15	2	1162	1233	959	1216	1128	1052
16	2	1129	1442	1197	1477	1324	1487
17	2	1166	1221	1024	1651	1653	1497
18	2	1084	1398	1056	1062	1224	1481
19	2	1231	1281	1248	1287	1676	1146

Appendix 2

Visual Search Time Averages

Exp 3 **N= 5**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
0	0	1445	1370	1277	1371	0	0
1	0	1256	1297	1204	1481	0	0
2	0	1849	1525	1441	1559	0	0
3	0	1409	1296	1183	1278	0	0
4	0	1409	1511	1273	1440	0	0
5	0	1443	1302	1197	1280	0	0
1	1	887	1161	628	732	0	0
2	1	677	874	792	937	0	0
3	1	953	867	640	938	0	0
4	1	783	753	837	720	0	0
5	1	693	730	688	961	0	0
0	2	632	933	844	880	0	0
1	2	962	750	668	561	0	0
2	2	648	754	929	595	0	0
3	2	741	1026	791	727	0	0
4	2	785	1046	598	888	0	0

Exp 3 **N= 20**

M	Cond.	TC RANDOM	TC CIRCLE	BR RANDOM	BR CIRCLE	KY RANDOM	KY CIRCLE
0	0	4076	3662	4176	3560	0	0
1	0	4164	3911	4083	4207	0	0
2	0	4163	3603	4027	3544	0	0
3	0	3883	3720	3759	3938	0	0
4	0	3846	3989	4061	3850	0	0
5	0	3728	3794	3935	3901	0	0
6	0	3581	3746	4371	3728	0	0
7	0	3766	3963	3770	3997	0	0
8	0	3846	3972	4336	4182	0	0
9	0	3571	3590	4263	3792	0	0
10	0	3785	3973	3949	3673	0	0
11	0	3647	3601	3727	3819	0	0
12	0	3651	3244	3930	4358	0	0
13	0	3580	3787	3855	3918	0	0
14	0	3884	3968	3915	3876	0	0
15	0	3891	3780	4060	3912	0	0
16	0	3973	3981	4074	4020	0	0
17	0	3647	3620	3843	3666	0	0
18	0	4138	3479	4614	3540	0	0
19	0	3692	4103	3981	4167	0	0
20	0	3658	3467	4105	3607	0	0
1	1	1511	1921	1709	1103	0	0
2	1	1802	1929	1952	1393	0	0
3	1	2219	2111	2148	1487	0	0
4	1	1698	2532	2328	2248	0	0
5	1	1377	2241	1762	1506	0	0
6	1	1060	1422	2181	1649	0	0
7	1	2588	1353	1692	2012	0	0
8	1	1591	2028	2269	1894	0	0
9	1	1916	1486	2418	2310	0	0

Appendix 2

Visual Search Time Averages

10	1	1265	1427	1860	2169	0	0
11	1	1736	1776	1989	1801	0	0
12	1	1437	1582	906	1914	0	0
13	1	1313	984	2290	1653	0	0
14	1	1643	2080	2282	1683	0	0
15	1	1087	1424	1031	1672	0	0
16	1	1353	2290	1730	1266	0	0
17	1	1895	2034	1616	1717	0	0
18	1	1558	1610	1830	1638	0	0
19	1	1748	2177	2336	2252	0	0
20	1	1617	1924	2857	1566	0	0
0	2	1476	2443	1968	1664	0	0
1	2	1324	1289	1772	1564	0	0
2	2	1218	1733	1865	1947	0	0
3	2	1640	1472	1042	1863	0	0
4	2	1215	1678	1419	2009	0	0
5	2	1398	1800	2439	1605	0	0
6	2	1330	1383	1952	1550	0	0
7	2	1744	2005	1562	1961	0	0
8	2	1789	1772	1267	1951	0	0
9	2	1989	2209	1891	1437	0	0
10	2	1762	2254	1618	1617	0	0
11	2	1528	1420	1512	2228	0	0
12	2	2428	1653	1846	1212	0	0
13	2	1408	1558	1707	1664	0	0
14	2	1798	1138	1097	1983	0	0
15	2	2393	2281	2427	2123	0	0
16	2	2766	2567	1523	1782	0	0
17	2	2970	2052	1961	1622	0	0
18	2	2056	2190	1626	1992	0	0
19	2	1526	1607	2072	1443	0	0

Appendix 2 - Part 2

Overall averages

Experiment	Type	Layout	Set size	Target Color (0=No-Target	Avg Time
Exp 1					
Exp 1	BR	CIRCLE	5	0	687
Exp 1	BR	CIRCLE	5	1	445
Exp 1	BR	CIRCLE	5	2	520
Exp 1	BR	CIRCLE	20	0	2467
Exp 1	BR	CIRCLE	20	1	1020
Exp 1	BR	CIRCLE	20	2	975
Exp 1	BR	RANDOM	5	0	796
Exp 1	BR	RANDOM	5	1	387
Exp 1	BR	RANDOM	5	2	446
Exp 1	BR	RANDOM	20	0	2518
Exp 1	BR	RANDOM	20	1	870
Exp 1	BR	RANDOM	20	2	814
Exp 1	KY	CIRCLE	5	0	777
Exp 1	KY	CIRCLE	5	1	484
Exp 1	KY	CIRCLE	5	2	490
Exp 1	KY	CIRCLE	20	0	2739
Exp 1	KY	CIRCLE	20	1	1047
Exp 1	KY	CIRCLE	20	2	1211
Exp 1	KY	RANDOM	5	0	679
Exp 1	KY	RANDOM	5	1	455
Exp 1	KY	RANDOM	5	2	452
Exp 1	KY	RANDOM	20	0	2766
Exp 1	KY	RANDOM	20	1	930
Exp 1	KY	RANDOM	20	2	981
Exp 1	TC	CIRCLE	5	0	734
Exp 1	TC	CIRCLE	5	1	504
Exp 1	TC	CIRCLE	5	2	455
Exp 1	TC	CIRCLE	20	0	2545
Exp 1	TC	CIRCLE	20	1	1090
Exp 1	TC	CIRCLE	20	2	1036
Exp 1	TC	RANDOM	5	0	729
Exp 1	TC	RANDOM	5	1	488
Exp 1	TC	RANDOM	5	2	472
Exp 1	TC	RANDOM	20	0	2368
Exp 1	TC	RANDOM	20	1	942
Exp 1	TC	RANDOM	20	2	910
Exp 2					
Exp 2	BR	CIRCLE	5	0	738
Exp 2	BR	CIRCLE	5	1	488
Exp 2	BR	CIRCLE	5	2	507
Exp 2	BR	CIRCLE	20	0	2542
Exp 2	BR	CIRCLE	20	1	1131
Exp 2	BR	CIRCLE	20	2	1250

Appendix 2 - Part 2

Overall averages

Experiment	Type	Layout	Set size	Target Color (0=No-Target	Avg Time
Exp 2	BR	RANDOM	5	0	778
Exp 2	BR	RANDOM	5	1	455
Exp 2	BR	RANDOM	5	2	502
Exp 2	BR	RANDOM	20	0	2523
Exp 2	BR	RANDOM	20	1	838
Exp 2	BR	RANDOM	20	2	1053
Exp 2	KY	CIRCLE	5	0	883
Exp 2	KY	CIRCLE	5	1	495
Exp 2	KY	CIRCLE	5	2	560
Exp 2	KY	CIRCLE	20	0	2702
Exp 2	KY	CIRCLE	20	1	993
Exp 2	KY	CIRCLE	20	2	1312
Exp 2	KY	RANDOM	5	0	835
Exp 2	KY	RANDOM	5	1	569
Exp 2	KY	RANDOM	5	2	582
Exp 2	KY	RANDOM	20	0	2654
Exp 2	KY	RANDOM	20	1	905
Exp 2	KY	RANDOM	20	2	1210
Exp 2	TC	CIRCLE	5	0	751
Exp 2	TC	CIRCLE	5	1	529
Exp 2	TC	CIRCLE	5	2	534
Exp 2	TC	CIRCLE	20	0	2736
Exp 2	TC	CIRCLE	20	1	1329
Exp 2	TC	CIRCLE	20	2	1257
Exp 2	TC	RANDOM	5	0	735
Exp 2	TC	RANDOM	5	1	451
Exp 2	TC	RANDOM	5	2	470
Exp 2	TC	RANDOM	20	0	2648
Exp 2	TC	RANDOM	20	1	1046
Exp 2	TC	RANDOM	20	2	1013
Exp 3					
Exp 3	BR	CIRCLE	5	0	1402
Exp 3	BR	CIRCLE	5	1	860
Exp 3	BR	CIRCLE	5	2	730
Exp 3	BR	CIRCLE	20	0	3868
Exp 3	BR	CIRCLE	20	1	1752
Exp 3	BR	CIRCLE	20	2	1764
Exp 3	BR	RANDOM	5	0	1262
Exp 3	BR	RANDOM	5	1	715
Exp 3	BR	RANDOM	5	2	770
Exp 3	BR	RANDOM	20	0	4040
Exp 3	BR	RANDOM	20	1	1950
Exp 3	BR	RANDOM	20	2	1738
Exp 3	TC	CIRCLE	5	0	1385
Exp 3	TC	CIRCLE	5	1	880
Exp 3	TC	CIRCLE	5	2	901
Exp 3	TC	CIRCLE	20	0	3760

Appendix 2 - Part 2

Overall averages

Experiment	Type	Layout	Set size	Target Color (0=No-Target	Avg Time
Exp 3	TC	CIRCLE	20	1	1818
Exp 3	TC	CIRCLE	20	2	1814
Exp 3	TC	RANDOM	5	0	1469
Exp 3	TC	RANDOM	5	1	797
Exp 3	TC	RANDOM	5	2	753
Exp 3	TC	RANDOM	20	0	3818
Exp 3	TC	RANDOM	20	1	1612
Exp 3	TC	RANDOM	20	2	1771

Appendix 2 – Part 3 : **ANOVA analysis based on the number of items in each color and reaction times.**

TYPE = BR, CIRCLE, 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	24178,690	1	24178,690	,545	,463
Within Groups	3460114,197	78	44360,438		
Total	3484292,888	79			

a TYPE = BR, CIRCLE , 5, Exp 1 , COLOR = 1

TYPE = BR, CIRCLE , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	124692,958	1	124692,958	1,164	,284
Within Groups	8245457,472	77	107083,863		
Total	8370150,430	78			

a TYPE = BR, CIRCLE , 5, Exp 1 , COLOR = 2

TYPE = BR, CIRCLE , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	51230,511	4	12807,628	,317	,866
Within Groups	2747372,010	68	40402,530		
Total	2798602,521	72			

a TYPE = BR, CIRCLE , 5, Exp 2 , COLOR = 1

TYPE = BR, CIRCLE , 5, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	58221,920	4	14555,480	,223	,925
Within Groups	4572987,867	70	65328,398		
Total	4631209,787	74			

a TYPE = BR, CIRCLE , 5, Exp 2 , COLOR = 2

TYPE = BR, CIRCLE , 5, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	422818,988	4	105704,747	,379	,822
Within Groups	8639462,012	31	278692,323		
Total	9062281,000	35			

a TYPE = BR, CIRCLE , 5, Exp 3 , COLOR = 1

TYPE = BR, CIRCLE , 5, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	754646,100	4	188661,525	1,770	,157
Within Groups	3731405,000	35	106611,571		
Total	4486051,100	39			

a TYPE = BR, CIRCLE , 5, Exp 3 , COLOR = 2

TYPE = BR, CIRCLE , 20, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	27906,990	1	27906,990	,071	,791
Within Groups	100361803,912	254	395125,212		
Total	100389710,902	255			

a TYPE = BR, CIRCLE , 20, Exp 1 , COLOR = 1

TYPE = BR, CIRCLE , 20, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	455295,091	1	455295,091	1,257	,263
Within Groups	98150770,828	271	362179,966		
Total	98606065,919	272			

a TYPE = BR, CIRCLE , 20, Exp 1 , COLOR = 2

TYPE = BR, CIRCLE , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9428832,629	19	496254,349	1,179	,276
Within Groups	98913692,642	235	420909,330		
Total	108342525,271	254			

a TYPE = BR, CIRCLE , 20, Exp 2 , COLOR = 1

TYPE = BR, CIRCLE , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5904617,327	19	310769,333	,513	,955
Within Groups	135010611,348	223	605428,750		
Total	140915228,675	242			

a TYPE = BR, CIRCLE , 20, Exp 2 , COLOR = 2

TYPE = BR, CIRCLE , 20, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	19708608,224	19	1037295,170	,955	,516
Within Groups	196533021,726	181	1085817,800		
Total	216241629,950	200			

a TYPE = BR, CIRCLE , 20, Exp 3 , COLOR = 1

TYPE = BR, CIRCLE , 20, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	13890029,600	19	731054,189	,826	,674
Within Groups	170795481,198	193	884950,680		
Total	184685510,798	212			

a TYPE = BR, CIRCLE , 20, Exp 3 , COLOR = 2

TYPE = BR, RANDOM , 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	486,799	1	486,799	,013	,910
Within Groups	2887022,073	76	37987,133		
Total	2887508,872	77			

a TYPE = BR, RANDOM , 5, Exp 1 , COLOR = 1

TYPE = BR, RANDOM , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	259960,134	1	259960,134	5,355	,024
Within Groups	3349456,739	69	48542,851		
Total	3609416,873	70			

a TYPE = BR, RANDOM , 5, Exp 1 , COLOR = 2

TYPE = BR, RANDOM , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	123719,825	4	30929,956	,630	,643
Within Groups	3684984,125	75	49133,122		
Total	3808703,950	79			

a TYPE = BR, RANDOM , 5, Exp 2 , COLOR = 1

TYPE = BR, RANDOM , 5, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	327755,223	4	81938,806	1,730	,153
Within Groups	3410552,049	72	47368,778		
Total	3738307,273	76			

a TYPE = BR, RANDOM , 5, Exp 2 , COLOR = 2

TYPE = BR, RANDOM , 5, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	308032,365	4	77008,091	,391	,814
Within Groups	7688138,431	39	197131,755		
Total	7996170,795	43			

a TYPE = BR, RANDOM , 5, Exp 3 , COLOR = 1

TYPE = BR, RANDOM , 5, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	611379,678	4	152844,920	,898	,475
Within Groups	6640484,208	39	170268,826		
Total	7251863,886	43			

a TYPE = BR, RANDOM , 5, Exp 3 , COLOR = 2

TYPE = BR, RANDOM , 20, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	580270,291	1	580270,291	1,378	,242
Within Groups	98922391,945	235	420946,349		
Total	99502662,236	236			

a TYPE = BR, RANDOM , 20, Exp 1 , COLOR = 1

TYPE = BR, RANDOM , 20, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	374489,960	1	374489,960	1,423	,234
Within Groups	70517310,192	268	263124,292		
Total	70891800,152	269			

a TYPE = BR, RANDOM , 20, Exp 1 , COLOR = 2

TYPE = BR, RANDOM , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2095706,617	19	110300,348	,414	,987
Within Groups	72771471,253	273	266562,166		
Total	74867177,870	292			

a TYPE = BR, RANDOM , 20, Exp 2 , COLOR = 1

TYPE = BR, RANDOM , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8540657,952	19	449508,313	,907	,575
Within Groups	128852915,819	260	495588,138		
Total	137393573,771	279			

a TYPE = BR, RANDOM , 20, Exp 2 , COLOR = 2

TYPE = BR, RANDOM , 20, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	37830533,838	19	1991080,728	1,238	,233
Within Groups	260499023,881	162	1608018,666		
Total	298329557,720	181			

a TYPE = BR, RANDOM , 20, Exp 3 , COLOR = 1

TYPE = BR, RANDOM , 20, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	24167006,665	19	1271947,719	1,130	,325
Within Groups	192411544,581	171	1125213,711		
Total	216578551,246	190			

a TYPE = BR, RANDOM , 20, Exp 3 , COLOR = 2

TYPE = KY, CIRCLE , 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	123636,365	1	123636,365	2,251	,137
Within Groups	4559257,047	83	54930,808		
Total	4682893,412	84			

a TYPE = KY, CIRCLE , 5, Exp 1 , COLOR = 1

TYPE = KY, CIRCLE , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9475,240	1	9475,240	,203	,654
Within Groups	3548654,709	76	46692,825		
Total	3558129,949	77			

a TYPE = KY, CIRCLE , 5, Exp 1 , COLOR = 2

TYPE = KY, CIRCLE , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	29025,129	4	7256,282	,121	,974
Within Groups	4780707,647	80	59758,846		
Total	4809732,776	84			

a TYPE = KY, CIRCLE , 5, Exp 2 , COLOR = 1

TYPE = KY, CIRCLE   , 5, Exp 2   , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	129269,548	4	32317,387	1,018	,404
Within Groups	2349962,225	74	31756,246		
Total	2479231,772	78			

a TYPE = KY, CIRCLE   , 5, Exp 2   , COLOR = 2

TYPE = KY, CIRCLE   , 20, Exp 1   , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	331351,171	1	331351,171	,833	,362
Within Groups	101477513,218	255	397951,032		
Total	101808864,389	256			

a TYPE = KY, CIRCLE   , 20, Exp 1   , COLOR = 1

TYPE = KY, CIRCLE   , 20, Exp 1   , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4816042,910	1	4816042,910	9,796	,002
Within Groups	138147307,790	281	491627,430		
Total	142963350,700	282			

a TYPE = KY, CIRCLE   , 20, Exp 1   , COLOR = 2

TYPE = KY, CIRCLE , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6692943,613	19	352260,190	,987	,476
Within Groups	118520736,160	332	356990,169		
Total	125213679,773	351			

a TYPE = KY, CIRCLE , 20, Exp 2 , COLOR = 1

TYPE = KY, CIRCLE , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	10889474,974	19	573130,262	1,184	,270
Within Groups	146715141,856	303	484208,389		
Total	157604616,830	322			

a TYPE = KY, CIRCLE , 20, Exp 2 , COLOR = 2

TYPE = KY, RANDOM , 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	83,217	1	83,217	,002	,966
Within Groups	3955560,072	88	44949,546		
Total	3955643,289	89			

a TYPE = KY, RANDOM , 5, Exp 1 , COLOR = 1

TYPE = KY, RANDOM , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	99903,424	1	99903,424	2,629	,109
Within Groups	3191932,076	84	37999,191		
Total	3291835,500	85			

a TYPE = KY, RANDOM , 5, Exp 1 , COLOR = 2

TYPE = KY, RANDOM , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	222444,332	4	55611,083	,644	,633
Within Groups	6737918,463	78	86383,570		
Total	6960362,795	82			

a TYPE = KY, RANDOM , 5, Exp 2 , COLOR = 1

TYPE = KY, RANDOM , 5, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	63283,572	4	15820,893	,152	,962
Within Groups	8131690,669	78	104252,444		
Total	8194974,241	82			

a TYPE = KY, RANDOM , 5, Exp 2 , COLOR = 2

TYPE = KY, RANDOM , 20, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4969200,641	1	4969200,641	11,885	,001
Within Groups	129607804,663	310	418089,692		
Total	134577005,305	311			

a TYPE = KY, RANDOM , 20, Exp 1 , COLOR = 1

TYPE = KY, RANDOM , 20, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	30404,471	1	30404,471	,074	,785
Within Groups	123726348,139	303	408337,783		
Total	123756752,610	304			

a TYPE = KY, RANDOM , 20, Exp 1 , COLOR = 2

TYPE = KY, RANDOM , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5619898,226	19	295784,117	1,041	,414
Within Groups	77597705,576	273	284240,680		
Total	83217603,802	292			

a TYPE = KY, RANDOM , 20, Exp 2 , COLOR = 1

TYPE = KY, RANDOM , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	14590447,683	19	767918,299	1,572	,063
Within Groups	128946430,299	264	488433,448		
Total	143536877,982	283			

a TYPE = KY, RANDOM , 20, Exp 2 , COLOR = 2

TYPE = TC, CIRCLE , 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	62460,000	1	62460,000	,781	,380
Within Groups	5917130,671	74	79961,225		
Total	5979590,671	75			

a TYPE = TC, CIRCLE , 5, Exp 1 , COLOR = 1

TYPE = TC, CIRCLE , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	875,940	1	875,940	,019	,890
Within Groups	3925949,049	86	45650,570		
Total	3926824,989	87			

a TYPE = TC, CIRCLE , 5, Exp 1 , COLOR = 2

TYPE = TC, CIRCLE , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	189662,222	4	47415,556	,718	,582
Within Groups	5084938,997	77	66038,169		
Total	5274601,220	81			

a TYPE = TC, CIRCLE , 5, Exp 2 , COLOR = 1

TYPE = TC, CIRCLE , 5, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	73912,624	4	18478,156	,286	,886
Within Groups	5098458,614	79	64537,451		
Total	5172371,238	83			

a TYPE = TC, CIRCLE , 5, Exp 2 , COLOR = 2

TYPE = TC, CIRCLE , 5, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1036190,619	4	259047,655	1,257	,304
Within Groups	7832725,986	38	206124,368		
Total	8868916,605	42			

a TYPE = TC, CIRCLE , 5, Exp 3 , COLOR = 1

TYPE = TC, CIRCLE , 5, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	695707,980	4	173926,995	1,360	,266
Within Groups	4730365,639	37	127847,720		
Total	5426073,619	41			

a TYPE = TC, CIRCLE , 5, Exp 3 , COLOR = 2

TYPE = TC, CIRCLE , 20, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	663063,237	1	663063,237	1,103	,294
Within Groups	188700865,911	314	600958,172		
Total	189363929,149	315			

a TYPE = TC, CIRCLE , 20, Exp 1 , COLOR = 1

TYPE = TC, CIRCLE , 20, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1412371,508	1	1412371,508	3,093	,080
Within Groups	135600385,924	297	456566,956		
Total	137012757,431	298			

a TYPE = TC, CIRCLE , 20, Exp 1 , COLOR = 2

TYPE = TC, CIRCLE , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	7624234,521	19	401275,501	,735	,782
Within Groups	160496400,221	294	545906,123		
Total	168120634,742	313			

a TYPE = TC, CIRCLE , 20, Exp 2 , COLOR = 1

TYPE = TC, CIRCLE , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5524730,110	19	290775,269	,592	,912
Within Groups	143501885,275	292	491444,813		
Total	149026615,385	311			

a TYPE = TC, CIRCLE , 20, Exp 2 , COLOR = 2

TYPE = TC, CIRCLE , 20, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	22474250,517	19	1182855,290	,935	,542
Within Groups	169568219,119	134	1265434,471		
Total	192042469,636	153			

a TYPE = TC, CIRCLE , 20, Exp 3 , COLOR = 1

TYPE = TC, CIRCLE , 20, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	22970230,777	19	1208959,515	,856	,636
Within Groups	180681526,649	128	1411574,427		
Total	203651757,426	147			

a TYPE = TC, CIRCLE , 20, Exp 3 , COLOR = 2

TYPE = TC, RANDOM , 5, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	7889,038	1	7889,038	,093	,761
Within Groups	6546713,114	77	85022,248		
Total	6554602,152	78			

a TYPE = TC, RANDOM , 5, Exp 1 , COLOR = 1

TYPE = TC, RANDOM , 5, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	93587,533	1	93587,533	2,256	,137
Within Groups	3153455,147	76	41492,831		
Total	3247042,679	77			

a TYPE = TC, RANDOM , 5, Exp 1 , COLOR = 2

TYPE = TC, RANDOM , 5, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	94155,616	4	23538,904	,889	,475
Within Groups	1747137,933	66	26471,787		
Total	1841293,549	70			

a TYPE = TC, RANDOM , 5, Exp 2 , COLOR = 1

TYPE = TC, RANDOM , 5, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	80627,249	4	20156,812	,471	,757
Within Groups	2910407,381	68	42800,109		
Total	2991034,630	72			

a TYPE = TC, RANDOM , 5, Exp 2 , COLOR = 2

TYPE = TC, RANDOM , 5, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	512485,852	4	128121,463	,518	,723
Within Groups	9653376,875	39	247522,484		
Total	10165862,727	43			

a TYPE = TC, RANDOM , 5, Exp 3 , COLOR = 1

TYPE = TC, RANDOM , 5, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	633202,293	4	158300,573	,989	,425
Within Groups	6240438,889	39	160011,254		
Total	6873641,182	43			

a TYPE = TC, RANDOM , 5, Exp 3 , COLOR = 2

TYPE = TC, RANDOM , 20, Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	287785,173	1	287785,173	,580	,447
Within Groups	120144326,348	242	496464,158		
Total	120432111,521	243			

a TYPE = TC, RANDOM , 20, Exp 1 , COLOR = 1

TYPE = TC, RANDOM , 20, Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2901219,946	1	2901219,946	8,547	,004
Within Groups	88936568,645	262	339452,552		
Total	91837788,591	263			

a TYPE = TC, RANDOM , 20, Exp 1 , COLOR = 2

TYPE = TC, RANDOM , 20, Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4120174,939	19	216851,313	,467	,973
Within Groups	123410217,928	266	463948,188		
Total	127530392,867	285			

a TYPE = TC, RANDOM , 20, Exp 2 , COLOR = 1

TYPE = TC, RANDOM , 20, Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	7753454,387	19	408076,547	,988	,475
Within Groups	109858788,064	266	413002,963		
Total	117612242,451	285			

a TYPE = TC, RANDOM , 20, Exp 2 , COLOR = 2

TYPE = TC, RANDOM , 20, Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	21465089,869	19	1129741,572	,949	,524
Within Groups	192931464,109	162	1190934,964		
Total	214396553,978	181			

a TYPE = TC, RANDOM , 20, Exp 3 , COLOR = 1

TYPE = TC, RANDOM , 20, Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

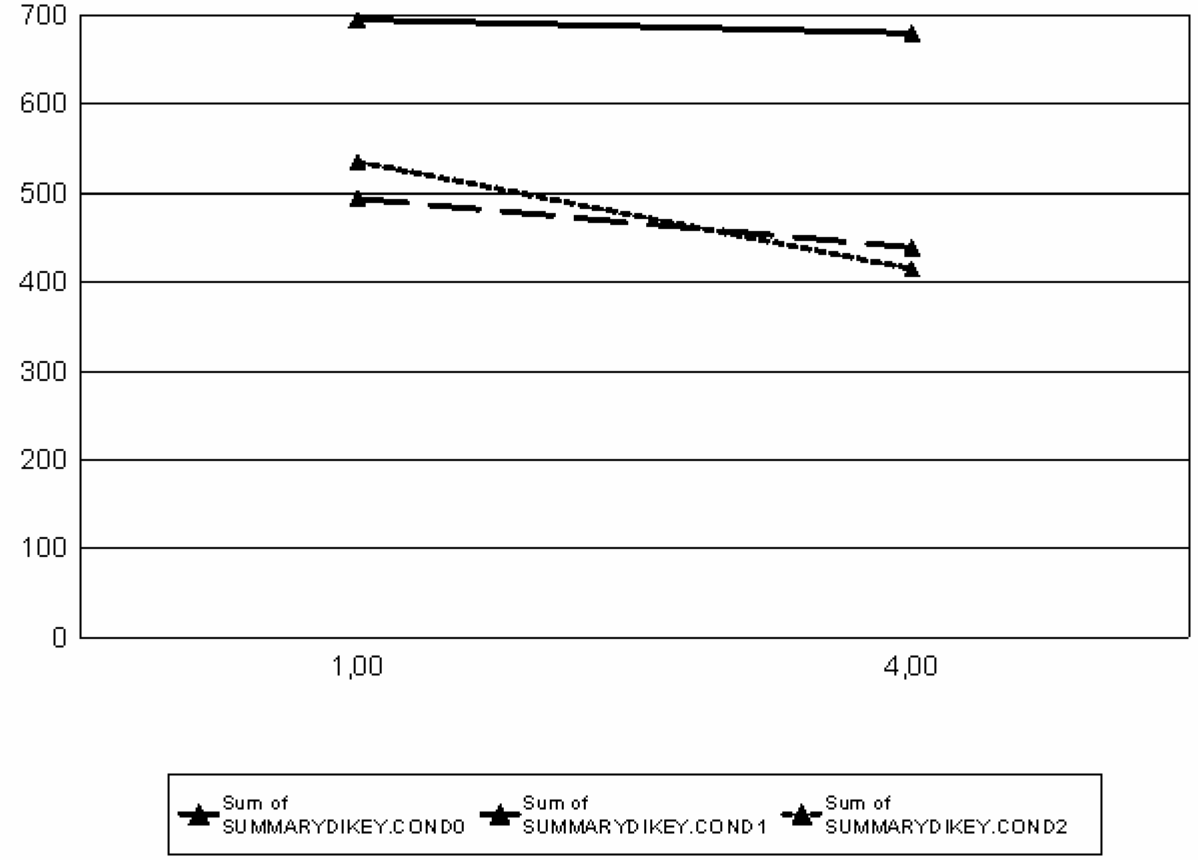
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	40954451,329	19	2155497,438	1,742	,035
Within Groups	185597250,577	150	1237315,004		
Total	226551701,906	169			

a TYPE = TC, RANDOM , 20, Exp 3 , COLOR = 2

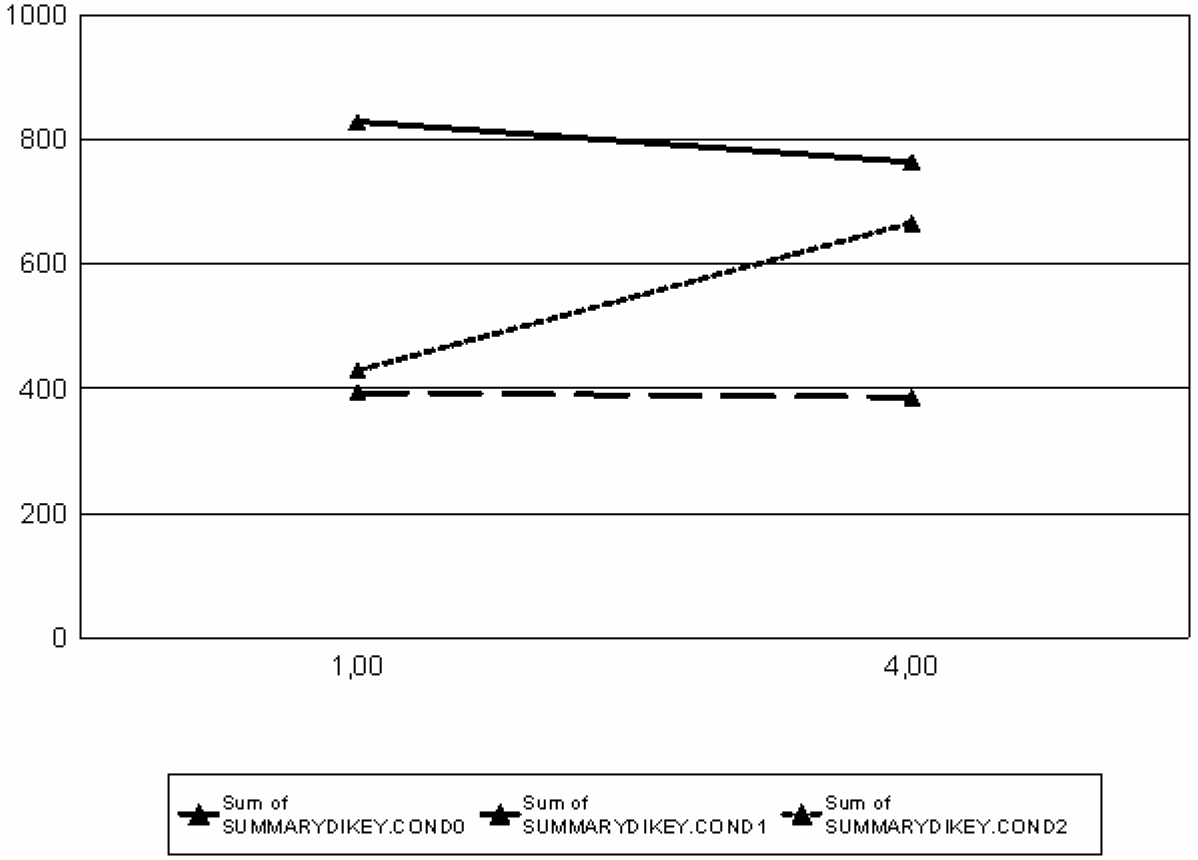
Appendix 3. Graphical representation of reaction types per experiment type

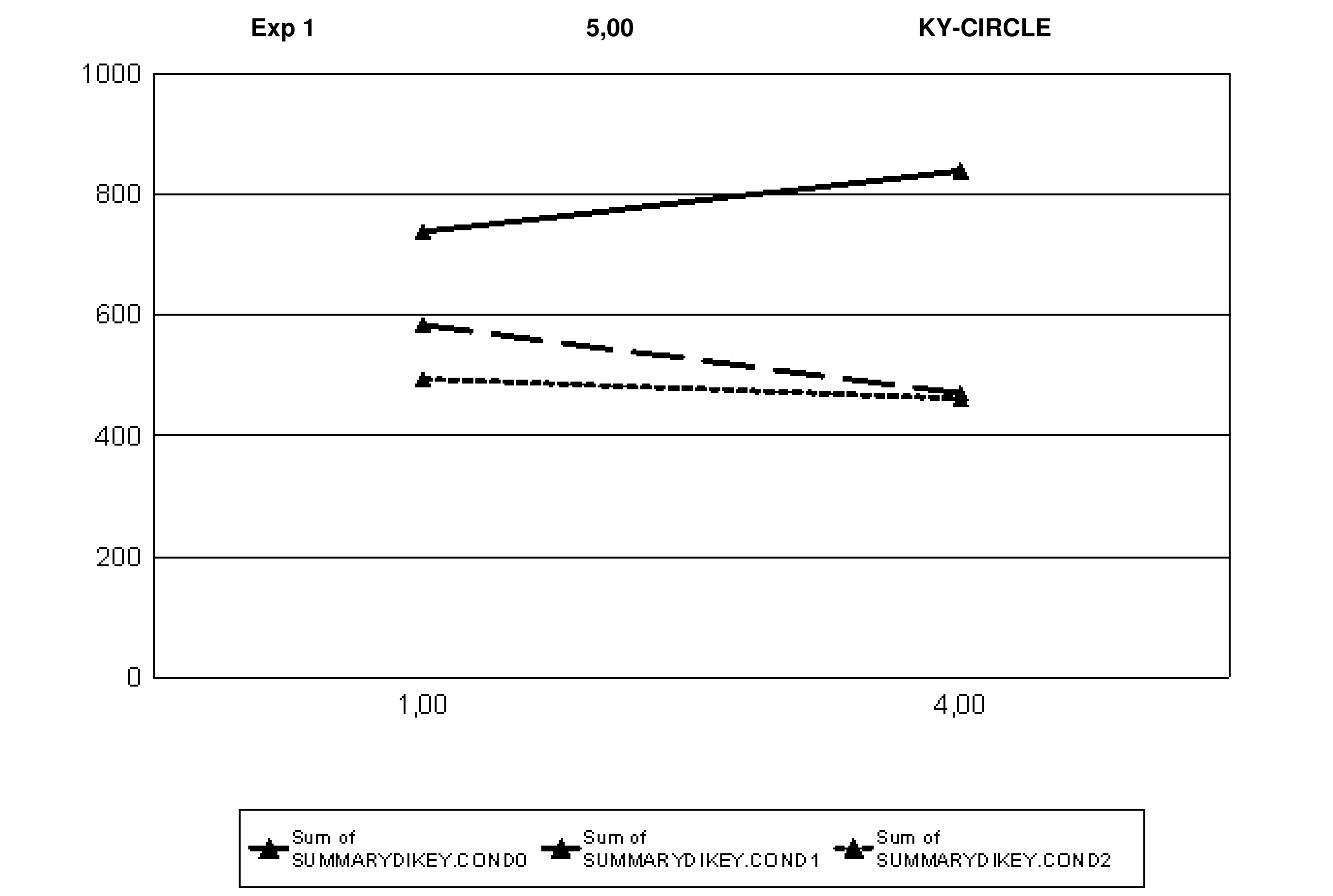
This report is the graphical representation of data presented in Appendix 2.

Exp 1 **5,00** **BR-CIRCLE**

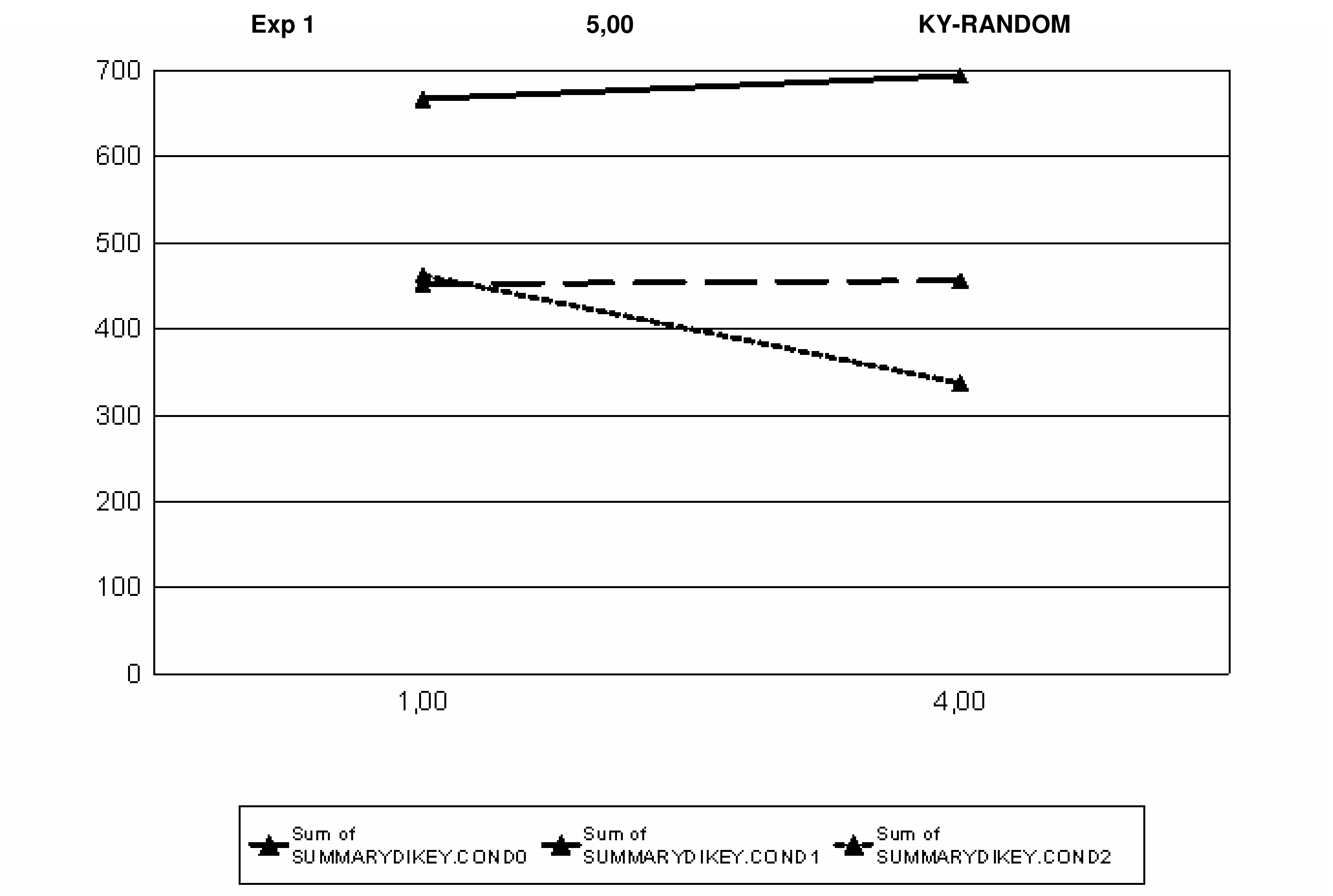


Exp 1 **5,00** **BR-RANDOM**

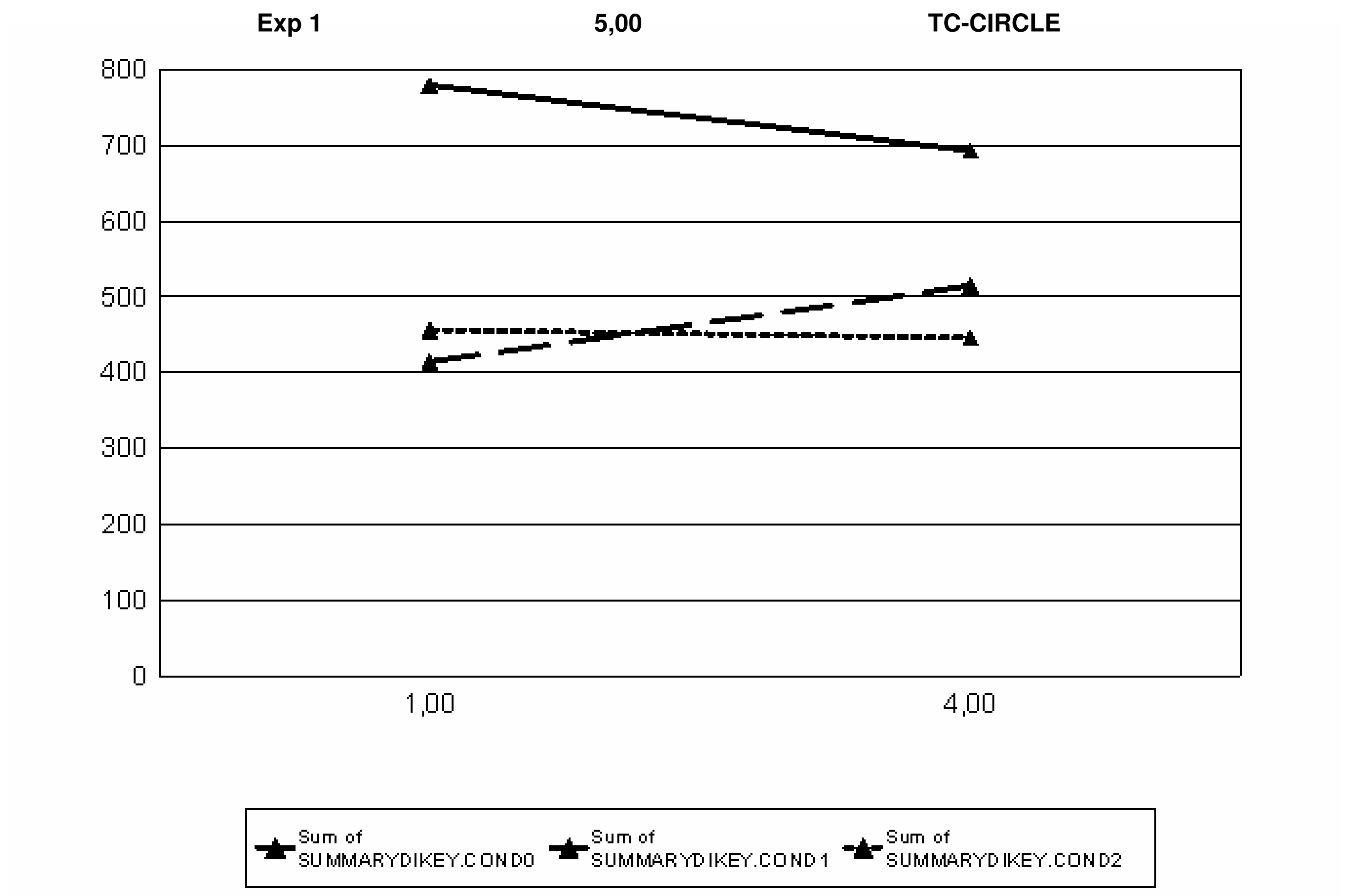


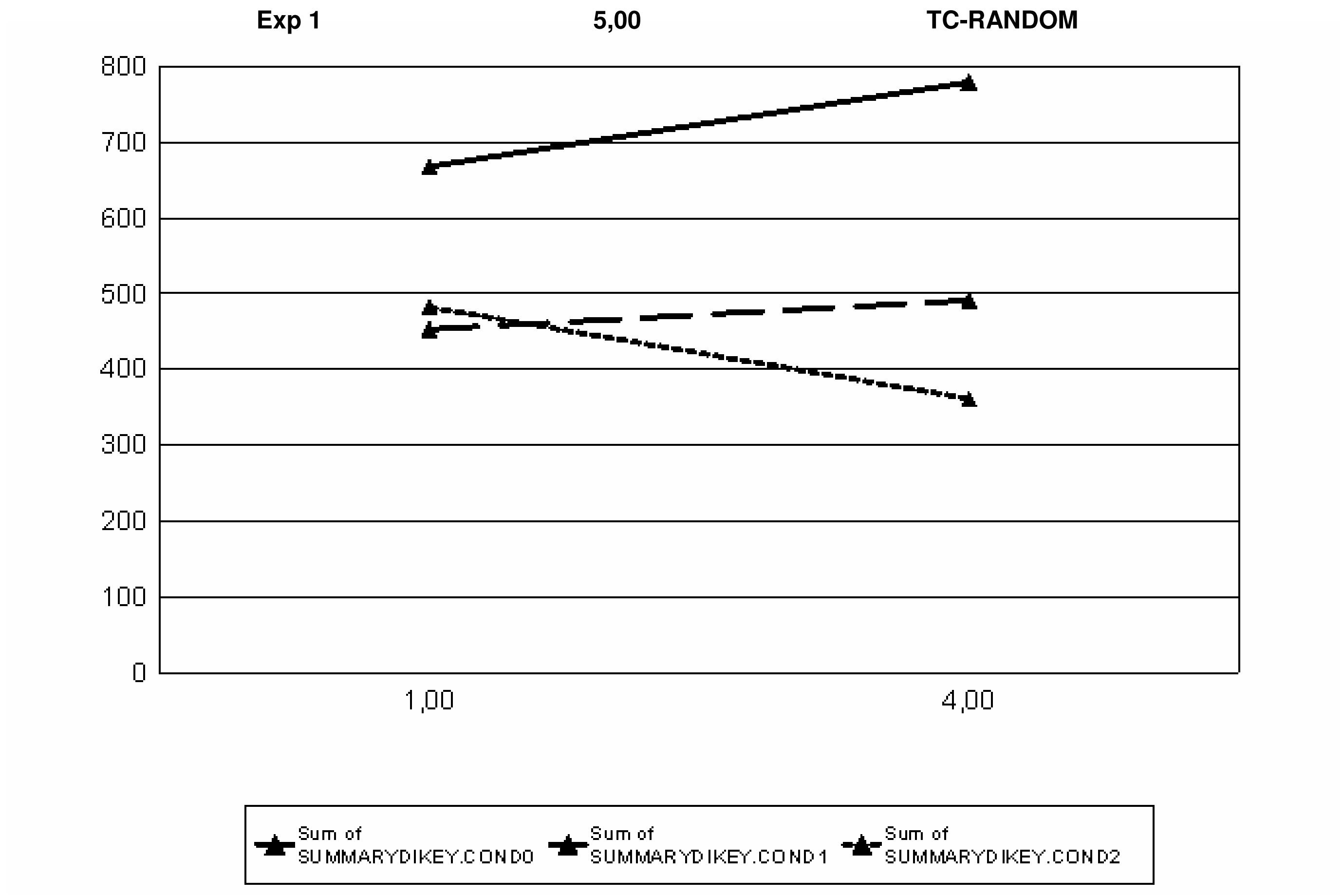


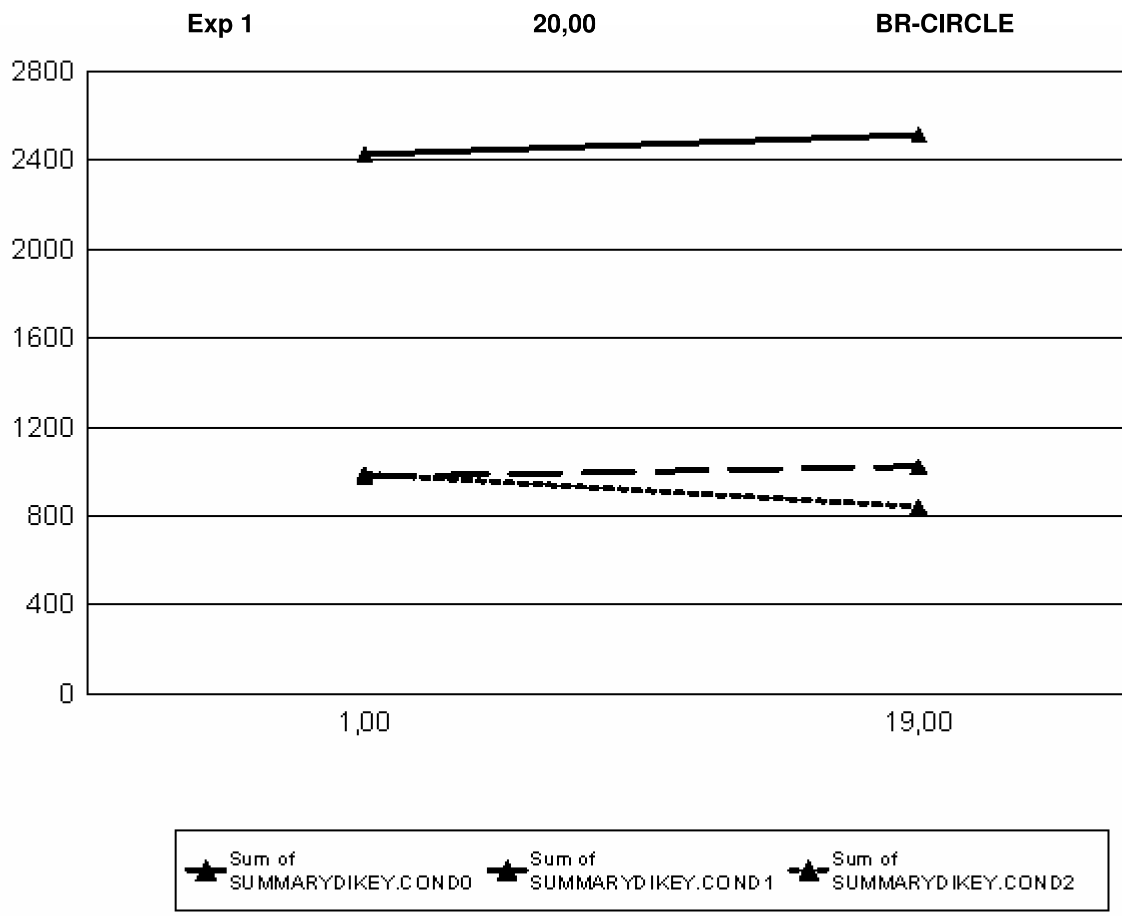
Exp 1 5,00 KY-CIRCLE

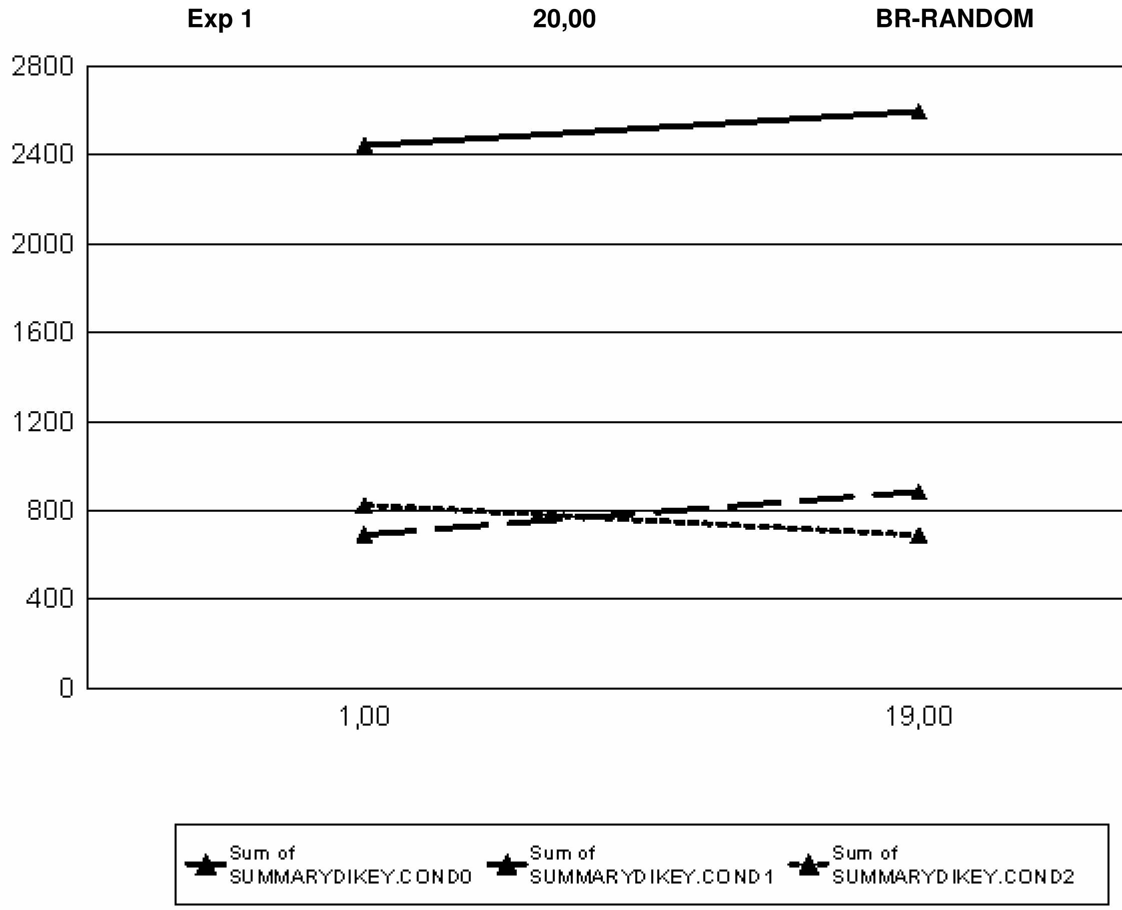


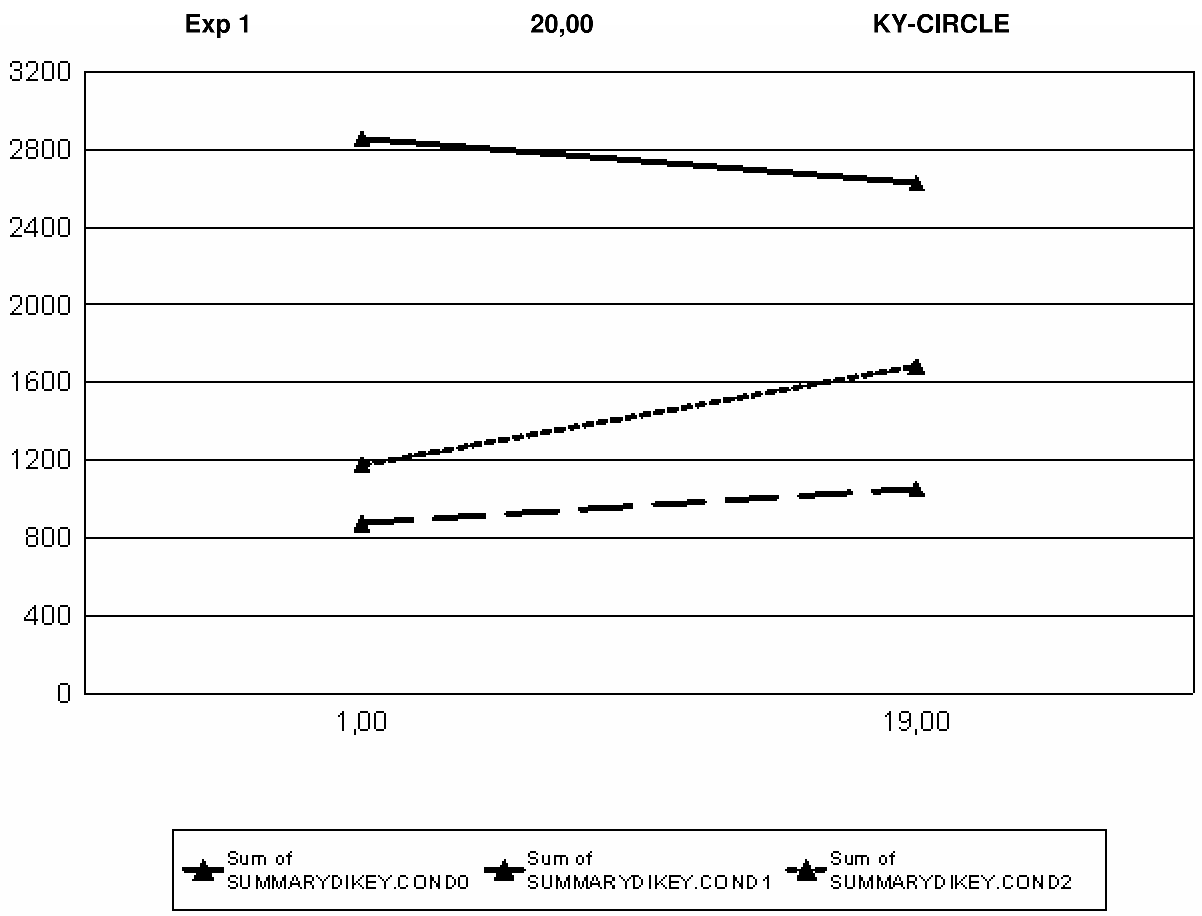
Exp 1 5,00 KY-RANDOM


Exp 1
5,00
TC-CIRCLE
800
700
600
500
400
300
200
100
0
1,00
4,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

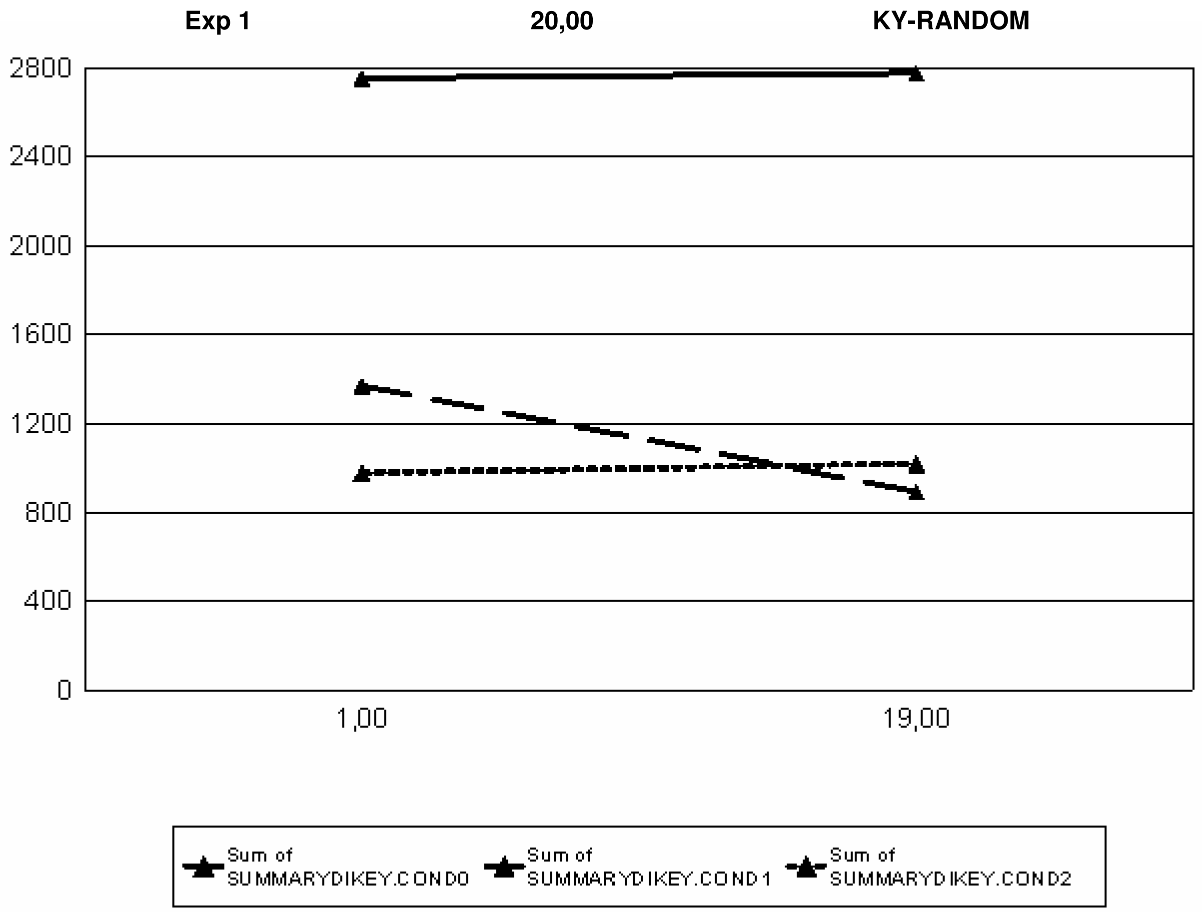

Exp 1
5,00
TC-RANDOM
800
700
600
500
400
300
200
100
0
1,00
4,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 1
20,00
BR-CIRCLE
2800
2400
2000
1600
1200
800
400
0
1,00
19,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

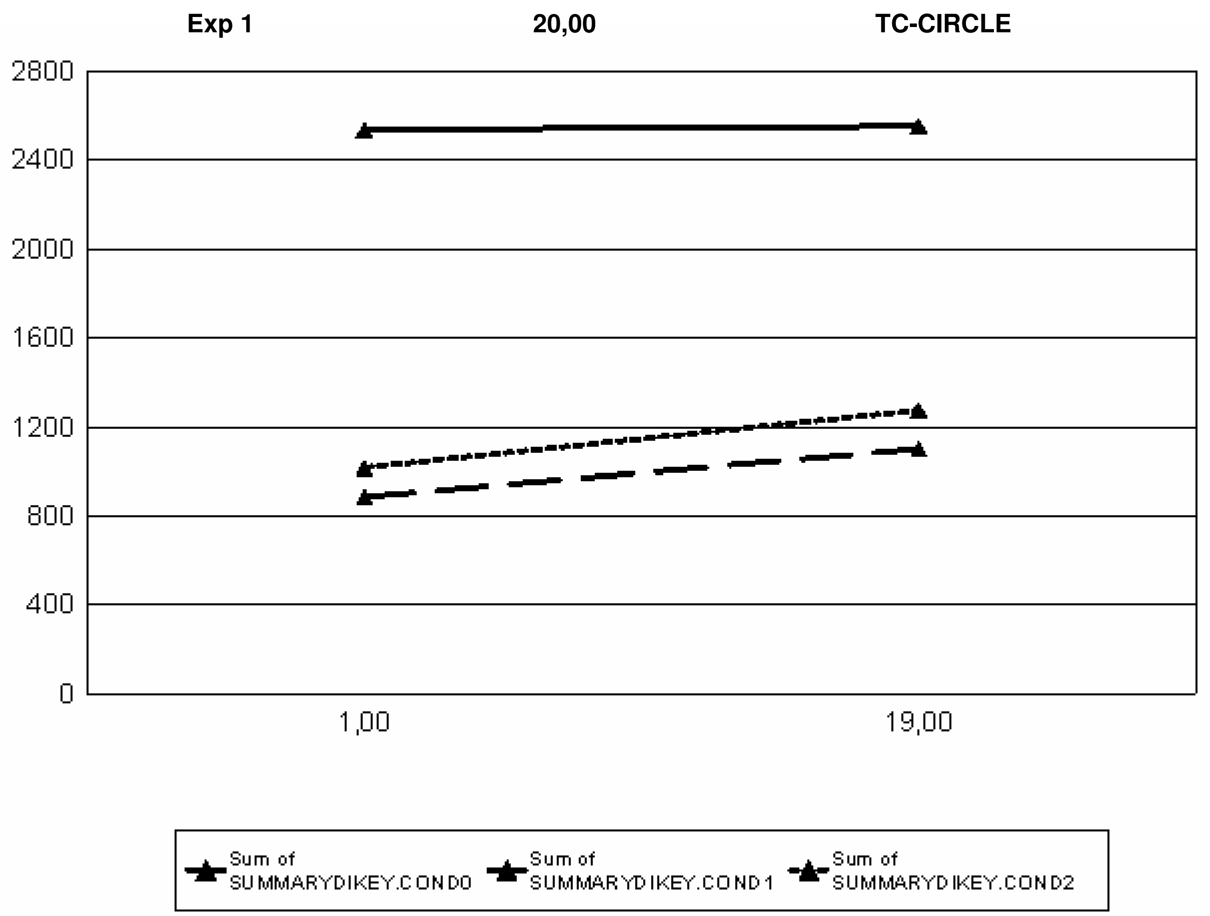

Exp 1
20,00
BR-RANDOM
2800
2400
2000
1600
1200
800
400
0
1,00
19,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

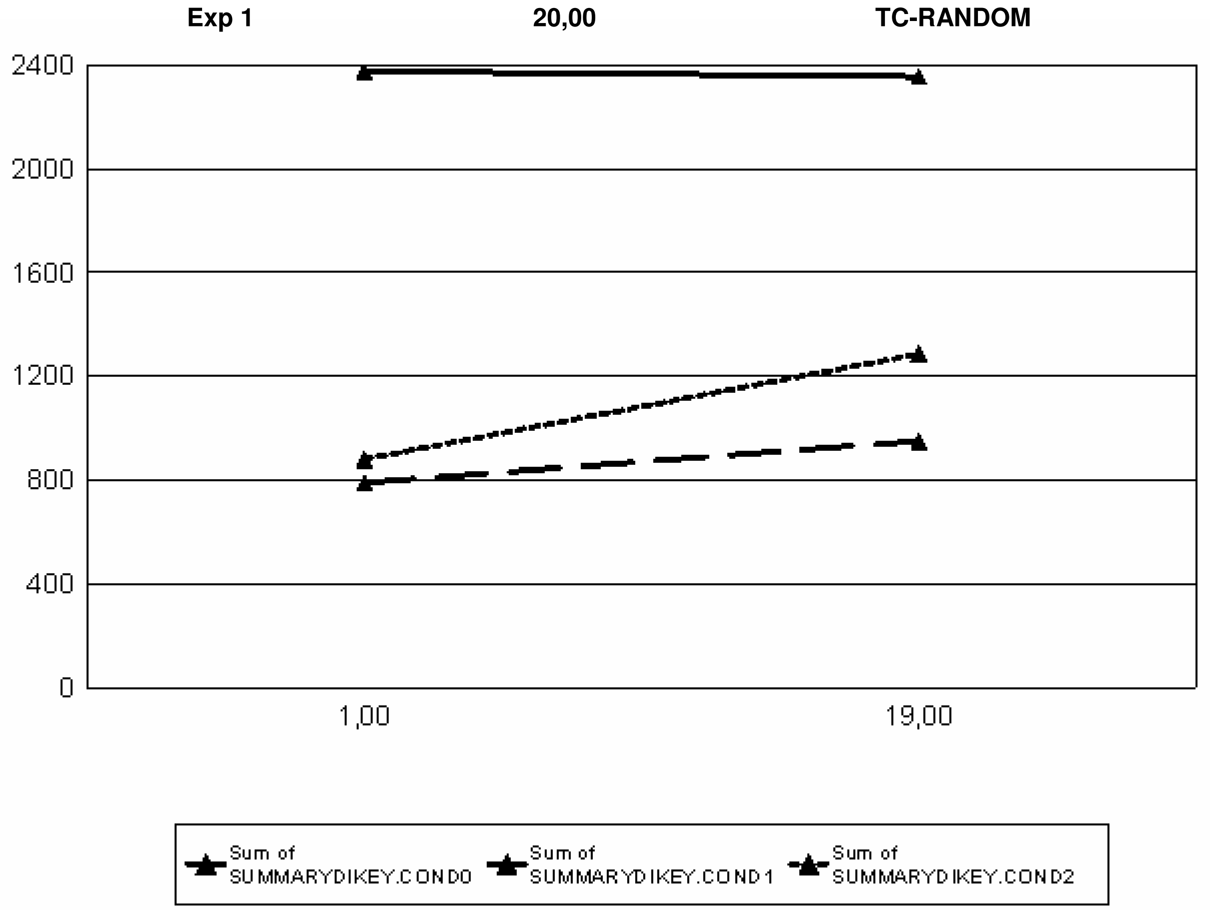


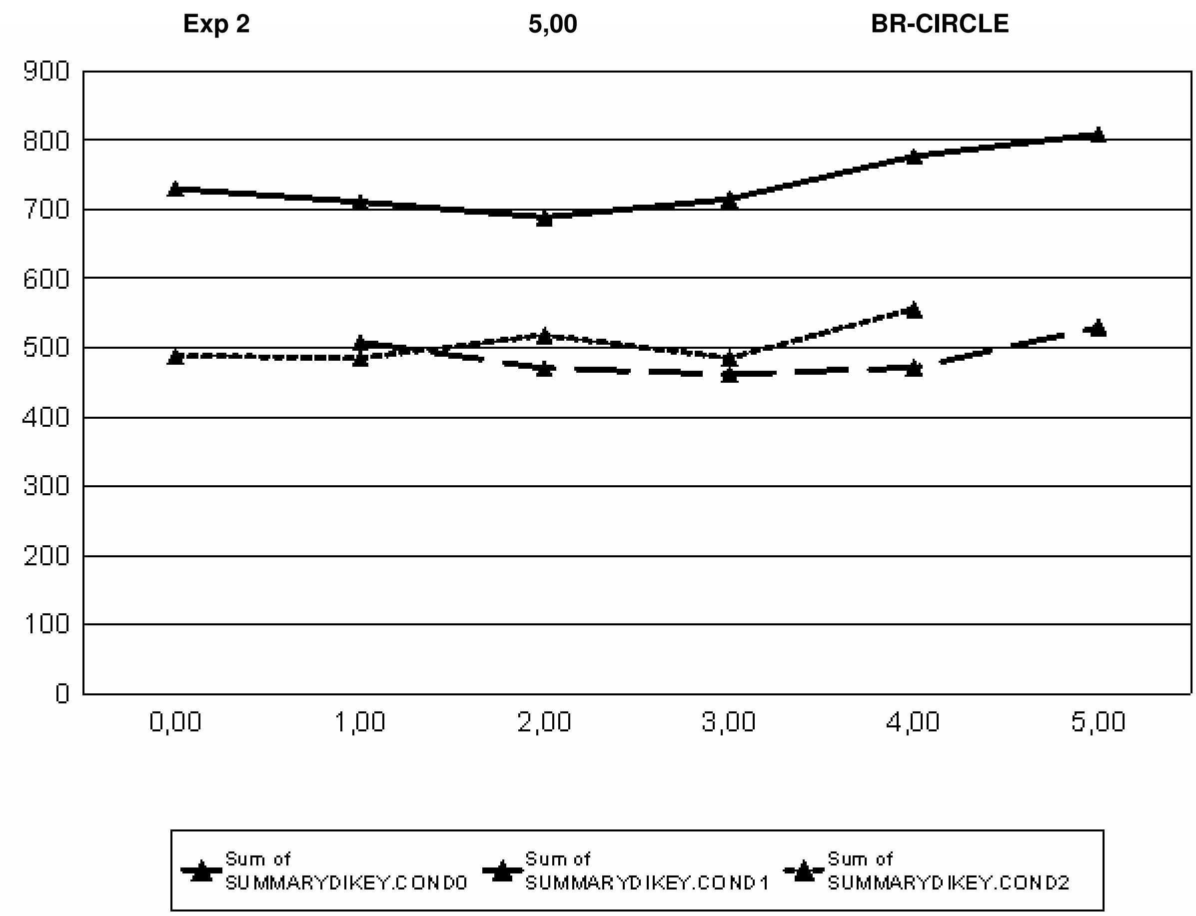
Exp 1 20,00 KY-CIRCLE

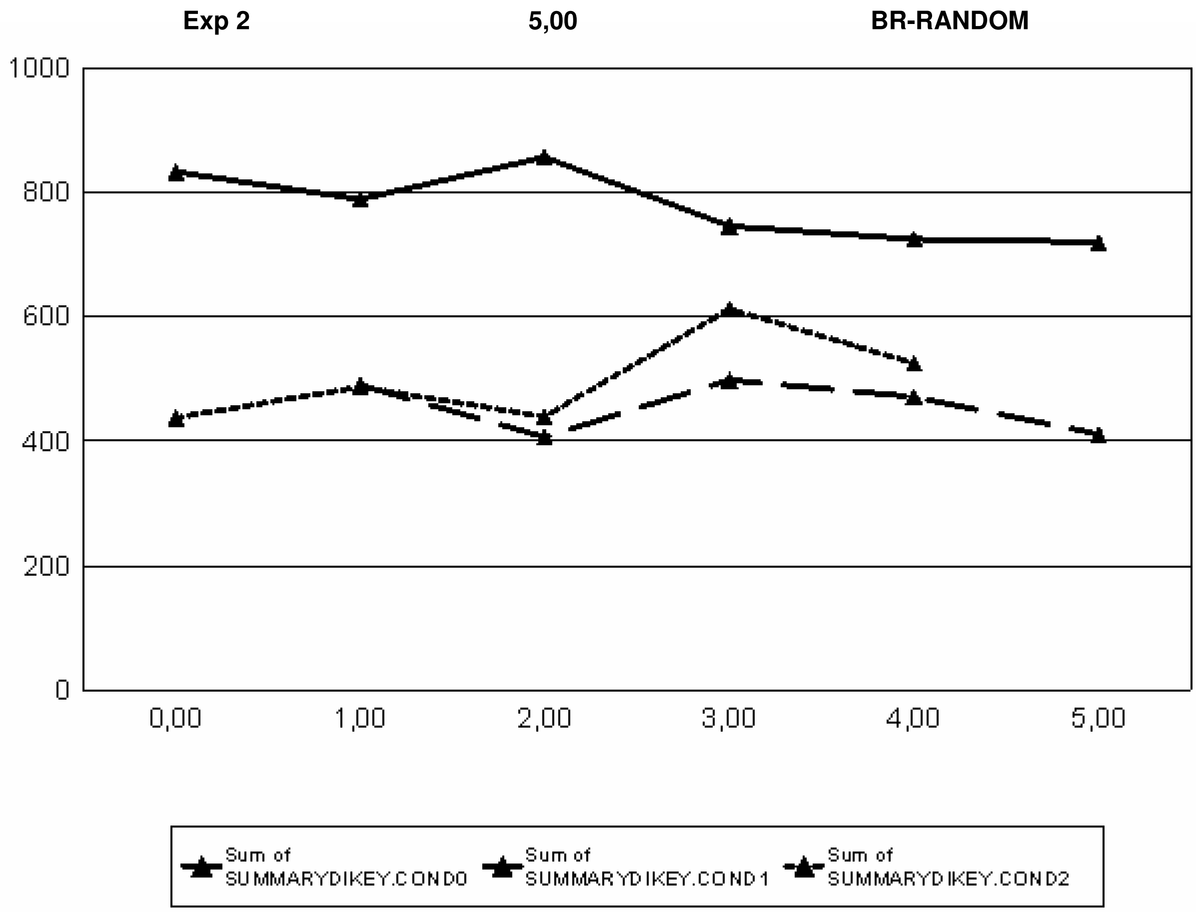


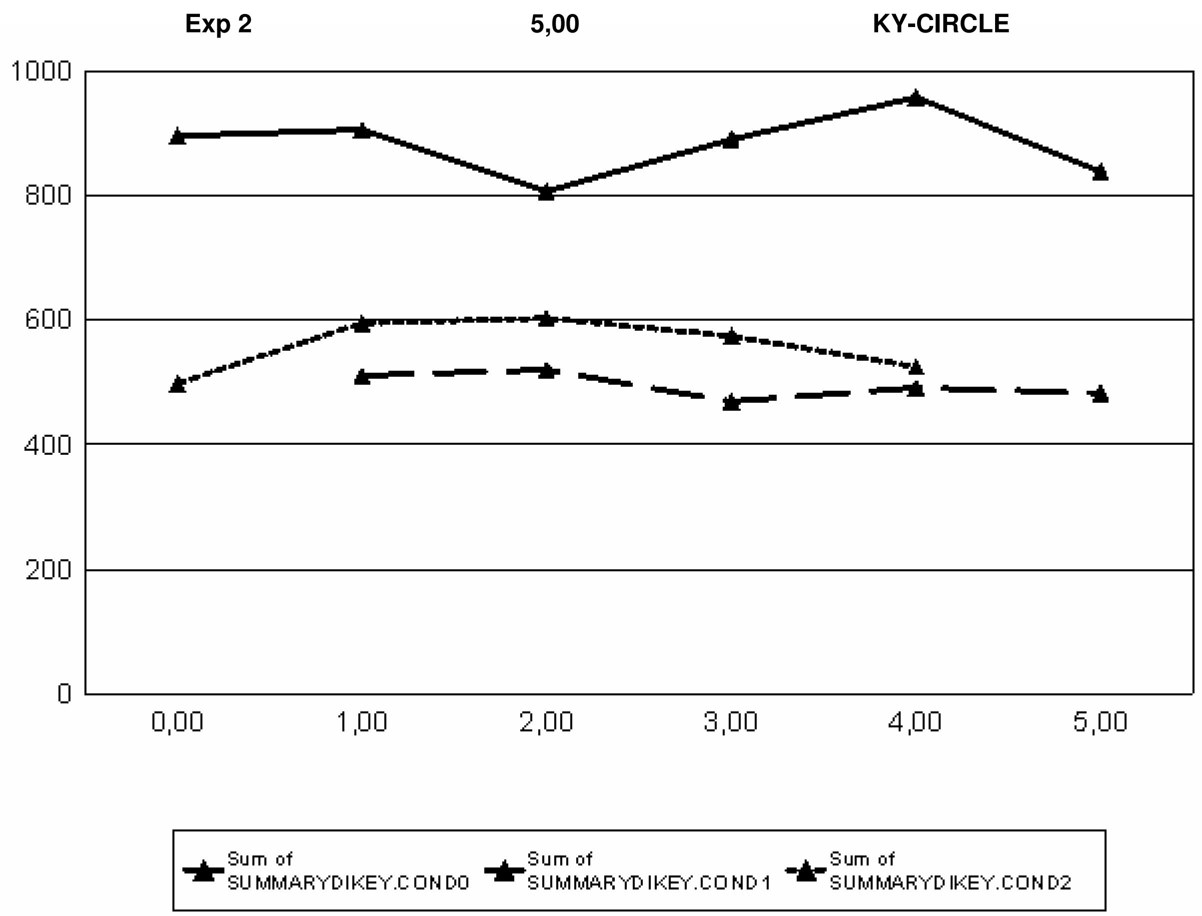
Exp 1 20,00 KY-RANDOM

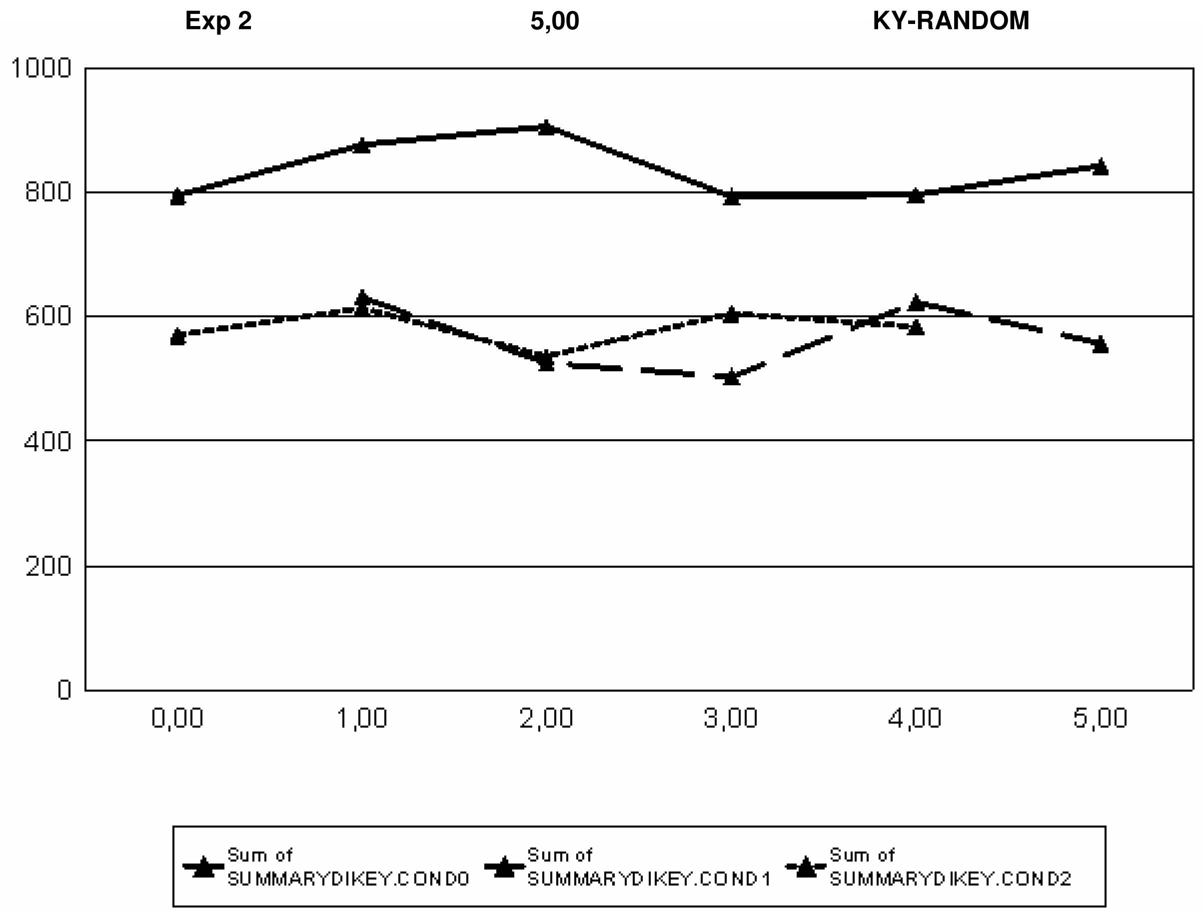

Exp 1
20,00
TC-CIRCLE
2800
2400
2000
1600
1200
800
400
0
1,00
19,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

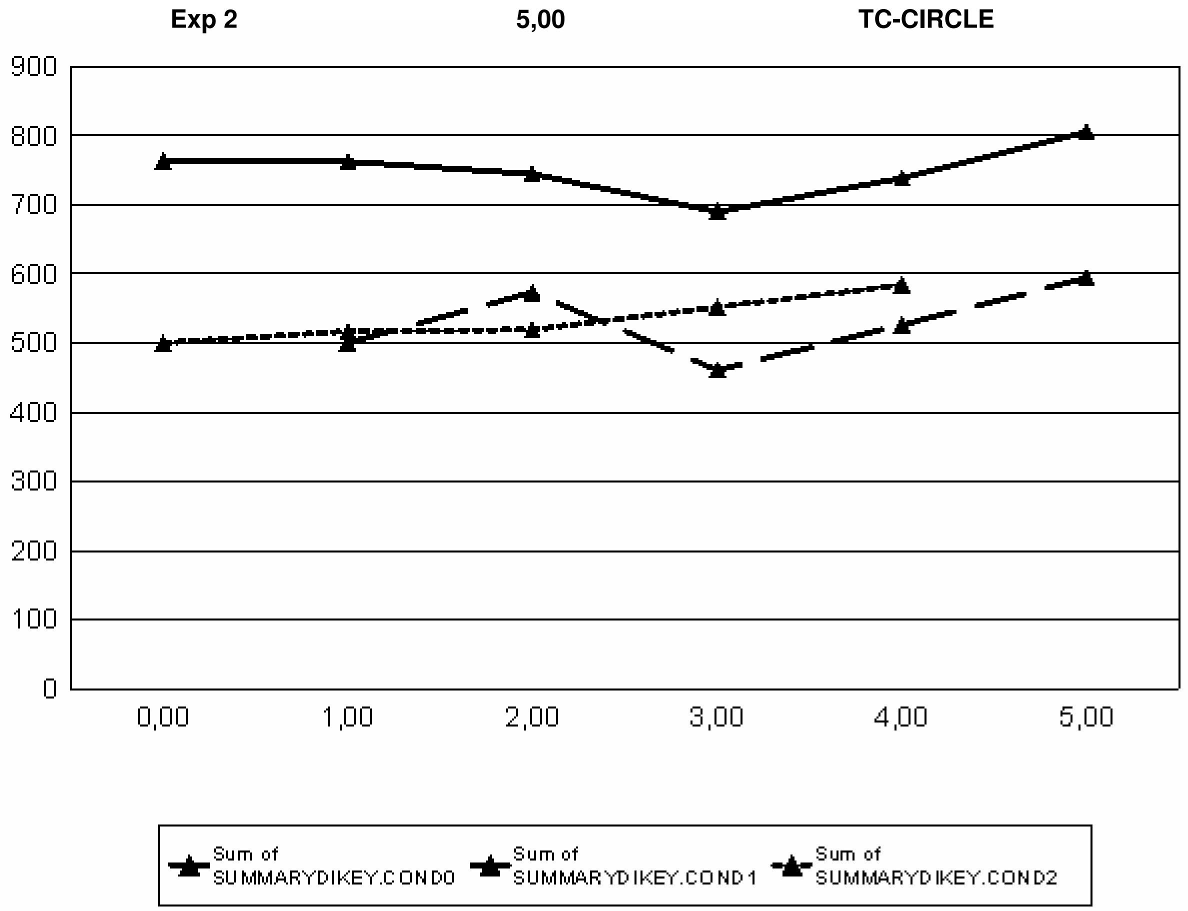

Exp 1
20,00
TC-RANDOM
2400
2000
1600
1200
800
400
0
1,00
19,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

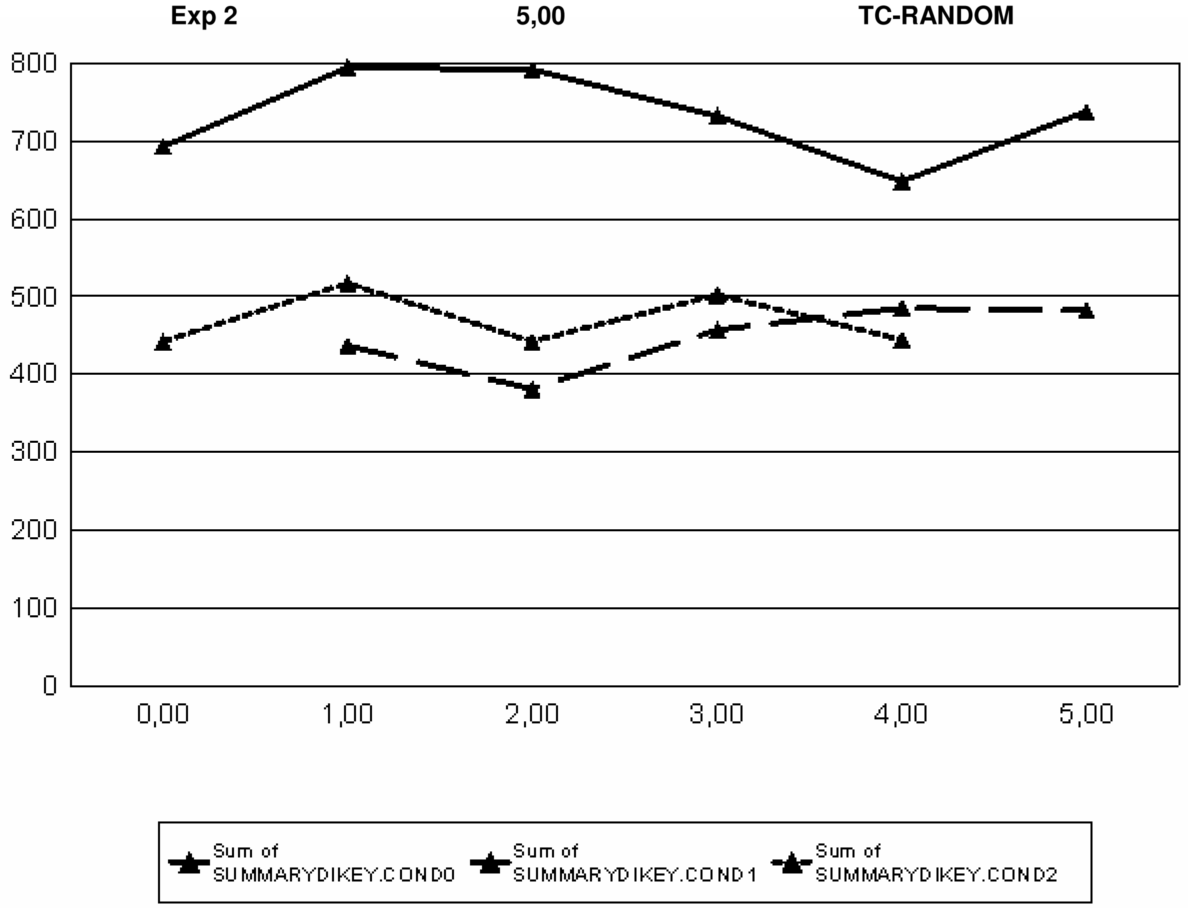

Exp 2
5,00
BR-CIRCLE
900
800
700
600
500
400
300
200
100
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

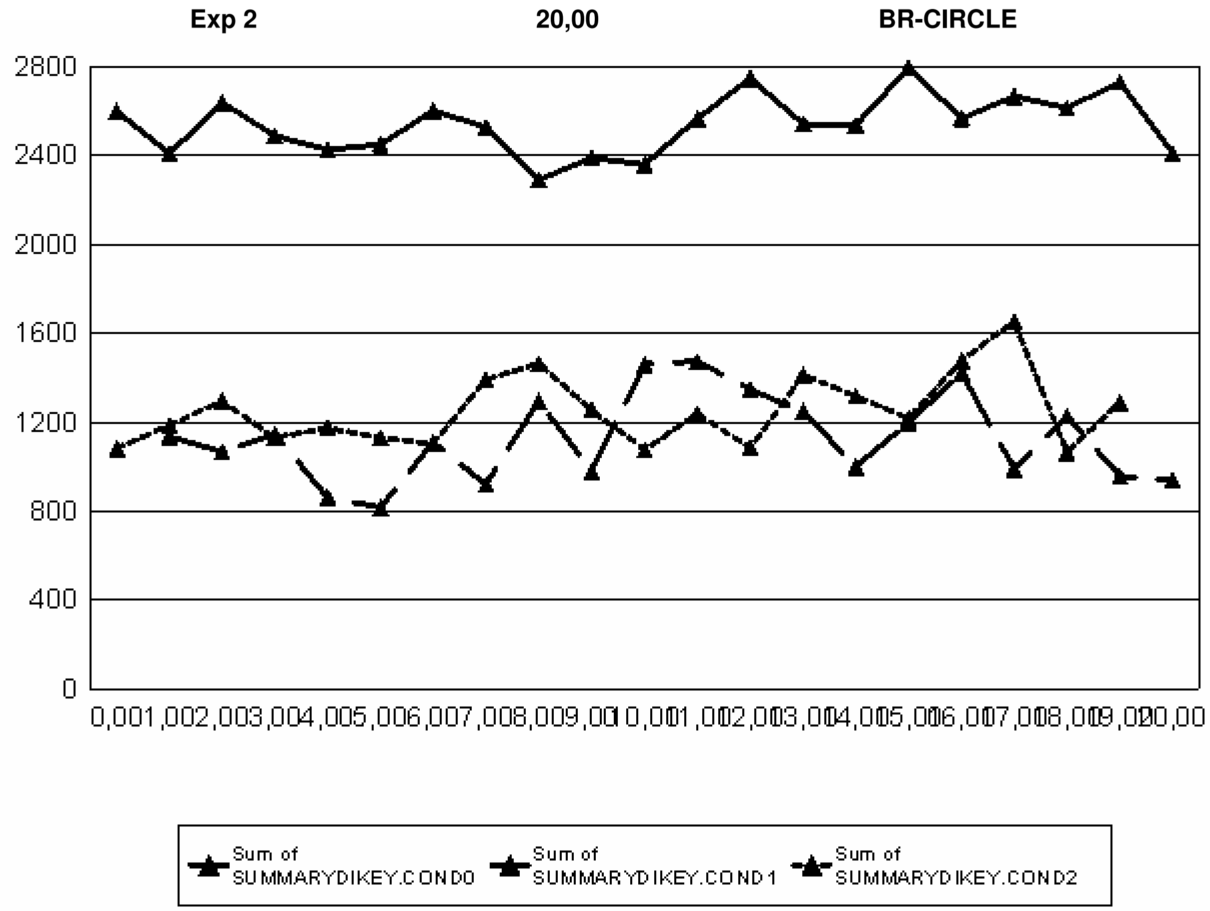

Exp 2
5,00
BR-RANDOM
1000
800
600
400
200
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

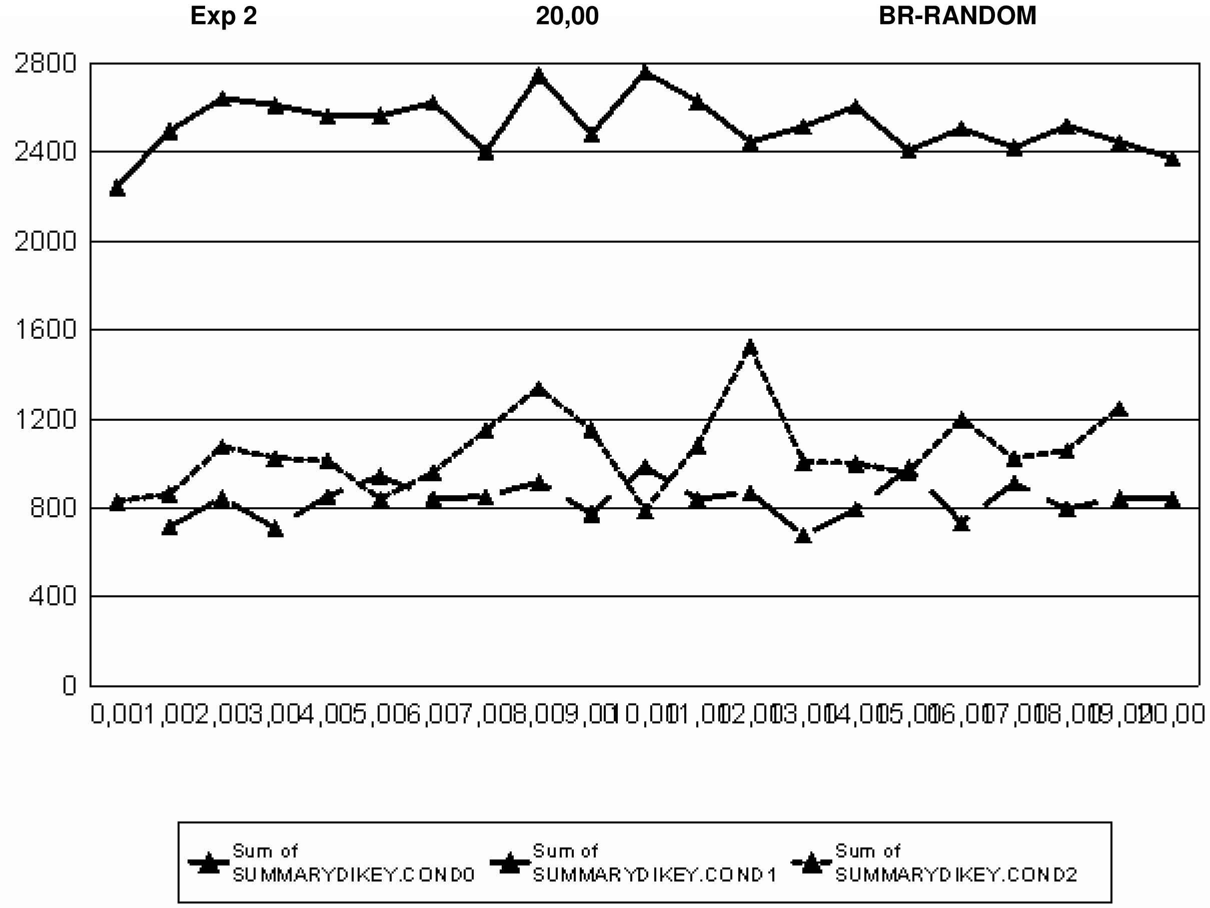

Exp 2
5,00
KY-CIRCLE
1000
800
600
400
200
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 2
5,00
KY-RANDOM
1000
800
600
400
200
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

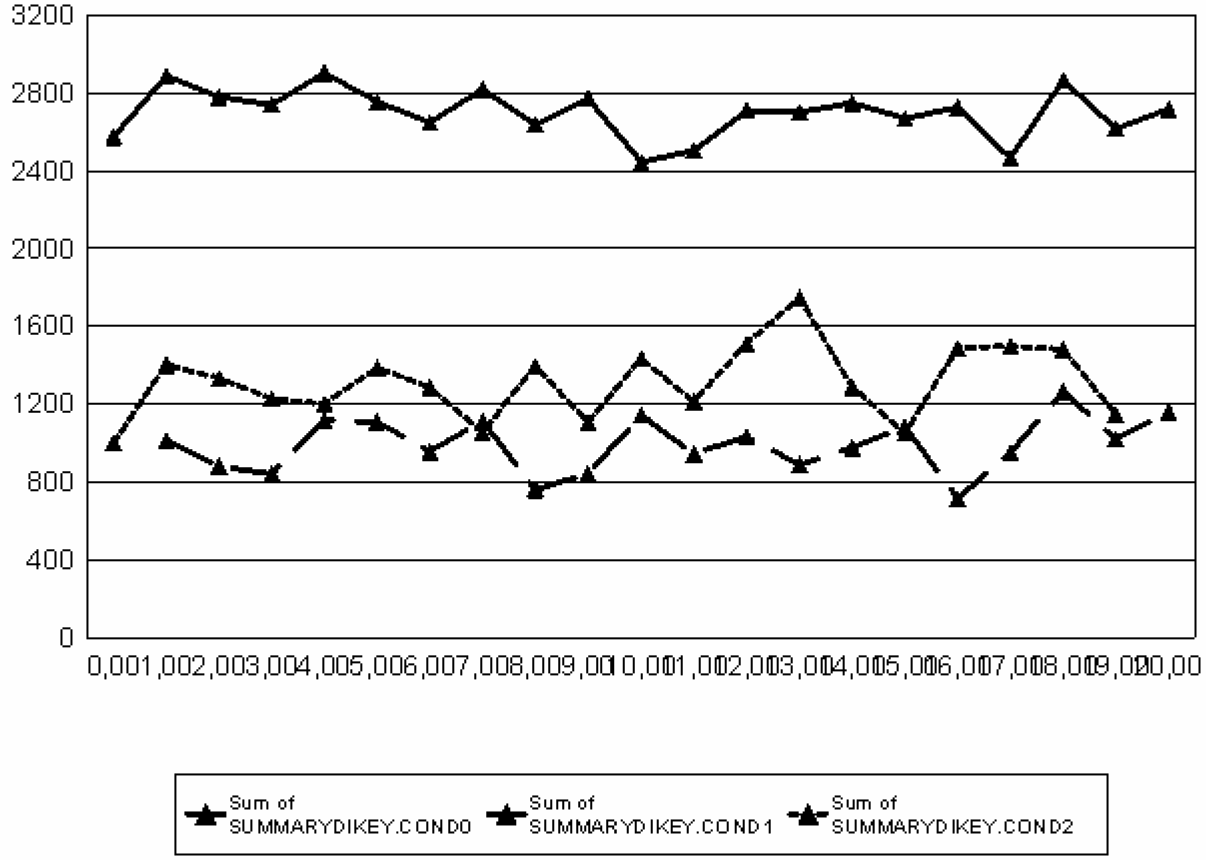

Exp 2
5,00
TC-CIRCLE
900
800
700
600
500
400
300
200
100
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 2
5,00
TC-RANDOM
800
700
600
500
400
300
200
100
0
0,00
1,00
2,00
3,00
4,00
5,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

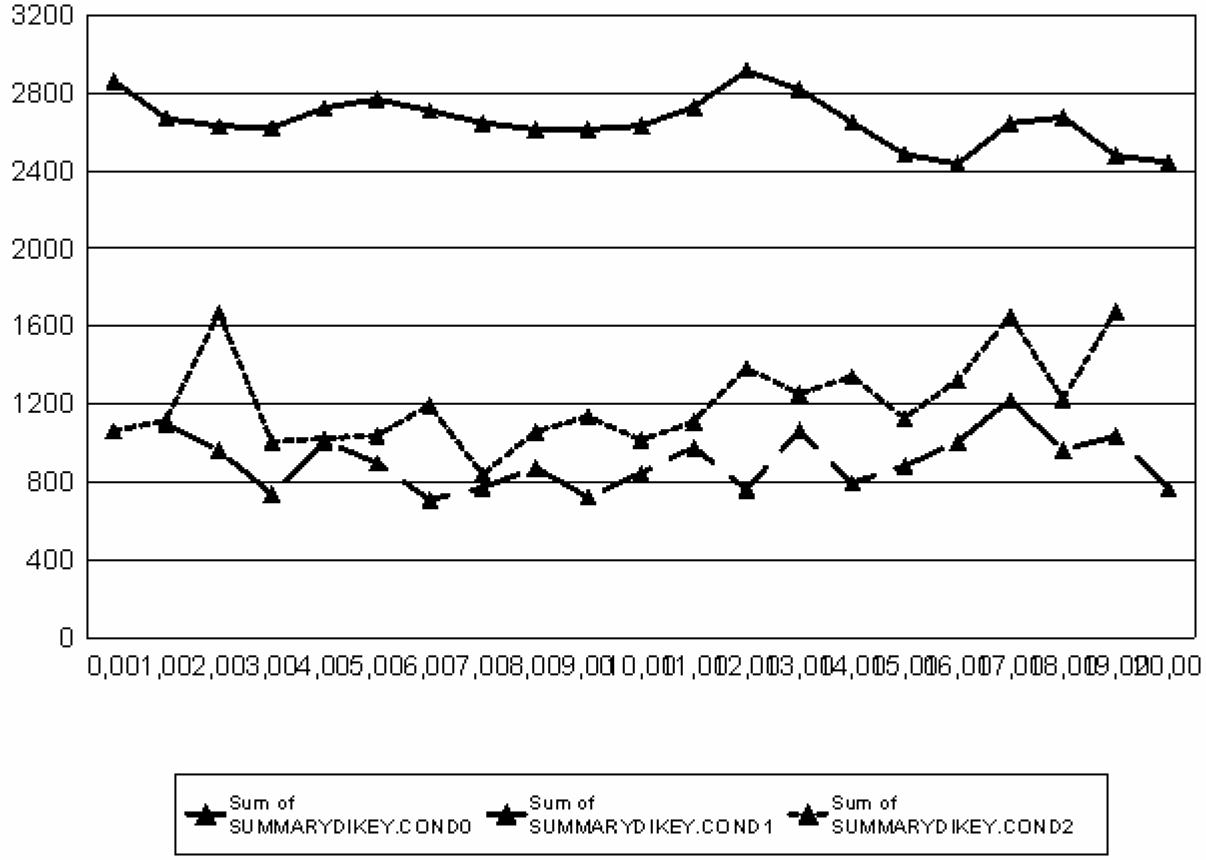

Exp 2
20,00
BR-CIRCLE
2800
2400
2000
1600
1200
800
400
0
0,00 1,00 2,00 3,00 4,00 5,00 6,00 7,00 8,00 9,00 10,00 11,00 12,00 13,00 14,00 15,00 16,00 17,00 18,00 19,00 20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

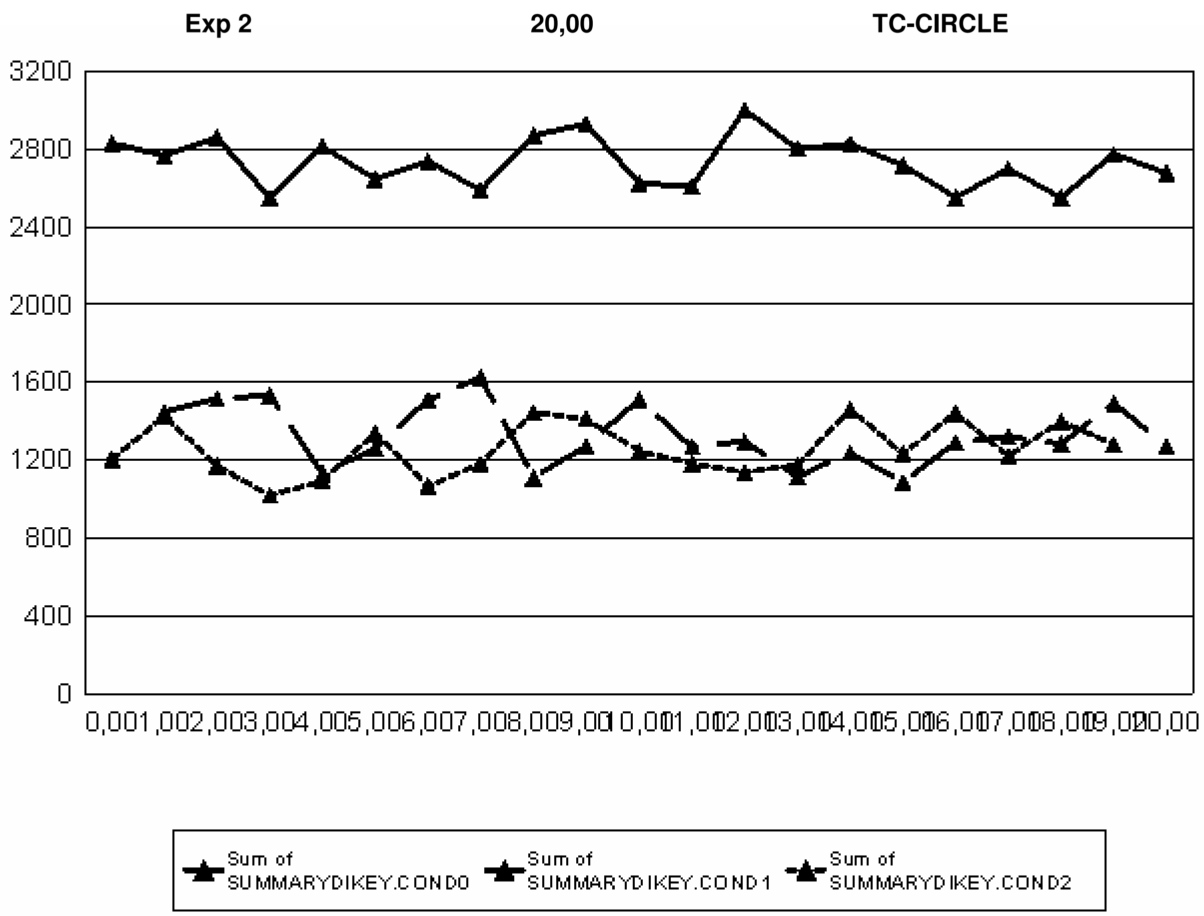

Exp 2
20,00
BR-RANDOM
2800
2400
2000
1600
1200
800
400
0
0,00 1,00 2,00 3,00 4,00 5,00 6,00 7,00 8,00 9,00 10,00 11,00 12,00 13,00 14,00 15,00 16,00 17,00 18,00 19,00 20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

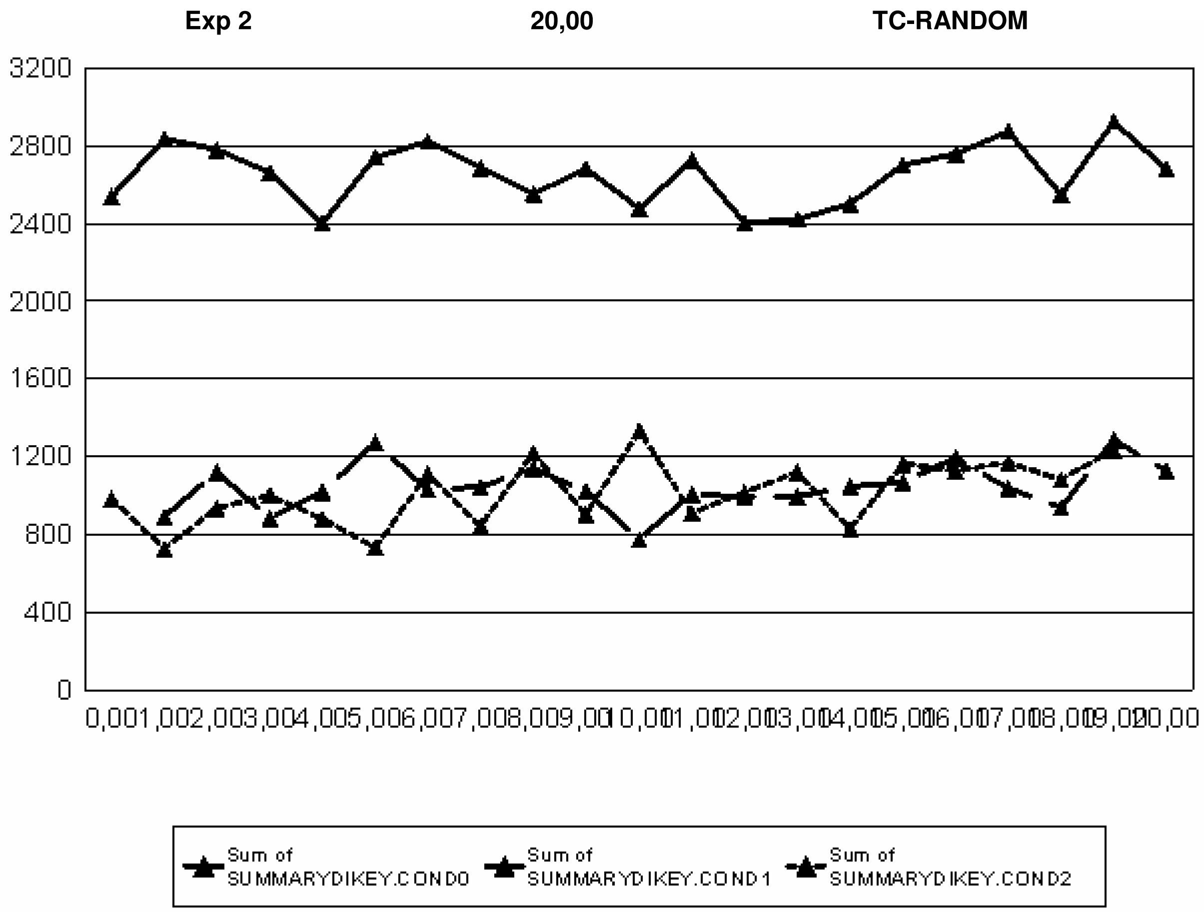
Exp 2 **20,00** **KY-CIRCLE**



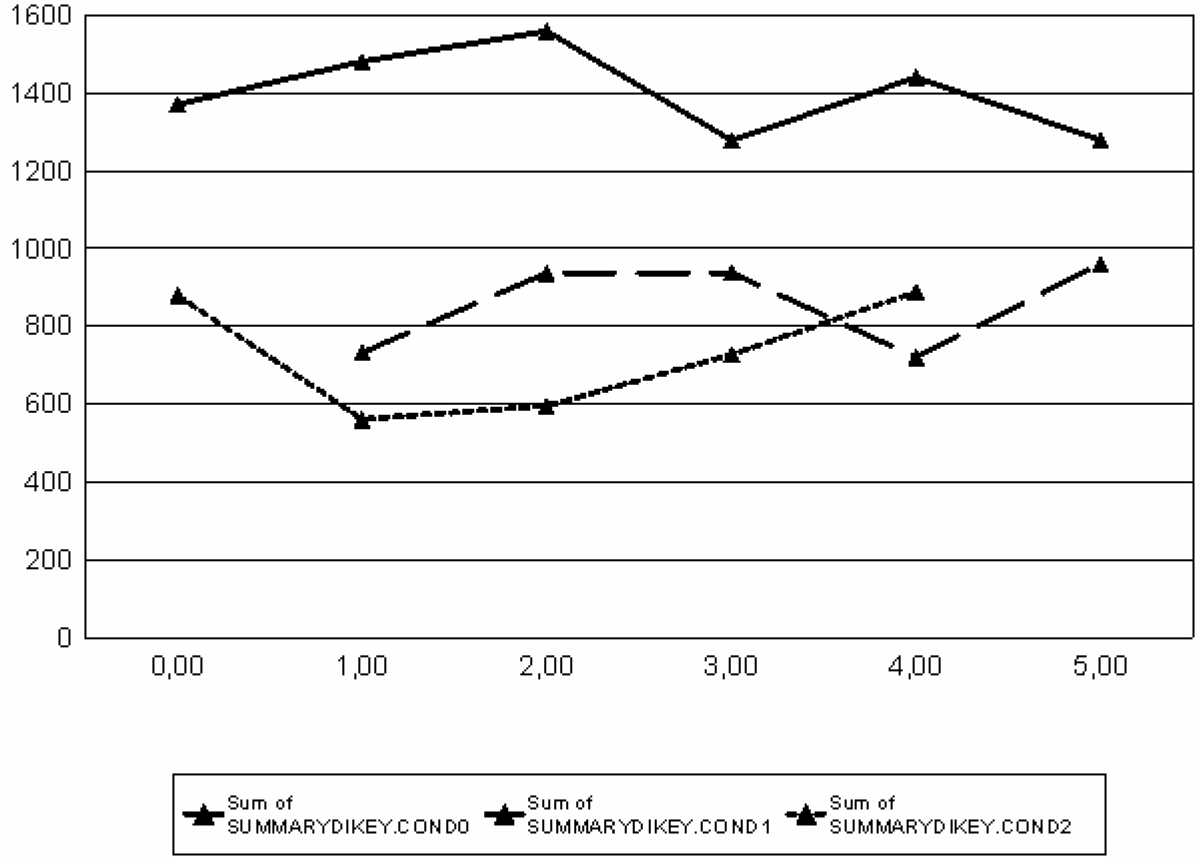
Exp 2 **20,00** **KY-RANDOM**



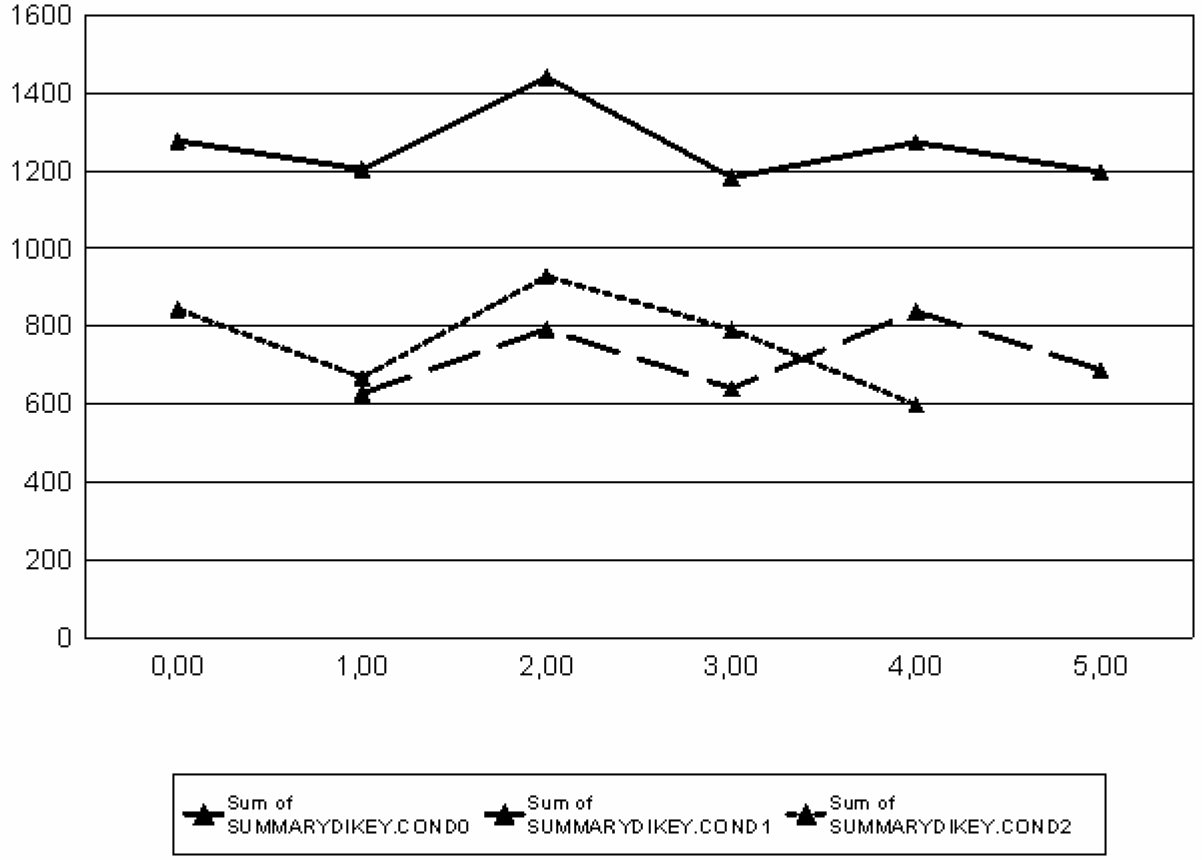

Exp 2
20,00
TC-CIRCLE
3200
2800
2400
2000
1600
1200
800
400
0
0,001,002,003,004,005,006,007,008,009,0010,0011,0012,0013,0014,0015,0016,0017,0018,0019,0020,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 2
20,00
TC-RANDOM
3200
2800
2400
2000
1600
1200
800
400
0
0,001,002,003,004,005,006,007,008,009,0010,0011,0012,0013,0014,0015,0016,0017,0018,0019,0020,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

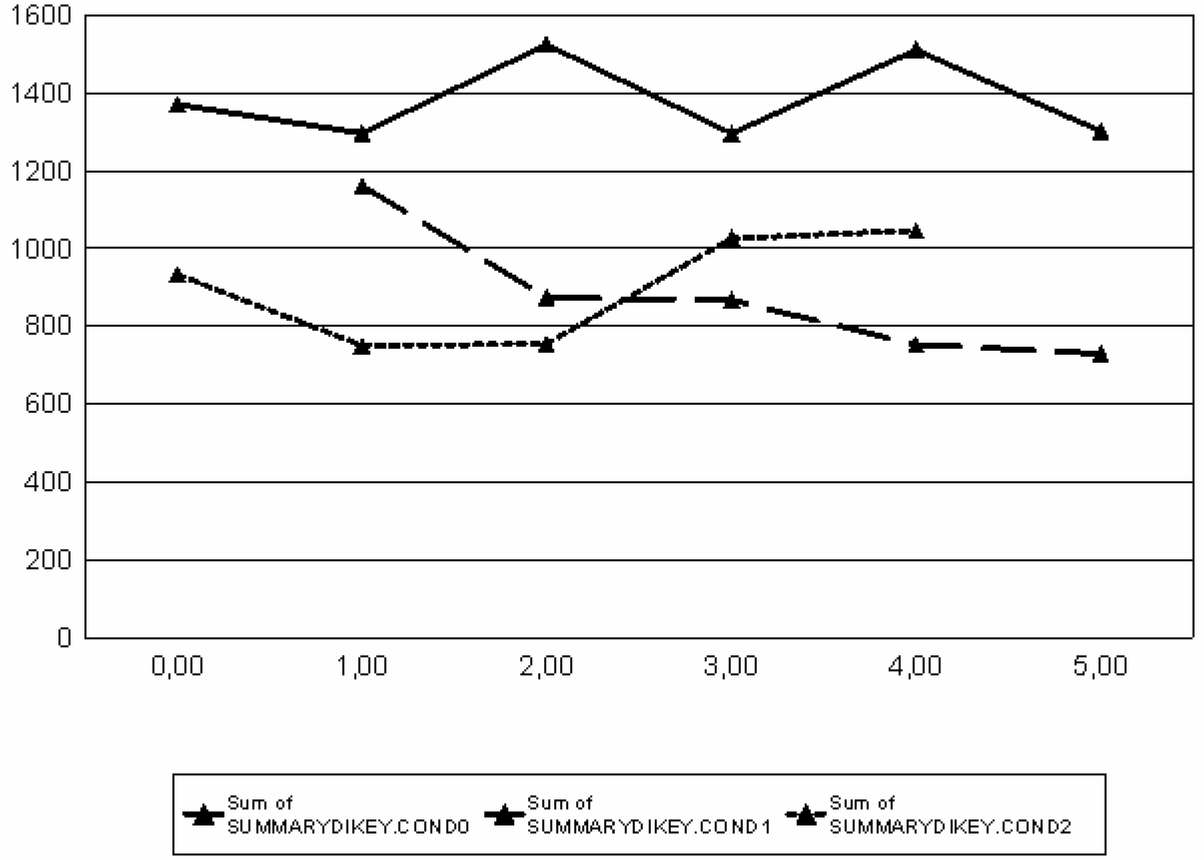
Exp 3 5,00 BR-CIRCLE



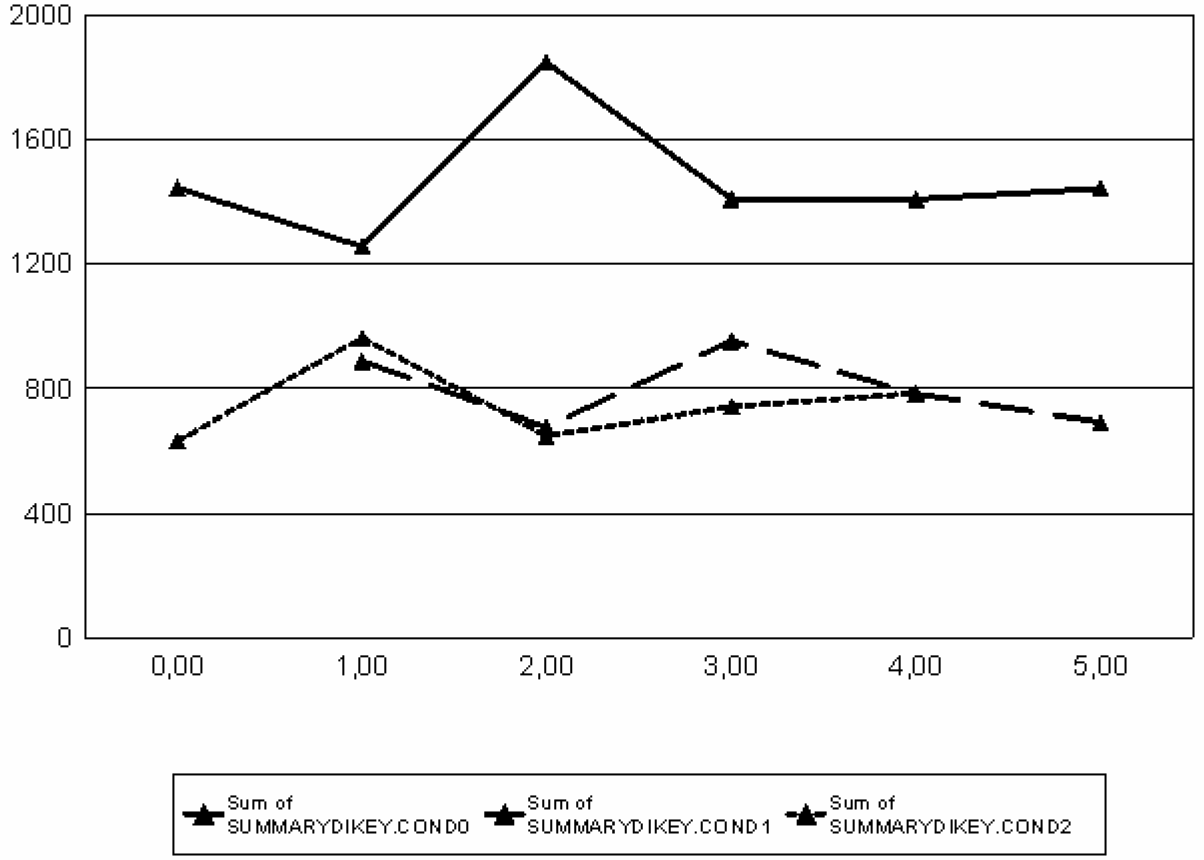
Exp 3 5,00 BR-RANDOM

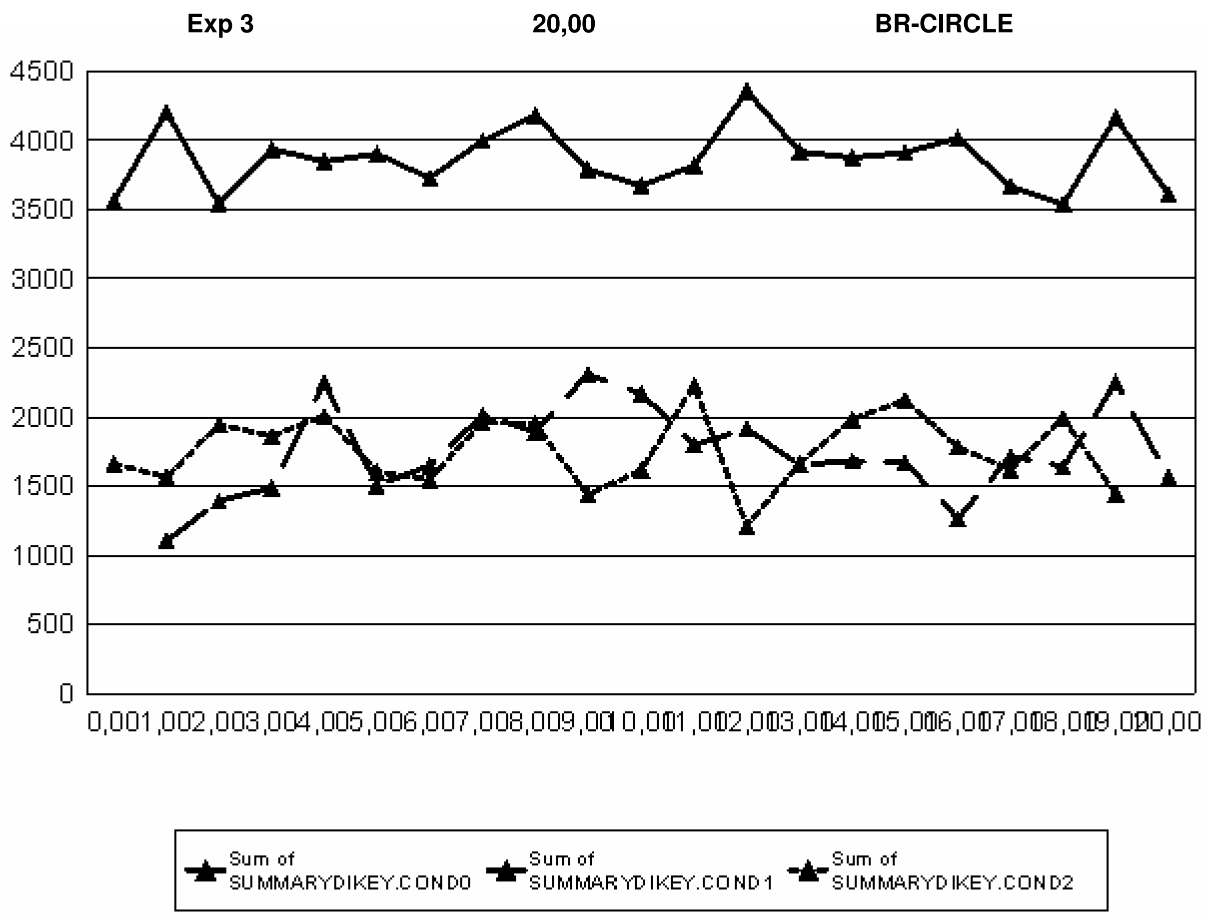


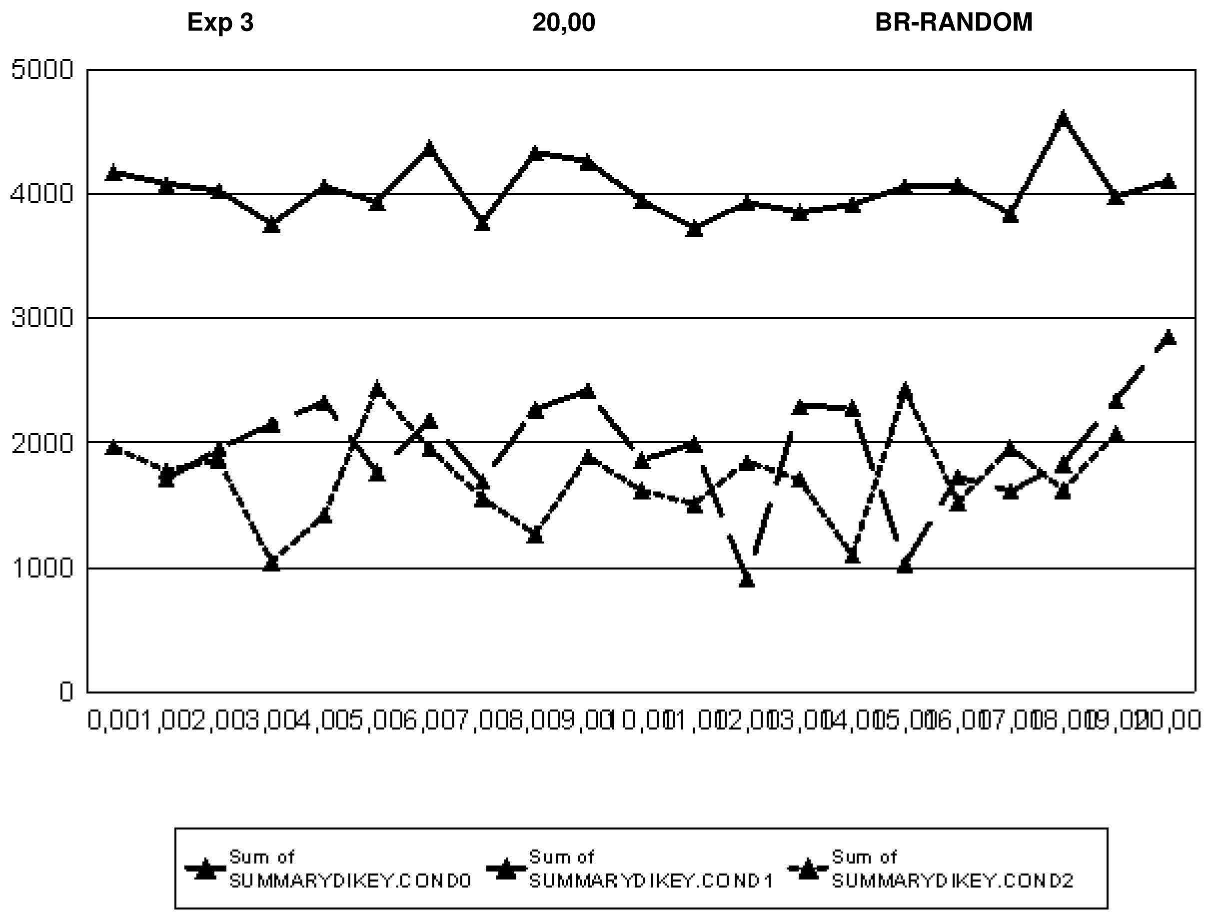
Exp 3 **5,00** **TC-CIRCLE**

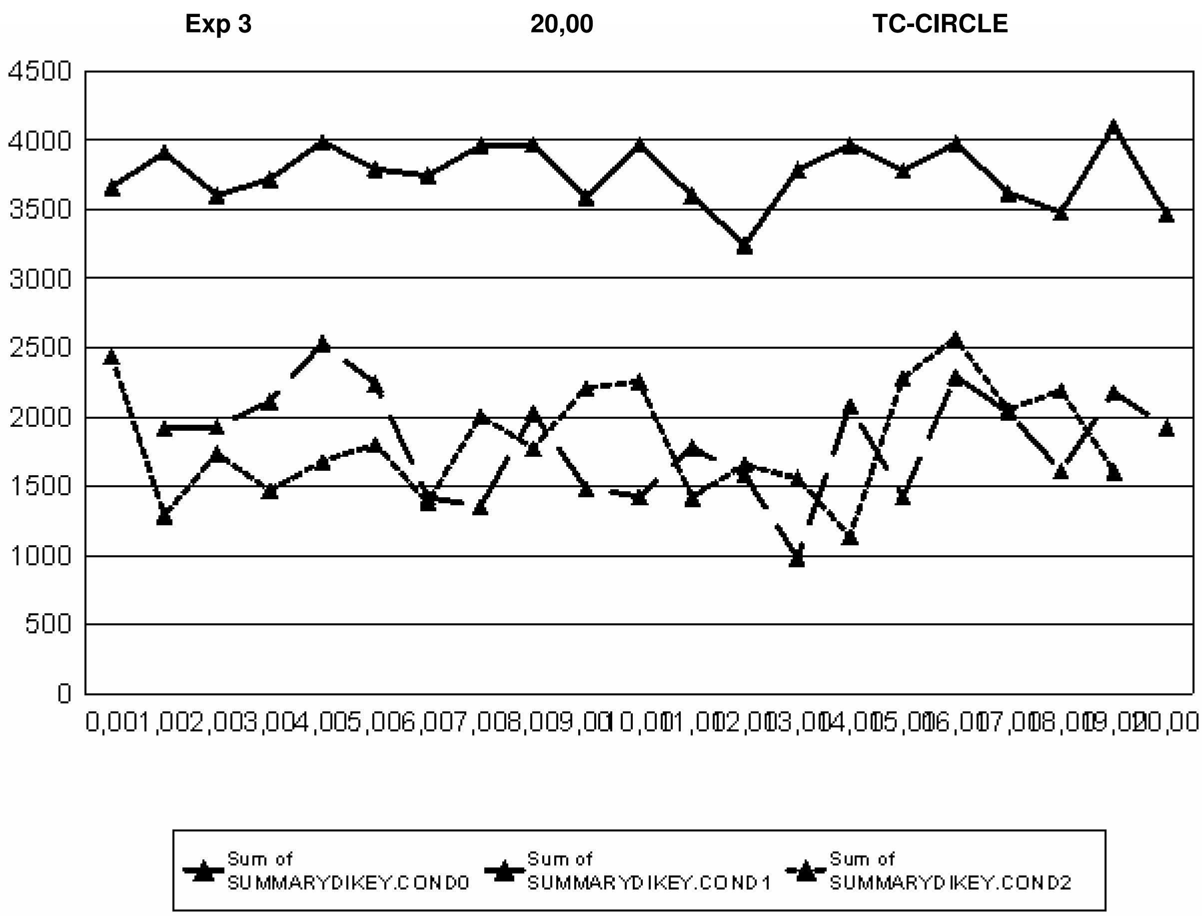


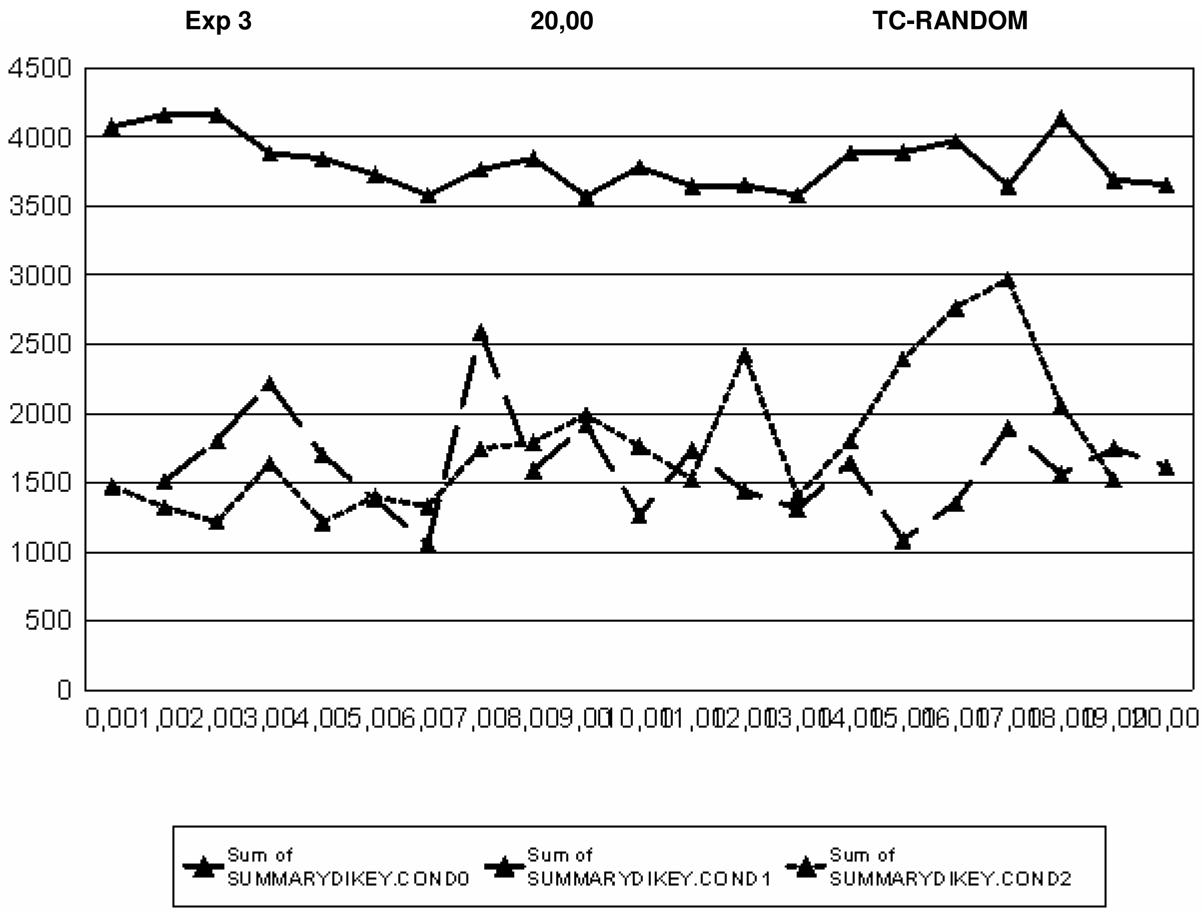
Exp 3 **5,00** **TC-RANDOM**




Exp 3
20,00
BR-CIRCLE
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0,00 1,00 2,00 3,00 4,00 5,00 6,00 7,00 8,00 9,00 10,00 11,00 12,00 13,00 14,00 15,00 16,00 17,00 18,00 19,00 20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 3
20,00
BR-RANDOM
5000
4000
3000
2000
1000
0
0,00 1,00 2,00 3,00 4,00 5,00 6,00 7,00 8,00 9,00 10,00 11,00 12,00 13,00 14,00 15,00 16,00 17,00 18,00 19,00 20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 3
20,00
TC-CIRCLE
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0,00
1,00
2,00
3,00
4,00
5,00
6,00
7,00
8,00
9,00
10,00
11,00
12,00
13,00
14,00
15,00
16,00
17,00
18,00
19,00
20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2


Exp 3
20,00
TC-RANDOM
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0,00
1,00
2,00
3,00
4,00
5,00
6,00
7,00
8,00
9,00
10,00
11,00
12,00
13,00
14,00
15,00
16,00
17,00
18,00
19,00
20,00
Sum of SUMMARYDIKEY.COND0
Sum of SUMMARYDIKEY.COND1
Sum of SUMMARYDIKEY.COND2

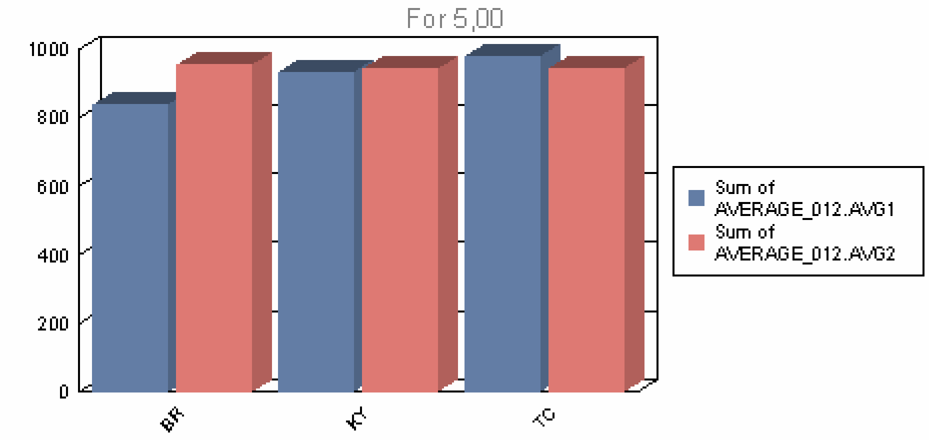
Appendix 4. Average Response times for each color.

This report shows the average search times for each different color.
ANOVA Analysis.

Appendix 4

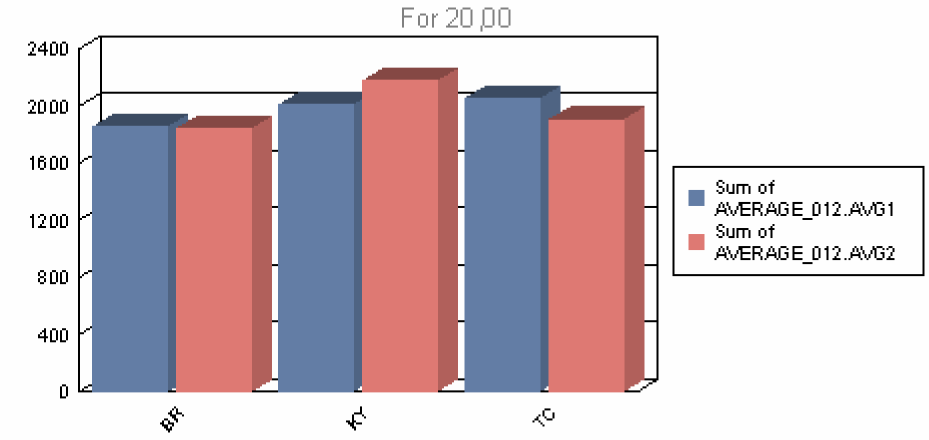
Color Comparison

Exp 1 **5,00**



COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	680	448	510	ms
BR	RANDOM	783	389	445	ms
KY	CIRCLE	761	484	486	ms
KY	RANDOM	735	450	457	ms
TC	CIRCLE	777	483	469	ms
TC	RANDOM	748	497	476	ms

Exp 1 **20,00**

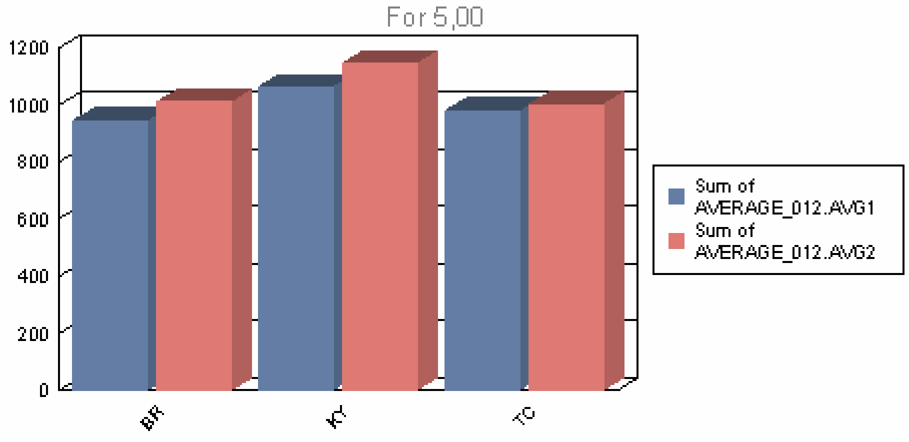


COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	2443	995	1007	ms
BR	RANDOM	2488	861	837	ms
KY	CIRCLE	2726	1076	1194	ms
KY	RANDOM	2738	933	984	ms
TC	CIRCLE	2576	1106	1011	ms
TC	RANDOM	2363	950	896	ms

Appendix 4

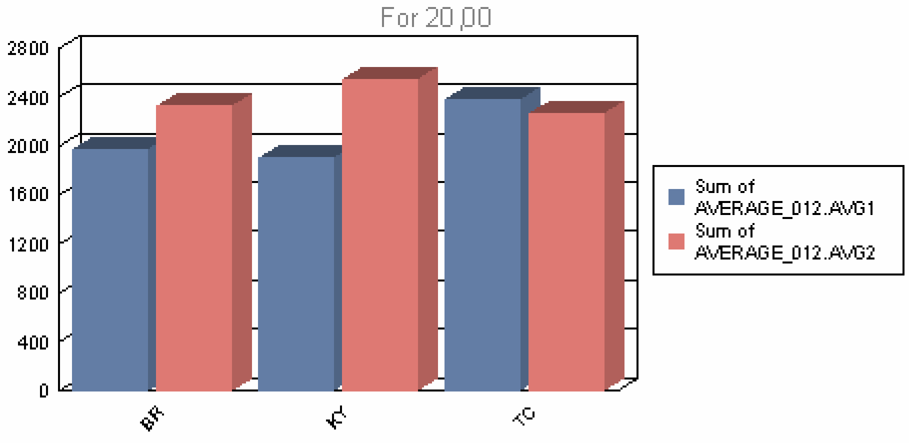
Color Comparison

Exp 2 **5,00**



COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	737	491	507	ms
BR	RANDOM	775	455	506	ms
KY	CIRCLE	886	495	568	ms
KY	RANDOM	835	571	580	ms
TC	CIRCLE	749	526	535	ms
TC	RANDOM	733	456	470	ms

Exp 2 **20,00**



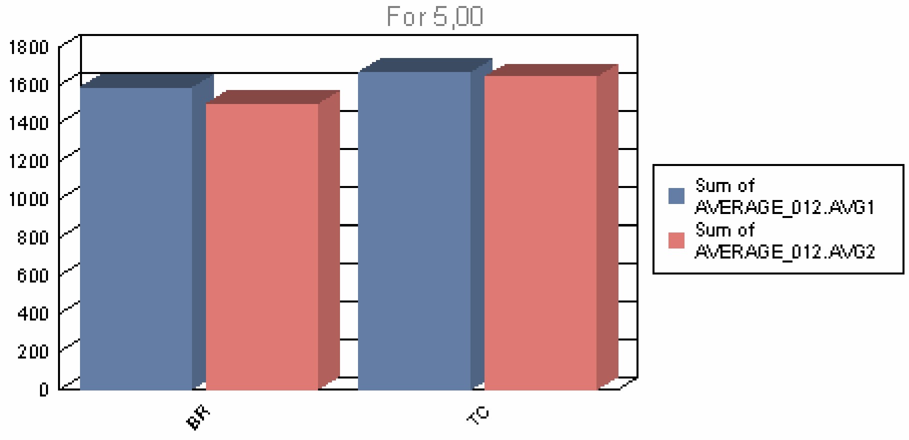
COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	2552	1135	1258	ms
BR	RANDOM	2513	841	1073	ms
KY	CIRCLE	2723	996	1326	ms
KY	RANDOM	2670	906	1223	ms
TC	CIRCLE	2751	1331	1256	ms

Appendix 4

Color Comparison

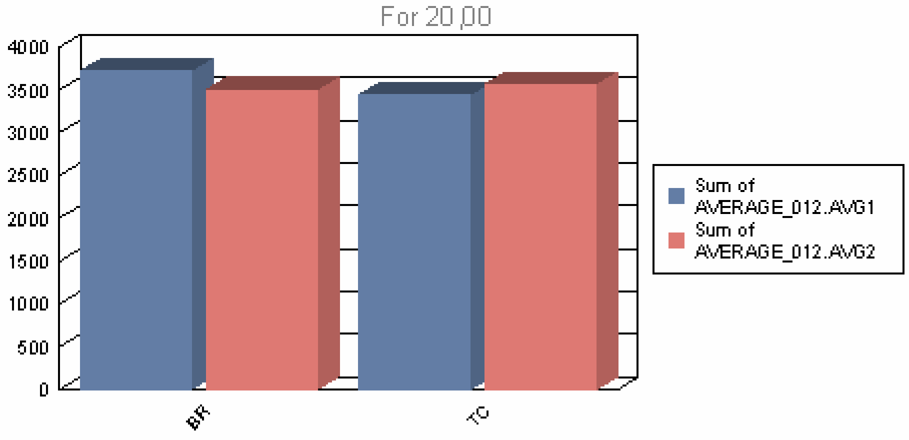
TC	RANDOM	2641	1052	1013	ms

Exp 3 **5,00**



COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	1402	868	730	ms
BR	RANDOM	1262	722	773	ms
TC	CIRCLE	1378	876	895	ms
TC	RANDOM	1469	794	752	ms

Exp 3 **20,00**



COLOR	LAYOUT	No Target	Color1	Color2	
BR	CIRCLE	3864	1771	1763	ms
BR	RANDOM	4034	1952	1743	ms
TC	CIRCLE	3757	1828	1816	ms
TC	RANDOM	3821	1617	1758	ms

ANOVA REACTION TIME BASED ON COLOR GROUPS.

TYPE = BR, CIRCLE, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	223120,053	1	223120,053	2,955	,088
Within Groups	11854443,318	157	75506,008		
Total	12077563,371	158			

a TYPE = BR, CIRCLE, 5, VERSION = Exp 1

TYPE = BR, CIRCLE, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	12442,017	1	12442,017	,244	,622
Within Groups	7429812,307	146	50889,125		
Total	7442254,324	147			

a TYPE = BR, CIRCLE, 5, VERSION = Exp 2

TYPE = BR, CIRCLE, 5, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	320292,637	1	320292,637	1,749	,190
Within Groups	13548332,100	74	183085,569		
Total	13868624,737	75			

a TYPE = BR, CIRCLE, 5, VERSION = Exp 3

TYPE = BR, CIRCLE, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	261797,148	1	261797,148	,693	,405
Within Groups	198995776,822	527	377601,095		
Total	199257573,970	528			

a TYPE = BR, CIRCLE, 20, VERSION = Exp 1

TYPE = BR, CIRCLE, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1771662,932	1	1771662,932	3,525	,061
Within Groups	249257753,946	496	502535,794		
Total	251029416,878	497			

a TYPE = BR, CIRCLE, 20, VERSION = Exp 2

TYPE = BR, CIRCLE, 20, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	13600,730	1	13600,730	,014	,906
Within Groups	400927140,748	412	973124,128		
Total	400940741,478	413			

a TYPE = BR, CIRCLE, 20, VERSION = Exp 3

TYPE = BR, RANDOM, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	130322,013	1	130322,013	2,949	,088
Within Groups	6496925,745	147	44196,774		
Total	6627247,758	148			

a TYPE = BR, RANDOM, 5, VERSION = Exp 1

TYPE = BR, RANDOM, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	85577,007	1	85577,007	1,758	,187
Within Groups	7547011,223	155	48690,395		
Total	7632588,229	156			

a TYPE = BR, RANDOM, 5, VERSION = Exp 2

TYPE = BR, RANDOM, 5, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	64693,136	1	64693,136	,365	,547
Within Groups	15248034,682	86	177302,729		
Total	15312727,818	87			

a TYPE = BR, RANDOM, 5, VERSION = Exp 3

TYPE = BR, RANDOM, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	387489,245	1	387489,245	1,148	,284
Within Groups	170394462,388	505	337414,777		
Total	170781951,633	506			

a TYPE = BR, RANDOM, 20, VERSION = Exp 1

TYPE = BR, RANDOM, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6668453,524	1	6668453,524	17,939	,000
Within Groups	212260751,642	571	371735,117		
Total	218929205,166	572			

a TYPE = BR, RANDOM, 20, VERSION = Exp 2

TYPE = BR, RANDOM, 20, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4179379,887	1	4179379,887	3,011	,084
Within Groups	514908108,966	371	1387892,477		
Total	519087488,853	372			

a TYPE = BR, RANDOM, 20, VERSION = Exp 3

TYPE = KY, CIRCLE, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1137,425	1	1137,425	,022	,882
Within Groups	8241023,360	161	51186,480		
Total	8242160,785	162			

a TYPE = KY, CIRCLE, 5, VERSION = Exp 1

TYPE = KY, CIRCLE, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	176584,207	1	176584,207	3,925	,049
Within Groups	7288964,549	162	44993,608		
Total	7465548,756	163			

a TYPE = KY, CIRCLE, 5, VERSION = Exp 2

TYPE = KY, CIRCLE, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3630010,215	1	3630010,215	7,979	,005
Within Groups	244772215,089	538	454966,943		
Total	248402225,304	539			

a TYPE = KY, CIRCLE, 20, VERSION = Exp 1

TYPE = KY, CIRCLE, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	17094442,263	1	17094442,263	40,678	,000
Within Groups	282818296,602	673	420235,210		
Total	299912738,865	674			

a TYPE = KY, CIRCLE, 20, VERSION = Exp 2

TYPE = KY, RANDOM, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	638,751	1	638,751	,015	,902
Within Groups	7247478,789	174	41652,177		
Total	7248117,540	175			

a TYPE = KY, RANDOM, 5, VERSION = Exp 1

TYPE = KY, RANDOM, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6692,265	1	6692,265	,072	,788
Within Groups	15155337,036	164	92410,592		
Total	15162029,301	165			

a TYPE = KY, RANDOM, 5, VERSION = Exp 2

TYPE = KY, RANDOM, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	395474,238	1	395474,238	,941	,332
Within Groups	258333757,914	615	420054,891		
Total	258729232,152	616			

a TYPE = KY, RANDOM, 20, VERSION = Exp 1

TYPE = KY, RANDOM, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	13387821,820	1	13387821,820	33,949	,000
Within Groups	226754481,784	575	394355,620		
Total	240142303,605	576			

a TYPE = KY, RANDOM, 20, VERSION = Exp 2

TYPE = TC, CIRCLE, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	99611,145	1	99611,145	1,629	,204
Within Groups	9906415,660	162	61150,714		
Total	10006026,805	163			

a TYPE = TC, CIRCLE, 5, VERSION = Exp 1

TYPE = TC, CIRCLE, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	902,747	1	902,747	,014	,905
Within Groups	10446972,458	164	63701,052		
Total	10447875,205	165			

a TYPE = TC, CIRCLE, 5, VERSION = Exp 2

TYPE = TC, CIRCLE, 5, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9063,188	1	9063,188	,053	,819
Within Groups	14294990,224	83	172228,798		
Total	14304053,412	84			

a TYPE = TC, CIRCLE, 5, VERSION = Exp 3

TYPE = TC, CIRCLE, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	442091,657	1	442091,657	,830	,363
Within Groups	326376686,580	613	532425,264		
Total	326818778,237	614			

a TYPE = TC, CIRCLE, 20, VERSION = Exp 1

TYPE = TC, CIRCLE, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	815848,744	1	815848,744	1,605	,206
Within Groups	317147250,127	624	508248,798		
Total	317963098,871	625			

a TYPE = TC, CIRCLE, 20, VERSION = Exp 2

TYPE = TC, CIRCLE, 20, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1336,517	1	1336,517	,001	,975
Within Groups	395694227,062	300	1318980,757		
Total	395695563,580	301			

a TYPE = TC, CIRCLE, 20, VERSION = Exp 3

TYPE = TC, RANDOM, 5, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	10798,761	1	10798,761	,171	,680
Within Groups	9801644,831	155	63236,418		
Total	9812443,592	156			

a TYPE = TC, RANDOM, 5, VERSION = Exp 1

TYPE = TC, RANDOM, 5, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	12370,709	1	12370,709	,364	,548
Within Groups	4832328,179	142	34030,480		
Total	4844698,889	143			

a TYPE = TC, RANDOM, 5, VERSION = Exp 2

TYPE = TC, RANDOM, 5, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	42328,409	1	42328,409	,214	,645
Within Groups	17039503,909	86	198133,766		
Total	17081832,318	87			

a TYPE = TC, RANDOM, 5, VERSION = Exp 3

TYPE = TC, RANDOM, 20, VERSION = Exp 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	132362,052	1	132362,052	,316	,575
Within Groups	212269900,111	506	419505,731		
Total	212402262,163	507			

a TYPE = TC, RANDOM, 20, VERSION = Exp 1

TYPE = TC, RANDOM, 20, VERSION = Exp 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	150005,540	1	150005,540	,349	,555
Within Groups	245142635,318	570	430074,799		
Total	245292640,858	571			

a TYPE = TC, RANDOM, 20, VERSION = Exp 2

TYPE = TC, RANDOM, 20, VERSION = Exp 3

ANOVA(a)

REACTION_TIME

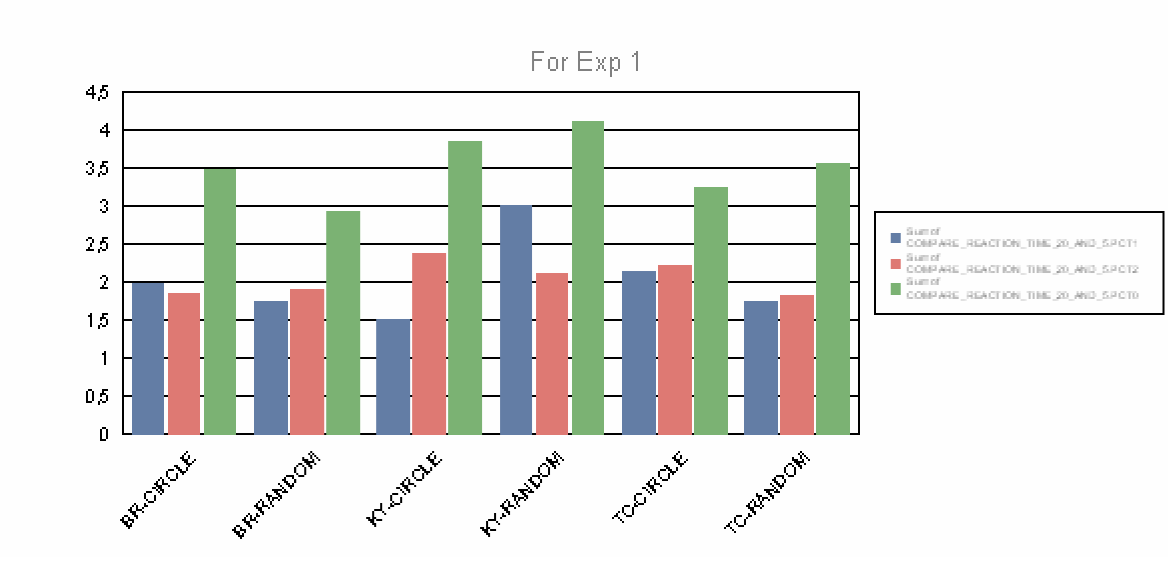
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2211325,571	1	2211325,571	1,755	,186
Within Groups	440948255,884	350	1259852,160		
Total	443159581,455	351			

a TYPE = TC, RANDOM, 20, VERSION = Exp 3

Appendix 5. Comparison of Reaction Times for 20 and 5 letter displays.

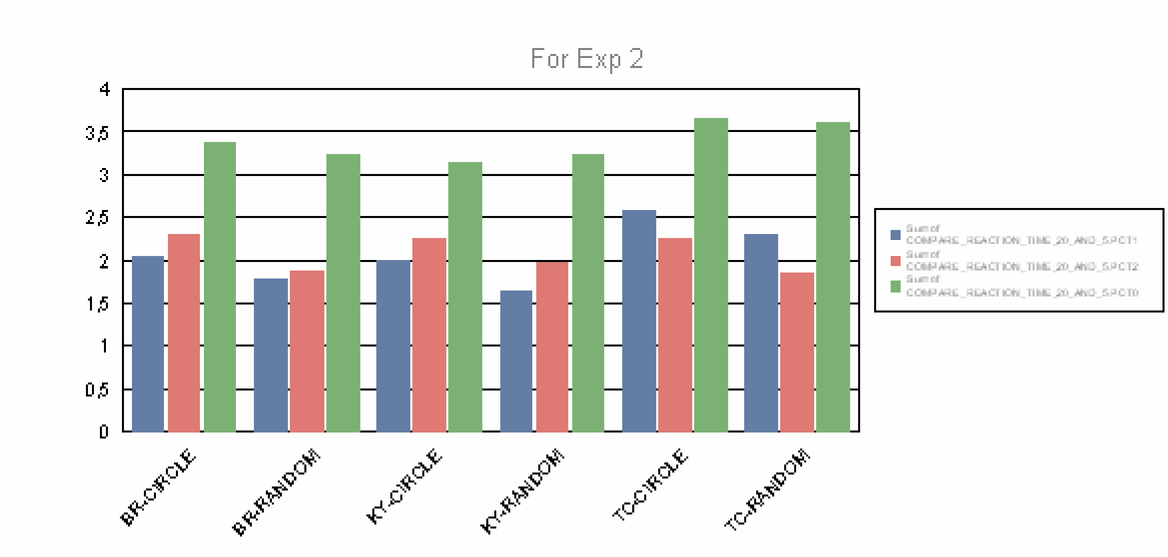
This report gives a comparison of 20-letter and 5-letter search times. The average time for Color1, Color2 and No-target conditions are displayed.

Appendix 5



Average V. Search Time (20 letter / 5 Letter)

Expriment Layout	No-Target	Color 1	Color 2
BR-CIRCLE	3,49	1,98	1,85
BR-RANDOM	2,95	1,76	1,92
KY-CIRCLE	3,87	1,50	2,38
KY-RANDOM	4,13	3,02	2,12
TC-CIRCLE	3,26	2,14	2,23
TC-RANDOM	3,56	1,75	1,83
Average	**3,54**	**2,03**	**2,05**

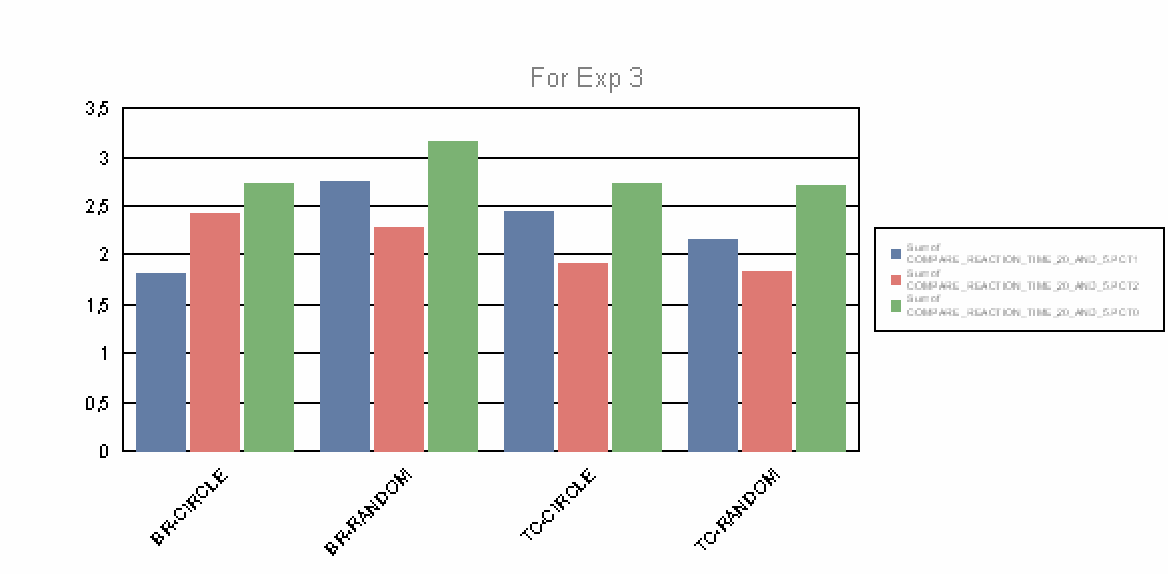


Average V. Search Time (20 letter / 5 Letter)

Expriment Layout	No-Target	Color 1	Color 2
BR-CIRCLE	3,39	2,06	2,30
BR-RANDOM	3,24	1,79	1,88
KY-CIRCLE	3,14	2,01	2,25
KY-RANDOM	3,25	1,66	1,98
TC-CIRCLE	3,65	2,59	2,26
TC-RANDOM	3,63	2,32	1,87

Appendix 5

Average	**3,38**	**2,07**	**2,09**



Average V. Search Time (20 letter / 5 Letter)

Expriment Layout	No-Target	Color 1	Color 2
BR-CIRCLE	2,74	1,80	2,43
BR-RANDOM	3,17	2,76	2,29
TC-CIRCLE	2,73	2,45	1,92
TC-RANDOM	2,71	2,16	1,83
Average	**2,84**	**2,29**	**2,12**

Appendix 6. The valid experiment count

The number of valid experiments for each setup is listed.

Appendix 6 **Number of valid experiments per experiment setup**

Exp 1

		Set Size	# of valid experiments
BR	CIRCLE	5	13
KY	CIRCLE	5	14
TC	CIRCLE	5	13
BR	CIRCLE	20	13
KY	CIRCLE	20	13
TC	CIRCLE	20	15
BR	RANDOM	5	13
KY	RANDOM	5	15
TC	RANDOM	5	13
BR	RANDOM	20	12
KY	RANDOM	20	16
TC	RANDOM	20	12

Exp 2

		Set Size	# of valid experiments
BR	CIRCLE	5	15
KY	CIRCLE	5	17
TC	CIRCLE	5	17
BR	CIRCLE	20	13
KY	CIRCLE	20	18
TC	CIRCLE	20	16
BR	RANDOM	5	16
KY	RANDOM	5	17
TC	RANDOM	5	15
BR	RANDOM	20	15
KY	RANDOM	20	15
TC	RANDOM	20	15

Exp 3

		Set Size	# of valid experiments
BR	CIRCLE	5	8
TC	CIRCLE	5	9
BR	CIRCLE	20	11
TC	CIRCLE	20	8
BR	RANDOM	5	9
TC	RANDOM	5	9
BR	RANDOM	20	10
TC	RANDOM	20	10

Appendix 7. Comparison of Circle and Random Layouts

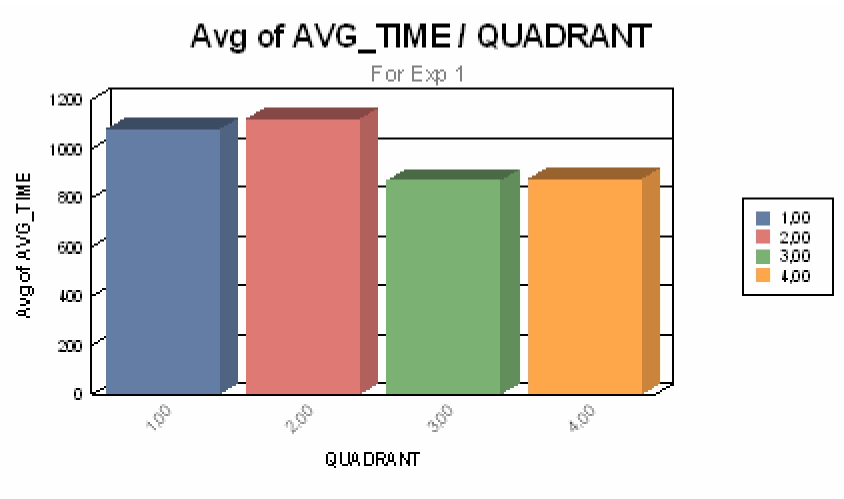
A comparison of all cases for each experiment is given followed by a more detailed list based on each experiment setup.

		CIRCLE			**RANDOM**			**Random / Circle Ratio**		
Experiment & Set size		Color1	Color2	No Target	Color1	Color2	No Target	C1Chg	C2 Chg	NT Chg
Exp 1	5	486	468	737	439	456	734	0,90	0,98	0,99
Exp 1	20	986	1163	2586	930	946	2551	0,94	0,81	0,99
Exp 2	5	505	533	791	490	517	782	0,97	0,97	0,99
Exp 2	20	1150	1273	2658	930	1095	2611	0,81	0,86	0,98
Exp 3	5	867	816	1393	758	760	1366	0,87	0,93	0,98
Exp 3	20	1782	1793	3814	1790	1758	3929	1,00	0,98	1,03

		Random / Circle Ratio		
Compared color&Layouts	Experiment	Color1	Color2	No Target
5,00				
BR-CIRCLEvs BR-RANDOM	Exp 1	0,83	1,15	1,16
KY-CIRCLEvs KY-RANDOM	Exp 1	0,86	0,84	0,86
TC-CIRCLEvs TC-RANDOM	Exp 1	1,02	0,93	0,98
BR-CIRCLEvs BR-RANDOM	Exp 2	0,93	0,99	1,05
KY-CIRCLEvs KY-RANDOM	Exp 2	1,15	1,04	0,95
TC-CIRCLEvs TC-RANDOM	Exp 2	0,84	0,88	0,98
BR-CIRCLEvs BR-RANDOM	Exp 3	0,84	1,05	0,90
TC-CIRCLEvs TC-RANDOM	Exp 3	0,91	0,84	1,06
20,00				
BR-CIRCLEvs BR-RANDOM	Exp 1	0,79	0,83	1,02
KY-CIRCLEvs KY-RANDOM	Exp 1	1,17	0,70	1,01
TC-CIRCLEvs TC-RANDOM	Exp 1	0,88	0,95	0,93
BR-CIRCLEvs BR-RANDOM	Exp 2	0,74	0,84	0,99
KY-CIRCLEvs KY-RANDOM	Exp 2	0,91	0,92	0,98
TC-CIRCLEvs TC-RANDOM	Exp 2	0,79	0,81	0,97
BR-CIRCLEvs BR-RANDOM	Exp 3	1,12	0,98	1,04
TC-CIRCLEvs TC-RANDOM	Exp 3	0,89	0,98	1,02

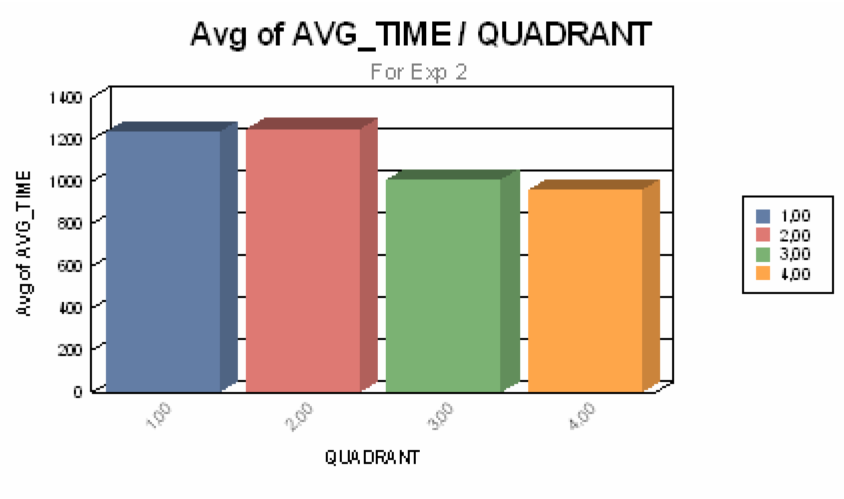
Appendix 8. Analysis of Response Times By location

This report lists the average response time for each layout and for each color based on the quadrant. The right bottom quadrant is #1, left bottom is #2, left upper is #3 and right upper is #4.



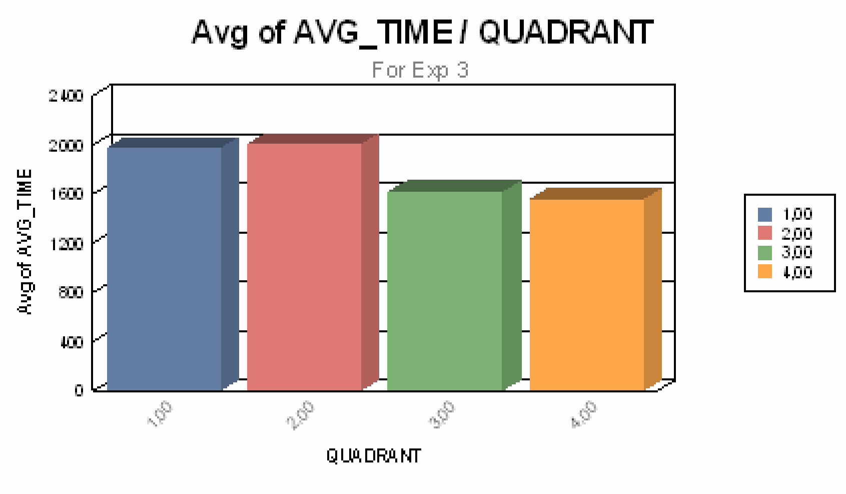
LAPTOP **Exp 1**

EXPERIMENT	COLOR	QUADRANT	AVG_TIME
Exp 1	1	1	1084
Exp 1	1	2	1100
Exp 1	1	3	856
Exp 1	1	4	878
Exp 1	2	1	1072
Exp 1	2	2	1142
Exp 1	2	3	885
Exp 1	2	4	875



LAPTOP **Exp 2**

EXPERIMENT	COLOR	QUADRANT	AVG_TIME
Exp 2	1	1	1160
Exp 2	1	2	1200
Exp 2	1	3	943
Exp 2	1	4	862
Exp 2	2	1	1313
Exp 2	2	2	1299
Exp 2	2	3	1077
Exp 2	2	4	1057



	LAPTOP	Exp 3	
EXPERIMENT	COLOR	QUADRANT	AVG_TIME
Exp 3	1	1	1985
Exp 3	1	2	1984
Exp 3	1	3	1632
Exp 3	1	4	1575
Exp 3	2	1	1967
Exp 3	2	2	2032
Exp 3	2	3	1606
Exp 3	2	4	1542

ANOVA Analysis for 20 Letter Experiments Reaction times based on Quadrants

TYPE = BR, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3749287,188	3	1249762,396	3,259	,022
Within Groups	96640423,714	252	383493,745		
Total	100389710,902	255			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

TYPE = BR, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3572279,280	3	1190759,760	3,371	,019
Within Groups	95033786,639	269	353285,452		
Total	98606065,919	272			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

TYPE = BR, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8443148,700	3	2814382,900	7,071	,000
Within Groups	99899376,570	251	398005,484		
Total	108342525,271	254			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

TYPE = BR, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4135447,150	3	1378482,383	2,409	,068
Within Groups	136779781,525	239	572300,341		
Total	140915228,675	242			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

TYPE = BR, CIRCLE , 20, VERSION = Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	4728536,915	3	1576178,972	1,468	,225
Within Groups	211513093,035	197	1073670,523		
Total	216241629,950	200			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 3 , COLOR = 1

TYPE = BR, CIRCLE , 20, VERSION = Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8130744,100	3	2710248,033	3,208	,024
Within Groups	176554766,698	209	844759,649		
Total	184685510,798	212			

a TYPE = BR, CIRCLE , 20, VERSION = Exp 3 , COLOR = 2

TYPE = BR, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3659475,850	3	1219825,283	2,965	,033
Within Groups	95843186,386	233	411344,148		
Total	99502662,236	236			

a TYPE = BR, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

TYPE = BR, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5217853,863	3	1739284,621	7,045	,000
Within Groups	65673946,288	266	246894,535		
Total	70891800,152	269			

a TYPE = BR, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

TYPE = BR, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5803914,609	3	1934638,203	8,096	,000
Within Groups	69063263,261	289	238973,229		
Total	74867177,870	292			

a TYPE = BR, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

TYPE = BR, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8232416,051	3	2744138,684	5,864	,001
Within Groups	129161157,720	276	467975,209		
Total	137393573,771	279			

a TYPE = BR, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

TYPE = BR, RANDOM , 20, VERSION = Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	17412393,458	3	5804131,153	3,678	,013
Within Groups	280917164,262	178	1578186,316		
Total	298329557,720	181			

a TYPE = BR, RANDOM , 20, VERSION = Exp 3 , COLOR = 1

TYPE = BR, RANDOM , 20, VERSION = Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6477846,313	3	2159282,104	1,922	,128
Within Groups	210100704,933	187	1123533,181		
Total	216578551,246	190			

a TYPE = BR, RANDOM , 20, VERSION = Exp 3 , COLOR = 2

TYPE = KY, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6103213,292	3	2034404,431	5,378	,001
Within Groups	95705651,097	253	378283,206		
Total	101808864,389	256			

a TYPE = KY, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

TYPE = KY, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3017082,826	3	1005694,275	2,005	,113
Within Groups	139946267,874	279	501599,526		
Total	142963350,700	282			

a TYPE = KY, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

TYPE = KY, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	13410190,777	3	4470063,592	13,914	,000
Within Groups	111803488,995	348	321274,394		
Total	125213679,773	351			

a TYPE = KY, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

TYPE = KY, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2071674,055	3	690558,018	1,416	,238
Within Groups	155532942,775	319	487564,084		
Total	157604616,830	322			

a TYPE = KY, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

TYPE = KY, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5284658,012	3	1761552,671	4,196	,006
Within Groups	129292347,293	308	419780,348		
Total	134577005,305	311			

a TYPE = KY, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

TYPE = KY, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1453702,735	3	484567,578	1,193	,313
Within Groups	122303049,875	301	406322,425		
Total	123756752,610	304			

a TYPE = KY, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

TYPE = KY, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3996431,312	3	1332143,771	4,860	,003
Within Groups	79221172,491	289	274121,704		
Total	83217603,802	292			

a TYPE = KY, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

TYPE = KY, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	817963,962	3	272654,654	,535	,659
Within Groups	142718914,021	280	509710,407		
Total	143536877,982	283			

a TYPE = KY, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

TYPE = TC, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9027132,865	3	3009044,288	5,206	,002
Within Groups	180336796,283	312	578002,552		
Total	189363929,149	315			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 1 , COLOR = 1

TYPE = TC, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	17783516,824	3	5927838,941	14,667	,000
Within Groups	119229240,607	295	404166,917		
Total	137012757,431	298			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 1 , COLOR = 2

TYPE = TC, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	14927096,814	3	4975698,938	10,069	,000
Within Groups	153193537,928	310	494172,703		
Total	168120634,742	313			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 2 , COLOR = 1

TYPE = TC, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	22428585,569	3	7476195,190	18,189	,000
Within Groups	126598029,815	308	411032,564		
Total	149026615,385	311			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 2 , COLOR = 2

TYPE = TC, CIRCLE , 20, VERSION = Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1807372,049	3	602457,350	,475	,700
Within Groups	190235097,587	150	1268233,984		
Total	192042469,636	153			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 3 , COLOR = 1

TYPE = TC, CIRCLE , 20, VERSION = Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5399159,306	3	1799719,769	1,307	,274
Within Groups	198252598,119	144	1376754,154		
Total	203651757,426	147			

a TYPE = TC, CIRCLE , 20, VERSION = Exp 3 , COLOR = 2

TYPE = TC, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3607576,290	3	1202525,430	2,470	,063
Within Groups	116824535,231	240	486768,897		
Total	120432111,521	243			

a TYPE = TC, RANDOM , 20, VERSION = Exp 1 , COLOR = 1

TYPE = TC, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2957284,798	3	985761,599	2,884	,036
Within Groups	88880503,793	260	341848,092		
Total	91837788,591	263			

a TYPE = TC, RANDOM , 20, VERSION = Exp 1 , COLOR = 2

TYPE = TC, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6527061,426	3	2175687,142	5,070	,002
Within Groups	121003331,441	282	429089,828		
Total	127530392,867	285			

a TYPE = TC, RANDOM , 20, VERSION = Exp 2 , COLOR = 1

TYPE = TC, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	6557891,759	3	2185963,920	5,551	,001
Within Groups	111054350,692	282	393809,754		
Total	117612242,451	285			

a TYPE = TC, RANDOM , 20, VERSION = Exp 2 , COLOR = 2

TYPE = TC, RANDOM , 20, VERSION = Exp 3 , COLOR = 1

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	10066746,390	3	3355582,130	2,923	,035
Within Groups	204329807,588	178	1147920,267		
Total	214396553,978	181			

a TYPE = TC, RANDOM , 20, VERSION = Exp 3 , COLOR = 1

TYPE = TC, RANDOM , 20, VERSION = Exp 3 , COLOR = 2

ANOVA(a)

REACTION_TIME

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	22506756,773	3	7502252,258	6,103	,001
Within Groups	204044945,133	166	1229186,416		
Total	226551701,906	169			

a TYPE = TC, RANDOM , 20, VERSION = Exp 3 , COLOR = 2

Appendix 9. ACT-R/PM models for Experiment 2/Two Colors/Circle

5 Letters Model and Output of an Example Run

20 Letters Model and Output of an Example Run

OUTPUT OF A MODEL RUN (5 LETTERS - NO-TARGET)

```
Time  0.000: Vision found LOC45
Time  0.000: Find-Unattended-Letter Selected
Time  0.050: Find-Unattended-Letter Fired
Time  0.050: Module :VISION running command FIND-LOCATION
Time  0.050: Vision found LOC46
Time  0.050: Attend-Letter Selected
Time  0.100: Attend-Letter Fired
Time  0.100: Module :VISION running command MOVE-ATTENTION
Time  0.185: Module :VISION running command ENCODING-COMPLETE
Time  0.185: Vision sees TEXT40
Time  0.185: Encode-Letter Selected
Time  0.235: Encode-Letter Fired
Time  0.235: Evaluate-Letter-Notg Selected
Time  0.285: Evaluate-Letter-Notg Fired
Time  0.285: Decide1 Selected
Time  0.335: Decide1 Fired
Time  0.335: Module :VISION running command FIND-LOCATION
Time  0.335: Vision found LOC48
Time  0.335: Attend-Letter Selected
Time  0.385: Attend-Letter Fired
Time  0.385: Module :VISION running command MOVE-ATTENTION
Time  0.470: Module :VISION running command ENCODING-COMPLETE
Time  0.470: Vision sees TEXT44
Time  0.470: Encode-Letter Selected
Time  0.520: Encode-Letter Fired
Time  0.520: Evaluate-Letter-Notg Selected
Time  0.570: Evaluate-Letter-Notg Fired
Time  0.570: Decide1 Selected
Time  0.620: Decide1 Fired
Time  0.620: Module :VISION running command FIND-LOCATION
Time  0.620: Vision found LOC45
Time  0.620: Attend-Letter Selected
Time  0.670: Attend-Letter Fired
Time  0.670: Module :VISION running command MOVE-ATTENTION
Time  0.755: Module :VISION running command ENCODING-COMPLETE
Time  0.755: Vision sees TEXT43
Time  0.755: Encode-Letter Selected
Time  0.805: Encode-Letter Fired
Time  0.805: Evaluate-Letter-Notg Selected
Time  0.855: Evaluate-Letter-Notg Fired
Time  0.855: Decide1 Selected
Time  0.905: Decide1 Fired
Time  0.905: Module :VISION running command FIND-LOCATION
Time  0.905: Vision found LOC51
Time  0.905: Attend-Letter Selected
Time  0.955: Attend-Letter Fired
Time  0.955: Module :VISION running command MOVE-ATTENTION
Time  1.040: Module :VISION running command ENCODING-COMPLETE
Time  1.040: Vision sees TEXT42
Time  1.040: Encode-Letter Selected
Time  1.090: Encode-Letter Fired
Time  1.090: Evaluate-Letter-Notg Selected
Time  1.140: Evaluate-Letter-Notg Fired
Time  1.140: Decide1 Selected
Time  1.190: Decide1 Fired
Time  1.190: Module :VISION running command FIND-LOCATION
Time  1.190: Vision found LOC53
Time  1.190: Attend-Letter Selected
Time  1.240: Attend-Letter Fired
Time  1.240: Module :VISION running command MOVE-ATTENTION
Time  1.325: Module :VISION running command ENCODING-COMPLETE
```

```
 Time  1.325: Vision sees TEXT41
 Time  1.325: Encode-Letter Selected
 Time  1.375: Encode-Letter Fired
 Time  1.375: Evaluate-Letter-Notg Selected
 Time  1.425: Evaluate-Letter-Notg Fired
 Time  1.425: Decide1 Selected
 Time  1.475: Decide1 Fired
 Time  1.475: Module :VISION running command FIND-LOCATION
 Time  1.475: Decide2 Selected
 Time  1.525: Decide2 Fired
 Time  1.525: Respond-Notfound Selected
 Time  1.575: Respond-Notfound Fired
 Time  1.575: Module :MOTOR running command PRESS-KEY
 Time  1.725: Module :MOTOR running command PREPARATION-COMPLETE
 Time  1.775: Module :MOTOR running command INITIATION-COMPLETE
 Time  1.785: Device running command OUTPUT-KEY

<< Window "Letter Recognition" got key #\l at time 1785 >>

 Time  1.870: Module :VISION running command ENCODING-COMPLETE
 Time  1.875: Module :MOTOR running command FINISH-MOVEMENT
 Time  1.875: Checking for silent events.
 Time  1.875: * Nothing to run:  No productions, no events.
"l"
```

OUTPUT OF A MODEL RUN (5 LETTERS -TARGET)

```
 Time  0.000: Vision found LOC65
 Time  0.000: Find-Unattended-Letter Selected
 Time  0.050: Find-Unattended-Letter Fired
 Time  0.050: Module :VISION running command FIND-LOCATION
 Time  0.050: Vision found LOC66
 Time  0.050: Attend-Letter Selected
 Time  0.100: Attend-Letter Fired
 Time  0.100: Module :VISION running command MOVE-ATTENTION
 Time  0.185: Module :VISION running command ENCODING-COMPLETE
 Time  0.185: Vision sees TEXT61
 Time  0.185: Encode-Letter Selected
 Time  0.235: Encode-Letter Fired
 Time  0.235: Evaluate-Letter-Notg Selected
 Time  0.285: Evaluate-Letter-Notg Fired
 Time  0.285: Decide1 Selected
 Time  0.335: Decide1 Fired
 Time  0.335: Module :VISION running command FIND-LOCATION
 Time  0.335: Vision found LOC68
 Time  0.335: Attend-Letter Selected
 Time  0.385: Attend-Letter Fired
 Time  0.385: Module :VISION running command MOVE-ATTENTION
 Time  0.470: Module :VISION running command ENCODING-COMPLETE
 Time  0.470: Vision sees TEXT62
 Time  0.470: Encode-Letter Selected
 Time  0.520: Encode-Letter Fired
 Time  0.520: Evaluate-Letter-G Selected
 Time  0.570: Evaluate-Letter-G Fired
 Time  0.570: Respond-Found Selected
 Time  0.620: Respond-Found Fired
 Time  0.620: Module :MOTOR running command PRESS-KEY
 Time  0.770: Module :MOTOR running command PREPARATION-COMPLETE
 Time  0.820: Module :MOTOR running command INITIATION-COMPLETE
 Time  0.830: Device running command OUTPUT-KEY

<< Window "Letter Recognition" got key #\a at time 830 >>

 Time  0.915: Module :VISION running command ENCODING-COMPLETE
 Time  0.920: Module :MOTOR running command FINISH-MOVEMENT
 Time  0.920: Checking for silent events.
 Time  0.920: * Nothing to run:  No productions, no events.
"a"
```

EXPERİMENT AND MODEL for 5 LETTER TWO COLOR

```
(defvar *response* nil)

 (defun do-experiment ()
  (if *actr-enabled-p*
```

```
      (do-experiment-model)
    (do-experiment-person)))


(defun do-experiment-person ()

 (let* ((lis (permute-list '("B" "A" "D" "F" "O" "H"
                              "J" "K" "L" "M" "N" "P"
                              "Q" "R" "S" "T" "V" "W"
                              "X" "Y" "Z")))
         (text1 (first lis))
         (lis2 (permute-list lis))
         (text2 (first lis2))
         (lis3 (permute-list lis))
         (text3 (first lis3))
         (lis4 (permute-list lis))
         (text4 (first lis4))
        (lis5 (permute-list lis))
         (text5 (first lis5))
         (lis6 (permute-list '("C" "G")))
         (target (first lis6))



         (window (open-exp-window "Letter Recognition" :x -5 :y -5 :width 1500
:height 900 )))
      (case (random 5)
               (0 (setf text1 target))
               (1 (setf text2 target))
               (2 (setf text3 target))
               (3 (setf text4 target))
               (4 (setf text5 target)))
    (add-text-to-exp-window :text text1 :x 350 :y 267 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text2 :x 574 :y 196 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text3 :x 711 :y 386 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text4 :x 575 :y 570 :color (first(permute-list
'(green red))))
    (add-text-to-exp-window :text text5 :x 350 :y 503 :color (first(permute-list
'(green red))))


    (setf *response* nil)

    (while (null *response*)
      (allow-event-manager window))

    *response*))

(defun do-experiment-model ()

(let* ((lis (permute-list '("B" "A" "D" "F" "O" "H"
                              "J" "K" "L" "M" "N" "P"
                              "Q" "R" "S" "T" "V" "W"
                              "X" "Y" "Z")))
         (text1 (first lis))
         (lis2 (permute-list lis))
         (text2 (first lis2))
         (lis3 (permute-list lis))
         (text3 (first lis3))
         (lis4 (permute-list lis))
         (text4 (first lis4))
        (lis5 (permute-list lis))
         (text5 (first lis5))
         (lis6 (permute-list '("C" "G")))
         (target (first lis6))



         (window (open-exp-window "Letter Recognition" :x -5 :y -5 :width 1500
:height 900 )))
      (case (random 5)
               (0 (setf text1 target))
```

```
               (1 (setf text2 target))
               (2 (setf text3 target))
               (3 (setf text4 target))
               (4 (setf text5 target)))

    (add-text-to-exp-window :text text1 :x 350 :y 267 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text2 :x 574 :y 196 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text3 :x 711 :y 386 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text4 :x 575 :y 570 :color (first(permute-list
'(blue red))))
    (add-text-to-exp-window :text text5 :x 350 :y 503 :color (first(permute-list
'(blue red))))

        (reset)
    (pm-install-device window)
    (pm-proc-display)
    (pm-set-params :real-time t
               :visual-num-finsts 6 :visual-finst-span 10)

    (setf *response* nil)

    (pm-run 10)

    *response*))

(defmethod rpm-window-key-event-handler ((win rpm-window) key)
  (setf *response* (string key))
  (clear-exp-window)
  (when *actr-enabled-p* (pm-proc-display))) (clear-all)
(pm-reset)

(chunk-type read-letters letter state)


(add-dm
 (goal isa read-letters state start))


(P find-unattended-letter
   =goal>
      ISA         read-letters
      state       start
 ==>
   +visual-location>
      ISA         visual-location
      attended    nil
   =goal>
      state       find-location
)

(P attend-letter
   =goal>
      ISA         read-letters
      state       find-location
   =visual-location>
      ISA         visual-location
   =visual-state>
      ISA         module-state
      modality    free
==>
   +visual>
      ISA         visual-object
      screen-pos  =visual-location
   =goal>
      state       attend
)


(P encode-letter
   =goal>
      ISA         read-letters
      state       attend
```

```
   =visual>
      ISA         text
      value       =letter
==>
   =goal>
      letter      =letter
      state       evaluate
)


(P evaluate-letter-G
   =goal>
      ISA         read-letters
      state       evaluate
   =visual>
      ISA         text
      value       "g"
==>
   =goal>
      state       respond-yes
)


(P evaluate-letter-NotG
   =goal>
      ISA         read-letters
      state       evaluate
   =visual>
      ISA         text
    -  value       "g"
==>
   =goal>
      letter      =letter
      state       decide-on-no-g
)

(P decide1
   =goal>
     ISA    read-letters
     state  decide-on-no-g
==>
   +visual-location>
     ISA   visual-location
     attended    nil
  =goal>
    state  find-location

)

(P decide2
  =goal>
    ISA    read-letters
    state  find-location
   =visual-location>
    ISA    error
==>
   =goal>
    state respond-no
)

(P respond-found
   =goal>
      ISA         read-letters
      state       respond-yes
   =manual-state>
      ISA         module-state
      modality    free
==>
   +manual>
      ISA         press-key
      key         "a"
   =goal>
      state       stop
)
```

```
(P respond-notfound
   =goal>
      ISA         read-letters
      letter      =letter
      state       respond-no
   =manual-state>
      ISA         module-state
      modality    free
==>
   +manual>
      ISA         press-key
      key         "l"
   =goal>
      state       stop
)

(sgp :v t)

(pm-set-params :real-time t :show-focus t)

(goal-focus goal)

(setf *actr-enabled-p* t)
```