Master Thesis
Content and Knowledge Engineering

Team task allocation support by using the human body as a sensor

Real-time estimated stress of team members as input for decision support for military commanders

Corian van der Steen (3143899)
corianvdsteen@gmail.com
June 25, 2012

External Supervisor:
dr. P.-P. van Maanen
TNO, Perceptual and Cognitive Systems
Kampweg 5
3769 ZG Soesterberg

Internal Supervisors:
dr. ir. R.-J. Beun
dr. H.H.A.M. Prüst
Universiteit Utrecht
Department of Information and Computing Sciences

Abstract

The quality of decision making by military commanders can be enhanced by providing extra information. Modern military equipment can be expanded by devices that monitor the condition of soldiers. Because the condition of a soldier influences his behavior and thereby his performance, this can be considered valuable input for the commander in making strategic decisions. The purpose of this study is to investigate whether the quality of decision making of a military commander improves when he can monitor the condition of the members of his team.

An experiment is carried out with 18 participants. Per session, five team members play a virtual first person shooter game where they have to work together to eliminate enemies. By using interbeat intervals, a stress model estimates the stress value of each team member real-time, as an indication of the condition of the team member. One commander has to allocate the team members to one of three tasks. To provide the team leader with the information, a decision support system (DSS) has been designed and built. This DSS presents the information visually and can give advice to the team leader in making decisions. The team leader can get three types of support: a) no support, b) stress information support, c) stress information and decision advice support. A fourth condition is added to check the quality of decision making by the model individually (without human interference).

The purpose of the experiment can be divided in two areas: 1) to test the relationship between the stress value of team members and their performance within the experiment, and 2) to test the effect of different types of support on the quality of decision making of the team leader. Firstly, we expect that the stress value is a negative predictor for performance. Secondly, we expect that a team leader makes better decisions when he can consider the advice from the DSS, as opposed to making the decision based on solely verbal communication with the team members, and only stress information support.

The analyses show that no clear relationship is found between the stress value and the performance of the team members, although trends indicate that the severeness of stress plays a role in this relationship. Though the DSS model bases the advice on this assumption, the model individually yielded better quality of decisions than the human team leader with both stress information and decision advice support. The subjective support effectivity results show that the team leaders prefer the decision advice support. All other relationships between support effectivity and conditions were not significant.

An important finding is that the objective and the subjective results of support effectivity did not match. Also, both expectations could not be confirmed, which indicates discrepancies between the theory and the results of this experiment. This can be due to the lack of stressfulness of the team member task and the lack of knowledge of human team leaders to deal with the stress information.

Contents

1 Introduction

Military forces can gain competitive advantages by using advanced military technologies. These technologies can be applied to weapons, vehicles, military robots, and sensors and communications. This last category includes detection of enemies, missile guidance, and coordinating forces. Support on coordinating forces can be facilitated by visualizing the positions of the members of the force on a map by using the global positioning system (GPS). This visualization can be presented to a commander on a mobile device to increase his awareness of the situation and to support his decision making. Providing more information about the situation could possibly improve the decision making even more, considering that providing extra (relevant) information leads to a more informed decision. For example, real-time communication of the number of bullets fired by a soldier can be valuable information for the commander.

In a combat situation, the coordination of forces greatly depends on the position and strength of the enemy and the position and strength of the own forces. Ideally, these four parameters are used as input for the coordination decisions. Currently, only one parameter (the position of the own forces) is used in military technology. The other three parameters can not be measured directly or retrieved at all, which requires an estimation of the value of the parameter. The accuracy of this estimation determines the relevancy to provide this information to the commander.

When making decisions, the commander uses information from support devices (such as visualization of the map and positions), but mostly information from verbal communication with the members of the force. They can provide information about their own condition, probable positions of the enemies and their strength. The accuracy of this information is dependent on the experience of the soldier providing this information, his condition, and the time available to communicate with the commander. In the end, the information provided by the members of the force are subjective and the accuracy and completeness of the information is dependent on the available time.

Although the verbal communication with the members of the force is highly valuable and indispensable, opportunities to enrich and support this information about the environment and condition of the soldiers should be investigated (Ammann et al., 2010). This research presents a possible approach to enrich the information provided to the commander to make a more informed decision and possibly increase the likelihood of mission success.

The most elementary requirement of providing new information in a decision process is that the new information should be relevant to the decision. Coordinating forces in a combat situation is essentially about allocating the right soldiers to the right place. Some places might be under attack heavily, while others require only a sentinel to look out for possible enemy attacks. The selection of specific soldiers for allocation in specific places depends on their skills, condition and the estimated opposition in each place.

There is no accurate measure to estimate the skills and condition of soldiers. Before going in combat, the specific skills of the soldiers are known by the commander, but the condition of the soldier can influence the performance on these skills heavily. An important aspect of the condition of a soldier is the amount of stress he is experiencing. Stress is defined as the nonspecific response of the body to a stimulus or event (stressor), which can have both negative and positive effects on the performance (Kavanagh, 2005). The most used and accepted model of the relationship between stress and performance is the inverted-U relationship, found by Yerkes and Dodson in 1908. It states that performance is optimal when arousal is at moderate levels; when arousal is either too high or too low, performance declines. Soldiers in combat have to deal with several stressors, like fear of death or injury and seeing teammates get injured or killed. In general, their body responds to those stressors by, for example, elevating the heart rate and dilating the pupils (Kavanagh, 2005). The individual and group performance, decision making processes and perception are affected by these stressors (Kamphuis, Gaillard, & Vogelaar, 2011). Figure 1 shows the relationship between stressors, stress and performance.

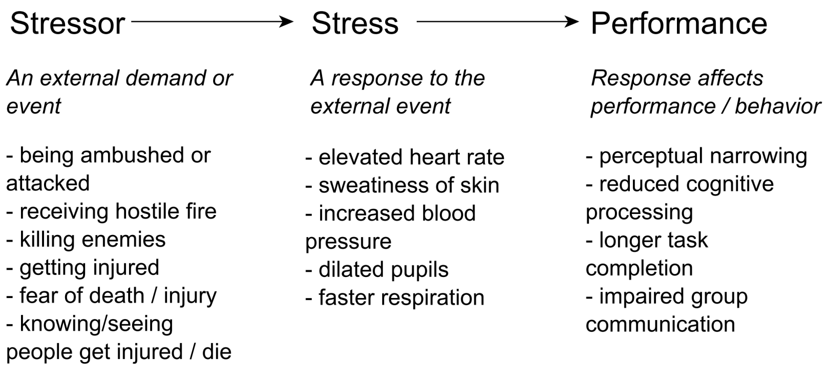

Stressor
Stress
Performance
An external demand or event
A response to the external event
Response affects performance / behavior
- being ambushed or attacked
- receiving hostile fire
- killing enemies
- getting injured
- fear of death / injury
- knowing/seeing people get injured / die
- elevated heart rate
- sweatiness of skin
- increased blood pressure
- dilated pupils
- faster respiration
- perceptual narrowing
- reduced cognitive processing
- longer task completion
- impaired group communication

Figure 1: Relationship between stressors, stress and performance in a combat situation (derived from Kavanagh, 2005)

Because the performance is influenced by the condition (in this case: stress) of the soldier, this would be valuable information to share with the commander. It is possible to extract the stress value of soldiers by asking them or their team mates, but this is subjective and also very intrusive. A less intrusive way would be to measure the heart rate variability (HRV), which is an accurate way of estimating the stress a person is experiencing by analyzing the heartbeat signal (Delaney & Brodie, 2000). Humans can not see whether someone is stressed by interpreting his heart rate variability, but a computer system can. The commander can thus be supported by a computer system that provides him with extra information about the stress of the military forces.

1.1 Research Question

Humans possess unique capabilities in their intuition and experience, where computers score well on consistency, processing capability and processing speed. Potentially, they can achieve more when they cooperate as opposed to functioning on their own. This leads to the following research question:

> *Does decision support, based on an estimation of stress of team members, increase the performance of the team leader in allocating team members to a set of tasks?*

Systems can support human information processing in different stages, like information acquisition, information analysis, decision and action selection, and action implementation (Parasuraman, Sheridan, & Wickens, 2000). For example, the support system can provide the stress information, thereby supporting the information acquisition stage. Another type of support could calculate an optimal task allocation of team members and present this as an advice to the team leader, which support the decision and action selection. The stage in which decision support is provided might influence the effectivity of the support. Therefore, this study implements different types of decision support aimed at different stages of human information processing to see whether the team leader performance is affected by different types of decision support.
The research question consists of three parts: decision support, stress estimation, and an allocation task. To specify what answers will be provided in this thesis, four sub questions are constructed:

1. Does communicating the stress values of the team members help the team leader to make better task allocation decisions?
2. How to design and build a model that takes stress values as input, and produces optimal task allocation as output?
3. How well does the decision support model perform?
4. Does communicating the output by the decision support model help the team leader to make better task allocation decisions?

The coherence of these four questions is shown in figure 2. The numbers in the figure represent the numbers of the related sub question.

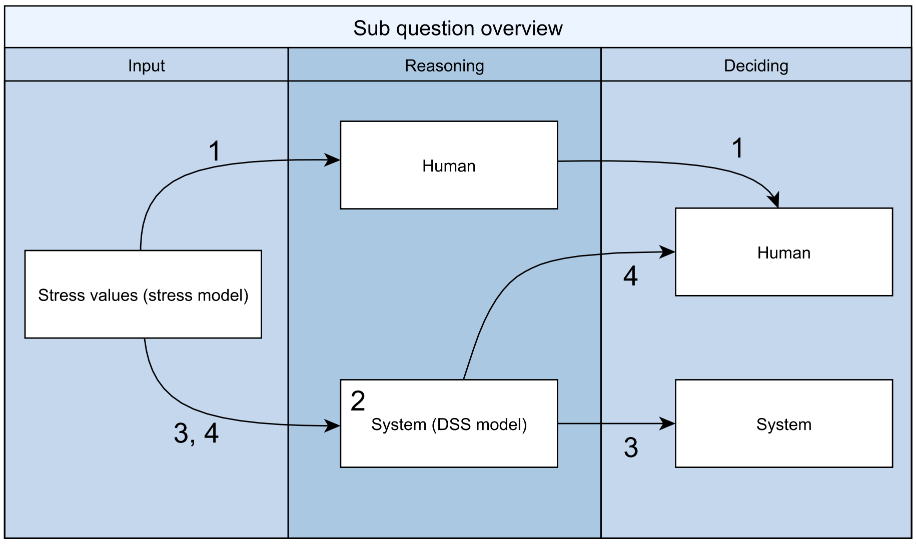


Figure 2: Coherence of sub questions shown graphically

The first sub question concerns the path of showing the stress values to the human, let the human reason about the meaning of the values and let the human decide which actions to take. The second sub question is more of an engineering question: how can a model be built that can approximate the optimal task allocation between subjects by only use stress values of these subjects? The answer to the third sub question should provide clarity about the performance of this constructed model. This can be achieved by letting the system reason about the stress input, and let the system make the decisions. Finally, it will be interesting to see whether a human can use the reasoning skills of the decision support model. The decision support model gives advice to the user by presenting its reasoning outcomes to the human. The human has to make the decisions.

1.2 Thesis overview

Section 2 is the background section, which provides some broad information about decision support systems, task allocation algorithms and human stress characteristics and measurements. This section discusses theories and definitions of the general topics in this thesis. In the next section, an answer is given to the second sub question, which discusses how to design and build a model that takes stress values as input, and optimal task allocation as output. The general idea of the decision support model is explained, the algorithms are presented in a couple of listings and at the end of section 3, an example is provided to see how the model actually comes to a decision advice. To answer the research question and sub questions, an experiment will be carried out. Four hypotheses will be presented in section 4. The fifth section describes the method, which provides all the information concerning the experiment. After that, the results section presents the findings on the independent and dependent variables mentioned in section 5. Also, the hypotheses are accepted or rejected. Finally, the discussion and conclusion (section 7 and 8) will discuss the scientific contribution of this research and the limitations of this study, and recommendations are given for future research. The textual material (task descriptions and questionnaires) used in the experiment can be found in the Appendices.

2 Background

This section will describe human decision making and how support systems can help in this process. The second part of this section concerns the definition of stress and human reactions on stress in terms of bodily reactions and performance, as well as different ways to measure stress.

2.1 Decision making support

The human decision making process generally involves four steps, which is best summarized by John Boyd's OODA loop: Observe, Orient, Decide and Act. The OODA model describes a continuous cycle of situation assessment to eventually reach decision making for action (Macklin & Dudfield, 2001). The OODA loop as it was originally created by John Boyd, is shown in figure 3.

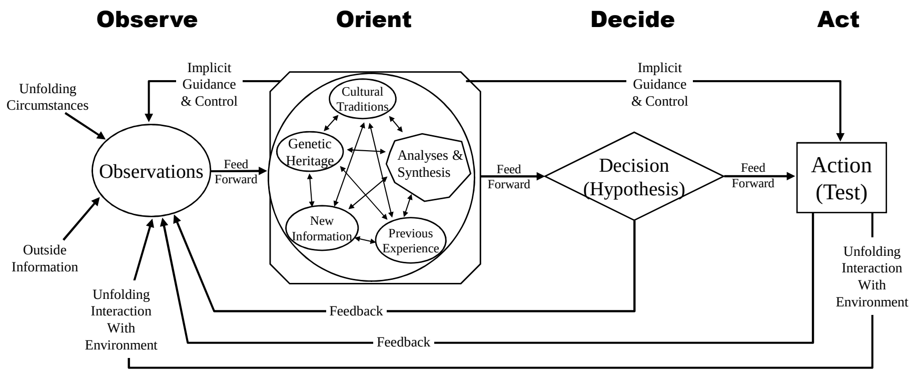


Figure 3: The OODA loop by John Boyd (Boyd, 1987)

In the first phase, the decision maker collects relevant information and identifies problems. If any problems arise, the decision maker uses his intrinsic mental model and his experience to come up with solutions for the problem. When analyzing difficult ambiguous problems, humans base their decisions on pieces of information that together form a mental model. This can be seen as the second phase: the generation of decision alternatives and identification of relationships between factors involving the problem space. This phase can also include verification and examination whether he is aiming for the right solution, if that is possible. The third phase involves making a choice: the decision maker will choose among the options he has created in the second phase. Finally, the decision maker will implement this choice (Rhee & Rao, 1990; Macklin & Dudfield, 2001).
In these four steps of decision making, humans can be supported by a computer system to provide extra or more structured information, and create alternative choices. These systems are called decision support systems (DSS). Central to any definition of decision support systems is the notion of "support" in the definition. A decision support system always supports or helps people to solve problems or make decisions, and is not intended to replace decision-makers. DSSs are usually characterized by three components (Shim et al., 2002; Power

& Sharda, 2009; Sauter, 2010):

1. Database: information and knowledge storage and easy access
2. Model: algorithms and reasoning to filter, merge and interpret information
3. Interface: the part of the system the users see and interact with

DSSs can help in streamlining the decision making process. For example, a DSS can help look at more facets of a decision, generate better alternatives, respond to situations more quickly, solve complex problems, and have new insights into problems by eliminating 'tunnel vision' (Sauter, 2010). The added value of a decision support system depends on the complexity of a task: with a simple task, a decision support system can be redundant, whereas it could be of great help with a more complex task (Power & Sharda, 2009). With a more complex task, the cognitive load and attention needed is much larger than with a simple task. The performance of a user carrying out that task will benefit more from the decision support system with a complex task than with a simple task.
In the military environment, decision support systems can support battlefield visualization, identify critical decisions, assist in determining or validating courses of action, filter information available to the commander, assist in the allocation of resources. In short, three types of command decisions can be supported by decision support systems (Diedrichsen, 2000):

1. Informational decisions: what is the situation?
2. Organizational decisions: how to organize to achieve goals?
3. Operational decisions: what actions should be taken?

Two types of decision support systems can be distinguished: decision support systems that support information acquisition and analysis, and decision support systems that support decision selection and action implementation (Rovira, McGarry, & Parasuraman, 2007). From the command decisions distinguished by Diedrichsen, the informational decisions can be supported by information acquisition and analysis support, which increases the knowledge about the situation of the decision maker. The organizational and operational decisions can be supported by decision selection and action implementation support. These types are discussed in the next two paragraphs.

2.1.1 Information acquisition and analysis

DSSs can help look at more facets of a decision, which is also known as increasing the situational awareness of the decision maker. Situational awareness is defined by Salmon et al. as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Salmon et al., 2007). When a decision maker knows more about the decision problem and its environment, he is likely to make a more informed decision. Situational awareness acquisition and maintenance is influenced by the individual (his experience, training, workload), the task at hand (complexity) and systemic factors (interface design).
More information can also make a decision more complex, because it can lead to information overload. It is very important to present the information in a way that is easy to process for the human brain. Visually presented information should contain cognitive properties such as saliency, priority and be meaningful in context (Ntuen et al., 1998). A good decision support system should

therefore present the information in a meaningful way, where the system should make an estimation of relevance and importance, which influences the way of presenting the information. Visual cues can be rapidly processed by human eyes, which means that if systems translate digitized information to visual cues, the information transfer can be most efficient (Zhu & Chen, 2008). A good visualization makes the data perceivable, which means that the user does not have to construct a mental image himself. In this way, the processing of complex information is reduced to a perceptual process, which requires less cognitive effort. Of all the codes used in modern symbology, color codes seem to have the most attractive qualities because of its characteristic to be knowledge rich, but data sparse (Barnes, 2003). When using color codes, the conventions about meanings of different colors should be taken into account. For example, red often means danger or heat, where green generally means a 'go'. To prevent the user from getting confused, these colors should be used along the lines of their conventional meaning.

2.1.2 Decision selection and action implementation

DSSs can also help in solving complex problems and generate better alternatives. For this type of decision support, the system must have an accurate model to add value to the decision making process of the human decision maker.
The decision making problem that is addressed in this research is a task allocation problem. Consider a number of tasks and a number of team members to carry out these tasks. The task allocation problem is defined by how to match the tasks to the team members to get the best task performance.
Designing a task allocation algorithm that can allocate tasks to different team members is considered an emerging field of research. Particularly, the moment of allocating a process-control task to a machine instead of a human to prevent cognitive overload is extensively researched, for example in the aviation science. Physiological signals that reflect autonomic or brain activity, and presumably changes in workload, could serve as a trigger for changing the task allocation (Prinzel, Freeman, Scerbo, Mikulka, & Pope, 2000). When the human has full attention on one task and can not monitor other tasks at the same time, the machine notices that and can help for the time necessary.
Hardly any research describes a system that can allocate tasks between humans, based on their physiological signals. To achieve this, the physiological signals should reflect an effect on the task performance of the humans, and the task allocation should help optimize the task performance across all tasks. The solution to the task allocation problem can be found in a combinatorial optimization solution: finding an optimal object from a finite set of objects, where each object would represent a task allocation solution. The optimal task allocation solution would be the right balance between task demand and what team members have to offer to carry out the task (capability), for each task in the problem. With multiple tasks, the optimal task allocation solution would be the solution where the summed absolute differences between the task demands and team member capabilities is the lowest.

2.2 Stress

Stress is a very general term and is used in various research fields where it covers slightly different meanings. Driskell and Salas (Driskell & Salas, 1996) provide an extensive definition of stress:

> ”A process by which certain environmental demands (. . .) evoke an appraisal process in which perceived demand exceeds resources and results in undesirable physiological, psychological, behavioral or social outcomes.”

This definition shows some key concepts of stress: the (perceived) expectations or actions from the outside world are perceived by the individual as exceeding his or her capabilities, which results in discomfort for the individual. Where stress used to be defined by the degree of matching between demand and capability of the individual, nowadays the definition has shifted to the perceived demand or expectations and the perceived capability (Driskell & Salas, 1996; Staal, 2004).
Stress can be related to an external event, which is called a stressor. Examples of these stressors are a trauma, certain life events or hassles or daily stressors. All these stressors have a temporal dimension: this involves the duration, rapidity of onset and fluctuation of stress (Aldwin & Werner, 2009).
This research focuses on acute stressors: those stressors that are sudden, novel, unexpected, and of short duration. This includes for example personal threat, time constraint, noise, and task overload (Driskell & Salas, 1996).

2.2.1 Influence of stress on performance

Acute stress can cause bodily reactions by the human being exposed to it (Simon & Zieve, 2009). For example, the production and release of the primary stress hormone cortisol is triggered. Also, certain neurotransmitters like dopamine, norepinephrine and epinephrine are released. These trigger an emotional response to the stressful event, increase alertness and a sense of anxiety. The neurotransmitters also suppress activity concerned with short-term memory, concentration, inhibition and rational thought.
Next to that, the heart rate and blood pressure will increase, lungs take in more oxygen and the transportation of oxygen throughout the body speeds up to provide the muscles, lungs and brain extra supply. The blood flows away from the skin to support the heart and muscle tissues, causing a sweaty skin.
Partly as a cause of these physical reactions, the cognitive performance can change in an acute stress situation. Hancock and Warm (1989) developed a graph that shows the course of human adaptability as a function of stress (see figure 4).

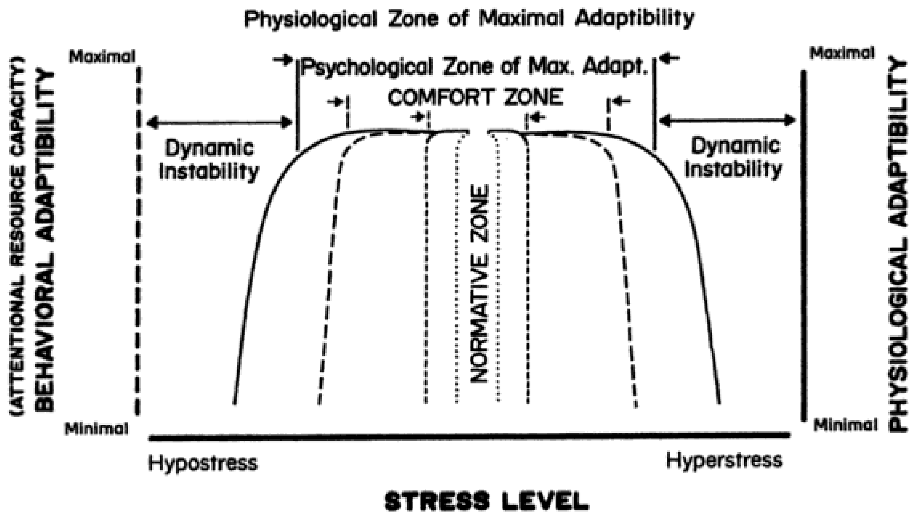


Figure 4: Performance as a function of stress (Hancock & Warm, 1989)

This theory states that moderate stress would lead to the most optimal performance, whereas hypostress (boredom) or hyperstress (severe stress) is associated with low performance.
Individuals under stress are sometimes less effective at scanning alternative solutions or checking the consequences before making a decision (Gohm, Baumann, & Sniezek, 2001). Research on severe and acute stress show that cognitive functions such as reaction time, vigilance, memory and logical reasoning are significantly impaired (Lieberman et al., 2005). The key of decreased cognitive performance lies in the amount of attention that is drawn to the stressfulness of the event. The more attention is captured by worrying and thinking about the stressful event, the less attention can be given to the task at hand. Individuals who are clear about their emotional reaction to an acute stress situation spend less attention struggling with their emotional responses, allowing more attention to be given to the current task (Gohm et al., 2001).
All the cognitive reactions on stress disable a person to concentrate fully on their task, which leads to a decrease in task performance. To what extent stress degrades performance is not generally known; it depends on the ability of the individual to cope with the stressful event (Delahaij & Gaillard, 2008). People differ in their actual coping behavior during a stressful situation, also referred to as situational coping. A predictor of behavior during a stressful situation is coping efficacy, which refers to the beliefs people have about their capability to cope with a specific stressful situation.
Another predictor for situational coping is personality. Personality is defined by Allport as "the dynamic organization within the person of the psychological and physical systems that underlie that person's pattern of actions, thoughts and feelings" (Allport, 1961).
Personality differs among individuals and can be measured or assessed individually by using the five-factor model of Costa and McCrae (Costa & McCrae, 1992). These five factors are extraversion, neuroticism, agreeableness, conscien-

tiousness, and openess to experience. Carver and Connor-Smith have collected studies on the relationship between personality and stress (Carver & Connor-Smith, 2010). They found that especially when a person scores high on neuroticism (sensitive to moodiness, anxiety, and threat), he is likely to show emotional or physical reactions to a stressful event (stressor reactivity). This finding is in accordance with findings from previous researches (Schneider, 2004).

2.2.2 Measuring stress

It is difficult to measure stress directly, but it can be approached by psychological and physiological measures. As a psychological measure, the subjective stress of a person can be measured by asking him to estimate (self-report) his stress. Because stress causes many bodily reactions, measuring the physical affects in the body can also make stress quantifiable. These measures are called physiological measures, and include heart rate, heart rate variability, salivary cortisol, galvanic skin response, eye related activity, brain activity, body temperature, and respiration measures. Many studies have been carried out to test these physiological measures against known stressors, with varying results (Kreibig, 2010).
A measure that is considered closely related to stress is the heart rate variability. Stressors are associated with an increase in sympathetic cardiac control, and a decrease in parasympathetic control. This causes an increase in low frequency heart rate variability, and a decrease in high frequency power (Berntson & Cacioppo, 2004). A low heart rate variability indicates a high stressor reactivity. The applicability of the heart rate variability measure for measuring stress is confirmed by various researches both in acute laboratory psychological/cognitive stressors as with real-life acute stressors and chronic perceived stress (Delaney & Brodie, 2000; Dishman et al., 2000; Friedman, Thayer, & Tyrrell, 1996; Hjortskov et al., 2004; Lucini, Di Fede, Parati, & Pagani, 2005).

3 Decision support model

This chapter will describe a model that can aid team leaders in allocating tasks to different team members, based on their stress values. It is concerned with the allocation task, given a certain stress input. The decision support model should be able to match team members to specific tasks and provide the best matching. The problem is best described using figure 5.

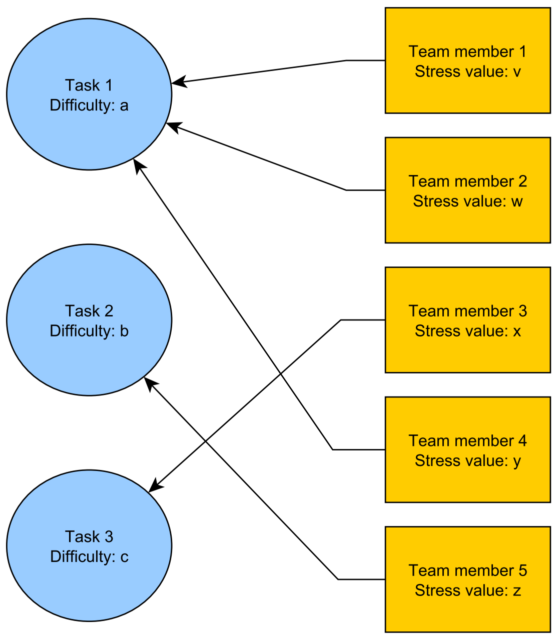


Figure 5: Schematic representation of the matching problem

In this example, three tasks and five team members need to be matched. Each task has a difficulty and each participant has a stress value that might say something about his ability to carry out tasks. To match the right team member to the right task, the difficulty of the tasks should be estimated. This can be deduced by using the stress value of a teammember or group of teammembers; when the average stress value of team members working on a task is high, the task is expected to be difficult. Each task can continually change in difficulty, both depending on the number and ability of the teammembers working on it and random factors. Using all teammembers to their fullest will result in the best performance overall. When stress values are high, performance on the task will decrease. When stress values are low, the teammember can maybe help another teammember. The right balance between over- and underload of activity will yield the best results.
The model uses a few basic assumptions to calculate the best task allocation:

- The task demand (relative difficulty of one task compared to others) should match the sum of capability values (inversely proportional to stress values) of the soldiers that are working on that task as closely as possible.

- When another task allocation will result in a better match, task re-allocation is assumed appropriate.
- A threshold will be included to protect the system from constantly improving the task allocation with only minor improvements. Changing the task allocation has some drawbacks (it takes time for the team members to adapt to their new task), and a threshold value filters those task allocations that are slightly better, but not outweigh the disadvantages of frequent changes.

The model that is developed to solve the problem and meet these requirements, is presented in figure 6.

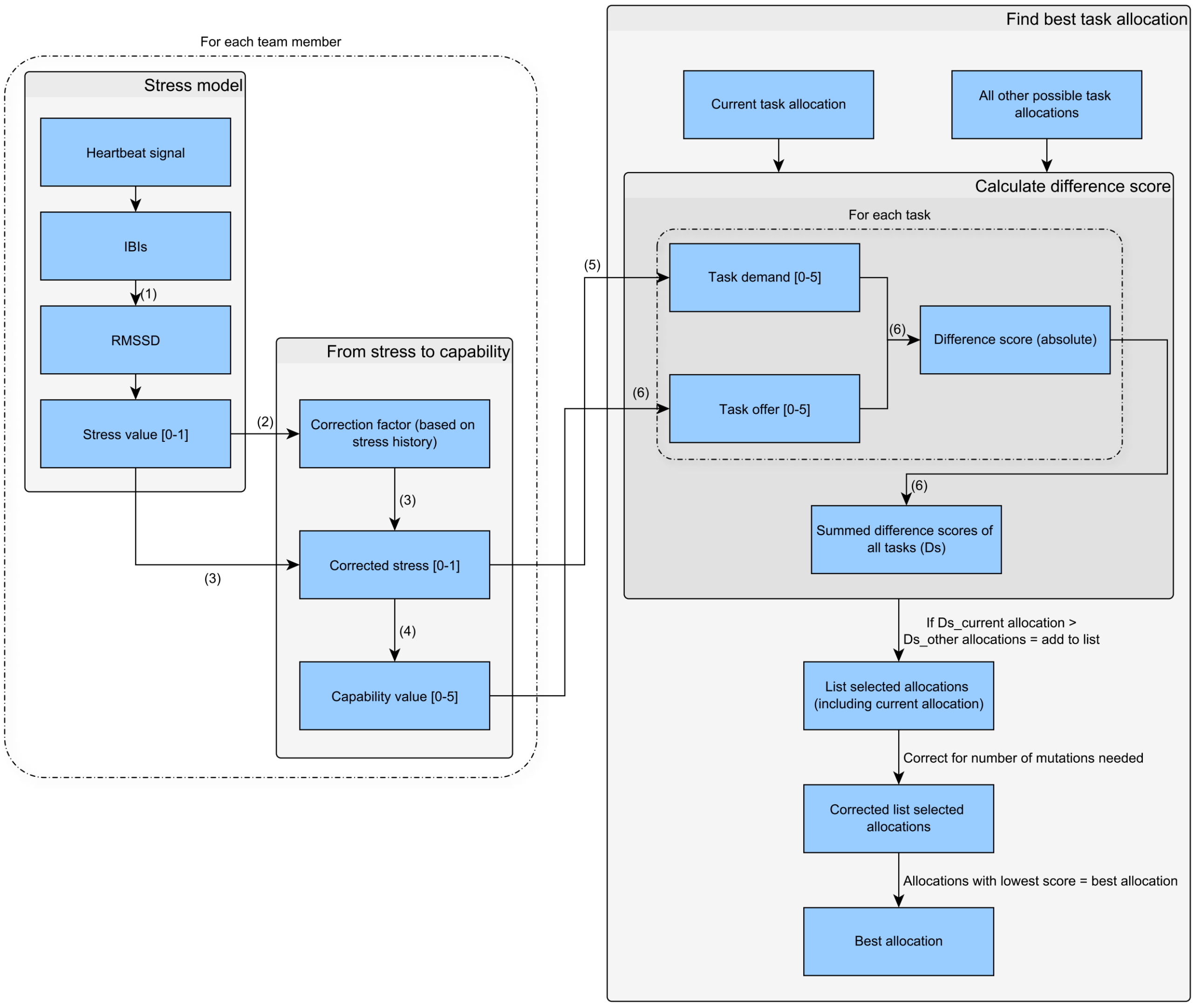

For each team member
Stress model
Heartbeat signal
IBIs
(1)
RMSSD
Stress value [0-1]
(2)
(3)
From stress to capability
Correction factor (based on stress history)
(3)
Corrected stress [0-1]
(4)
Capability value [0-5]
(5)
(6)
Find best task allocation
Current task allocation
All other possible task allocations
Calculate difference score
For each task
Task demand [0-5]
Task offer [0-5]
(6)
Difference score (absolute)
(6)
Summed difference scores of all tasks (Ds)
If Ds_current allocation > Ds_other allocations = add to list
List selected allocations (including current allocation)
Correct for number of mutations needed
Corrected list selected allocations
Allocations with lowest score = best allocation
Best allocation

Figure 6: Summary of the decision support model

The titles of the greyed boxes correspond to the subsections, and the numbers between brackets correspond to the formulas mentioned in these subsections.

3.1 Stress model

In this experiment, stress is measured by using heart rate variability (HRV). The stress model thus needs to transform a heart rate signal to a stress value. In figure 7, two heartbeats are shown.

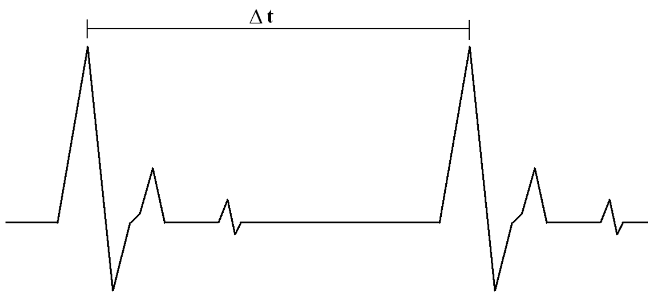


Figure 7: Schematic representation of two heartbeats

The time between the heartbeat peaks (Δt) is known as the interbeat interval (IBI). Heart rate variability must be measured over a period of time, so a list of IBIs is created. Every ten seconds, all items on the list of IBIs (n) are used to calculate the root mean square of successive differences between these IBIs (RMSSD). The RMSDD is calculated by using the following formula:

$$RMSSD = \sqrt{\frac{\sum_{t=1}^{n}(IBI_t - IBI_{t+1})^2}{n}} \tag{1}$$

This RMSSD value is a measure of the variability within the interbeat intervals. As described in the background chapter, a low variability corresponds to an aroused user state, where a high variability corresponds to a relaxed user state. Therefore, low RMSSD values should be transformed in a high stress value, and the other way around.
This stress value is used as input for the decision support model, which requires a stress value between certain values. Therefore, the stress calculation involves another step to normalize the RMSSD value to a value between 0 and 1. This requires a pre-known minimum and maximum RMSSD value, to calculate the values in between. Because the RMSSD value is personally dependent, the most convenient way to normalize it is to calculate the minimum and maximum RMSSD value for each participant before the real measurements start. In this so-called calibration, the participant is confronted with a low-stress situation and a high-stress situation. From this calibration, the lowest and highest RMSSD value is extracted and used as minimum and maximum to calculate the stress values in the rest of the experiment. New incoming RMSSD values will be compared to the minimum and maximum values: lower or equal to the minimum RMSSD will result in stress value 1 (which means highly stressed), higher or equal to the maximum RMSSD results in stress value 0 (which means very relaxed), and anything in between will result in a value between 0 and 1.
Because the stress values of the rest of the session depends on the calibration, it

is very important to make sure that the participants are confronted with both high and low stress situations to find the minimum and maximum values of the RMSSD correctly.

3.2 From stress to capability

Before the stress value is transformed to the capability value, a correction factor is needed. This correction factor corrects for unrealistically big changes in capability values and corrects for difficulty differences between team members. This correction factor is calculated by averaging the Z-score of each team member compared to other team members over a period of time. Each team member thus has a different correction factor. To use this Z-score as a factor, it is normalized to a value between 0 and 1 (instead of approximately between -2 and 2). Formula 2 shows this calculation, where x_i is the stress of the team member, μ_i the average stress of all team members and σ_i is the standard deviation of the stress values of all team members at time i. For each point in time available (maximum is t), the Z-score is calculated and averaged by dividing it by the number of time points t and the result is normalized.

$$Correction\ factor = \frac{1}{4} * \left(\frac{\sum_{i=1}^{t} \frac{x_i - \mu_i}{\sigma_i}}{t} \right) + 0.5 \tag{2}$$

With this correction factor, the corrected stress can be calculated. The corrected stress on time t is calculated by multiplying the stress value with the correction factor and normalizing it. For team member number s, the corrected stress can be calculated using the following formula, where n means the number of team members.

$$Corrected\ stress_s = \frac{stress_s * correction\ factor_s}{\sum_{i=0}^{n}(stress_{s+i} * correction\ factor_{s+i})} \tag{3}$$

The final step is calculating the capability value.The capability value is the relative capability of a team member to carry out a task, expressed in units of team members. A capability of 1 means that this team member is capable of carrying out a task for the value of 1 average team member. When the value is above 1, this team member is capable of more than 1 average team member can do. The same is the other way around: a value below 1 indicates that this team member can do less than one average team member. The capability value is calculated by substracting the corrected stress from 1, and normalizing this by dividing it by the sum of all inversed corrected stress scores and multiplying it by the number of team members.

$$Capability\ value_s = \frac{1 - corrected\ stress_s}{\sum_{i=0}^{n}(1 - corrected\ stress_{s+i})} * n \tag{4}$$

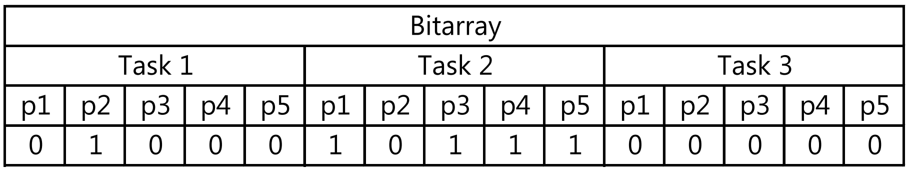
3.3 Find best task allocation

The task allocation algorithm is based on calculating the capability of a team member from his stressvalue and matching it with the capabilities needed to carry out a specific task. First, the notation of different task allocations is explained. After that, the algorithm of finding the best task allocation is presented.

The general approach in solving the task allocation is found in representing each possible solution in a bitarray with a fixed number of positions. A bitarray is a list of positions, where each position is filled with either a 0 or 1. The number of positions depends on the number of tasks and the number of team members. For example, if there are three tasks and five team members, the number of positions would be $3 * 5 = 15$. For each team member, there is a value 0 or 1 if he is working on a specific task. This is represented by taking the first five bits as a value whether each person is working on task 1, the next five bits as a value whether each person is working on task 2, and the last five bits as a value whether each person is working on task 3. For example, the following bitarray

01000 10111 00000

can be seen as figure 8.



Bitarray														
Task 1					Task 2					Task 3				
p1	p2	p3	p4	p5	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
0	1	0	0	0	1	0	1	1	1	0	0	0	0	0

Figure 8: Bitarray explained per bit

It means that person 2 (p2) is working on task 1, the rest (p1, p3, p4, p5) is working on task 2, and nobody is working on task 3. Because bitarrays use limited space, complex task allocations can be represented in a compact and efficient manner. Although it is not possible to change the number of tasks or team members during the decision support process, it is possible to start the model again and use different numbers.
This algorithm handles the relationship between tasks and team members as a one-to-many relationship as depicted in figure 9.

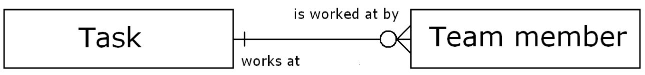


Figure 9: Diagrammatic representation of the relationship between task and team member

One task can be worked on by zero or more team members, but a team member can only work on precisely one task at a time. This means that a team member must always be assigned to one of the tasks.
When the model is run, it will generate all possible bitarrays and put them in a list. Because the number of tasks and team members during the task will not change, the model can generate these once. In the case of 3 tasks and 5 team members, the model generates 3^5 bitarrays and stores them in a list. An excerpt of the generation algorithm is shown in listing 1, where s is the number of team members and t is the number of tasks.

Listing 1: Generation algorithm

```
private static List<BitArray> legal = new List<BitArray>();
private static BitArray working = new BitArray(s * t);

private static void GenerateLegalDistributions(int teammember)
{
      for (int task = 0; task < t; task++)
      {
            working.Set(teammember + task * s, true);
            if (teammember + 1 < s)
            {
                  GenerateLegalDistributions(teammember + 1);
            }
            else
            {
                  legal.Add((BitArray)working.Clone());
            }
            working.Set(teammember + task * s, false);
      }
}
```

The algorithm needs to compare the current bitarray with alternative bitarrays to find the best bitarray and thus the best solution.

3.3.1 Calculate difference score

Now the capability values have been calculated, the task offer values can be deduced. The corrected stress values are used to get an indication of the difficulty of the task, which is called task demand. When the team members working on the task are stressed, the task is considered to be difficult. The corrected stress values are averaged per task, which gives an average stress score for each task. This is normalized to a task demand value, which is shown in the next formula where d means the current task number, n means the number of soldiers working on that task, and x means the number of tasks.

$$Task\ demand_d = \frac{\sum_{i=0}^{n} corrected\ stress_i}{n} * x \tag{5}$$

For each task allocation possibility, the task demand and team member capability are calculated and compared to the current task allocation. The comparison is based on scores of differences between task demand of each task and the capability allocated to each task, where td is task demand and Ds is difficulty score:

$$Ds = |td_0 - (\sum capability\ s_{td_0})| + \cdots + |td_n - (\sum capability\ s_{td_n})| \tag{6}$$

The calculation of differences between the task demand and the team member capabilities is shown in listing 2. The capability values of the team members are stored in the `listCapabilities`, which is a list with on position 0 the capability value of person 0, on position 1 the capability value of person 1, ...

until position n. The task demand values are stored in the `listTaskDemand`, which is a list with on position 0 the task demand of task 1, and so on.

Listing 2: Calculate bitarray difference

```
private double CalculateArrayDifference(BitArray bits)
{
      //Sum of stress values across tasks
      double subsetsum = 0;
      //Score of the difference of the current BitArray
      double difference = 0;
      for (int i = 0; i < t; i++)
      {
            for (int j = 0; j < s; j++)
            {
                  if (bits.Get(s * i + j))
                  {
                        //Sums the last added values
                        subsetsum += listCapabilities[j];
                  }
            }
            difference += Math.Abs(subsetsum -
                listTaskDemand[i]);
            subsetsum = 0;
      }
      return difference;
}
```

This method can be used on any bitarray. Each bitarray (task allocation) now has a difference score, which is used in the rest of the calculation to find the best task allocation.
Because of its size, the algorithm to find the best bitarray (the bitarray with the smallest `difference` value) is displayed in appendix A. The model calculates the bitarray differences of every possible bitarray of list `legal` from listing 1 and calculates the bitarray difference of the current task allocation. It then compares every possible bitarray difference score with the current task allocation score; whenever the alternative bitarray has a lower (which means better) score, it is added to a list (`listBetterArrays`). If all bitarray alternatives have passed this test and no items are present in the `listBetterArrays`, the `currentdistribution` is the best bitarray and is returned. However, if the `listBetterArrays` is not empty, the difference scores of the residual bitarrays (including the current task allocation) are compared with each other. Also, a punishment is added for each mutation needed because more mutations requires more adaptation from the team members and definitively some time loss. The bitarray with the lowest score is returned as the best bitarray.

3.4 Example

To clarify the process of decision making by the model, this paragraph provides a numerical example.
The input of the decision support model is stress values of the team members,

so fictive stress values of five team members are collected on five points in time.

Table 1: Stress values

	t_1	t_2	t_3	t_4	t_5
p_1	0.9	0.9	0.5	0.3	0.1
p_2	0.5	0.5	0.5	0.5	0.5
p_3	0.2	0.1	0.1	0	0.3
p_4	0.3	0.2	0.6	0.9	1
p_5	1	0.7	0.8	0.9	1
Σ	2.9	2.4	2.5	2.6	2.9
μ	0.58	0.48	0.5	0.52	0.58
σ	0.36	0.33	0.25	0.39	0.41

By using the mean and standard deviation, a Z-score can be calculated for each team member for each point in time. These are added and normalized to a value between 0 and 1, resulting in a correction factor as represented in equation (2).

Table 2: Correction factors

	t_1	t_2	t_3	t_4	t_5
p_1	0.72	0.77	0.68	0.60	0.52
p_2	0.44	0.22	0.49	0.49	0.48
p_3	0.23	0.22	0.19	0.18	0.21
p_4	0.30	0.30	0.40	0.48	0.54
p_5	0.79	0.73	0.75	0.75	0.75

Now, the corrected stress value for each team member can be calculated by multiplying the stress value from table 1 with the correction values from table 2. As presented in equation (2), this value is normalized to a value between 0 and 1.

Table 3: Corrected stress values

	t_1	t_2	t_3	t_4	t_5
p_1	0.36	0.45	0.24	0.12	0.03
p_2	0.12	0.16	0.17	0.16	0.15
p_3	0.03	0.01	0.01	0	0.04
p_4	0.05	0.04	0.17	0.28	0.33
p_5	0.44	0.33	0.42	0.44	0.46

Finally, the capability values can be calculated by substracting the corrected stress value from 1 and normalize it to a value between 0 and 5 (see equation (4)).

Table 4: Capability values

	t_1	t_2	t_3	t_4	t_5
p_1	0.80	0.68	0.96	1.10	1.21
p_2	1.10	1.05	1.04	1.05	1.07
p_3	1.22	1.23	1.23	1.25	1.20
p_4	1.19	1.20	1.04	0.89	0.84
p_5	0.70	0.83	0.73	0.70	0.68

Consider the task allocation at each moment in time as follows:

t_1: 10100 01010 00001

t_2: 10000 01110 00001

t_3: 10000 01100 00011

t_4: 01000 10100 00011

t_5: 10001 01000 00110

The model will calculate the task demand with these bitarrays, but to make it more clear, the tasks per team member for each point in time is shown in table 5.

Table 5: Task allocation

	t_1	t_2	t_3	t_4	t_5
p_1	1	1	1	2	1
p_2	2	2	2	1	2
p_3	1	2	2	2	3
p_4	2	2	3	3	3
p_5	3	3	3	3	1

Then, the average of the corrected stress values for each task are calculated and this value is normalized to a value between 0 and 5 to match the capability values. The equation linked to the following table is equation (5).

Table 6: Task demand

	t_1	t_2	t_3	t_4	t_5
Task 1	1.34	1.88	1.95	1.84	1.94
Task 2	0.60	0.52	0.59	0.57	0.65
Task 3	3.06	2.60	2.46	2.59	2.41

Adding up the capability values from table 4 for each task, the following table is generated.

Table 7: Task offer

	t_1	t_2	t_3	t_4	t_5
Task 1	2.01	0.68	0.96	1.05	1.89
Task 2	2.28	3.49	2.27	2.35	1.07
Task 3	0.70	0.83	1.77	1.59	2.04

The absolute difference between each cell of table 6 and table 7 is the difference score and an indicator of how well the tasks were distributed at that point in time. In this example, only five task allocation possibilities (in this context: five bitarrays at five points in time) were calculated, but the model will calculate a difference score for every possible bitarray for each point in time to find the optimal solution. Table 8 shows the difference scores on these five bitarrays, calculated with equation (6).

Table 8: Difference scores

	t_1	t_2	t_3	t_4	t_5
Task 1	0.67	1.20	0.97	0.79	0.05
Task 2	1.68	2.97	1.69	1.79	0.42
Task 3	2.35	1.77	0.69	0.99	0.37
Σ	4.70	5.94	3.35	3.57	0.84

The lower row of the table shows the difference scores per bitarray. With this value, the bitarrays are compared to each other and the best bitarray is chosen. To demonstrate this selection process, the last point in time (t_5) is taken as an example. For this point in time, the algorithm selected 8 bitarrays which have lower scores than 0.84.

Table 9: Selected bitarrays which had smaller difference scores than the current task allocation bitarray

Bitarray	Diff score	# Mutations
11000 00001 00110	0.74	2
10010 00001 01100	0.28	3
10001 00010 01100	0.38	2
01100 00001 10010	0.72	4
01010 00001 10100	0.06	4
01001 00010 10100	0.38	3
00110 00001 11000	0.26	5
00101 00010 11000	0.38	4

When we would not look at the number of mutations necessary, the bitarray with a difference score of 0.06 would be the optimal solution. However, a punishment for each mutation would change this decision. Which bitarray would be the best choice depends on the weight of a mutation. The relationship between the total score (difference score including punishment based on the number of mutations) and the value of the punishment can be seen in figure 10.

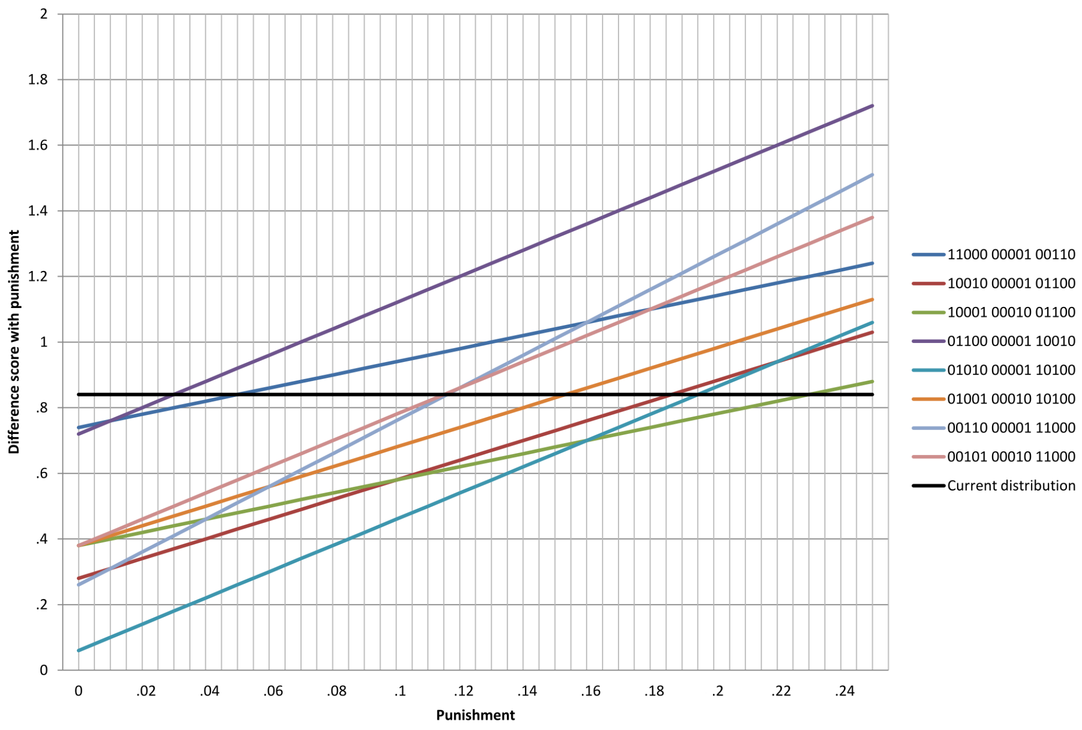


Figure 10: Relationship between value of punishment and total difference score

The bitarrays in the legend are in the same sequence as in table 9. The best bitarray on each point in time is the lowest value. The more mutations necessary, the more steep the line of the bitarray will increase. With a *punishment* $<$ 0.16, the 01010 00001 10100 bitarray would be chosen (the fifth value in the legend), because of its low starting value and the low punishment values which do not influence the total score that much. But with a punishment of $0.16 < punishment < 0.23$, the choice would be the third bitarray (10001 00010 01100). Any punishment above 0.23 would result in the current task allocation to be considered the best (black horizontal line). The weight of the punishment is thus very important for the outcome of the task allocation advice.

4 Hypotheses

As explained in the introduction, this research is an exploratory research with an experiment to explore the influence of decision support on team leader performance. The research question is:

> *Does decision support, based on an estimation of stress of team members, increase the performance of the team leader in allocating team members to a set of tasks?*

The sub questions, specified in the introduction, are:

1. Does communicating the stress values of the team members help the team leader to make better task allocation decisions?
2. How to design and build a model that takes stress values as input, and produces optimal task allocation as output?
3. How well does the decision support model perform?
4. Does communicating the output by the decision support model help the team leader to make better task allocation decisions?

The first hypothesis is not related to a specific subquestion, but is generally important. It concerns the assumption that stress is a negative predictor for performance, which is the input for the decision support model. Because the reasoning of the decision support model is dependent on this assumption, this assumption needs to be checked first.

Hypothesis 1. *The stress value of the team members is a negative predictor for their performance.*

The next step is to find out whether a team leader makes better decisions if he is provided with stress information about the team members. When the team leader is provided with extra (useful) information, the situational awareness of this team leader increases. As discussed in the previous section, better situational awareness leads to better decisions and guidance, which eventually will result in a greater likelihood of decision success.

Hypothesis 2. *Communicating stress values of team members will increase the team leader performance compared to the situation where no decision support information is provided.*

Although the sub questions can be a guideline for generating hypotheses, the second question is an engineering question which has already been answered by describing the constructing of the model in the previous section, but there will be no specific hypothesis about this sub question.
The third sub question concerns the performance of the constructed decision support model. When stress is a predictor for performance (see hypothesis 1), the model is expected to produce useful output. This can be tested by letting the model execute the allocation task without the human team leader interfering.

Hypothesis 3. *Execution of the team leader task by the decision support model without human team leader interference will outperform the human team leader without decision support information.*

The last hypothesis is linked to the fourth sub question. Because human and computers have different capabilities and specialties, combining those two in decision making could result in a better decision than only use one. It is hypothesized that complementing these capabilities, which lead to the computer carrying out algorithm-based decision making, and the human contributes experience, empathy and non-quantifiable factors as input, the best decision can be made.

Hypothesis 4. *Combining human and computer reasoning will lead to the best team leader performance.*

In the conclusion section of this thesis, the results of the experiment will be used to either accept or reject each hypothesis that is mentioned in this section.

5 Method

To test the hypotheses, a military environment is needed which has both stressful individual tasks and a team leader to coordinate these tasks and assign them to individuals. A game setting is chosen, where individuals (soldiers) perform a first-person shooting task and a team leader functions as a commander to send individuals to the right place. To make the soldier task stressful, the soldiers wear a gaming vest which gives feedback through pneumatic actuators based on events in the game (for example when they get hit by a bullet). Also, the soldiers are instructed to perform well on specific performance measures: get hit as little as possible, shoot as many enemies as they can, and shoot as accurately as they can. On top of that, they are fighting enemies that can shoot very accurately, which increases the difficulty of the task to shoot them. During the game, the stress of the soldiers is measured by a stress monitor. Based on these stress values, a decision support model can assist the commander by giving advice about this task allocation.

5.1 Task description

The experiment is based upon a real scenario in which soldiers are situated in an unfamiliar environment where they have to explore the environment and preserve peace. In this scenario, it is possible that a more violent situation arises, where a firefight is inevitable. This is where the experiment starts: the soldiers have to prevent enemies from entering the village by killing them. The commander can guide soldiers and can switch soldiers between different fronts (tasks) to minimize the amount of enemies entering the village.

5.1.1 Roles

Within the task, three roles can be distinguished.

1. Soldier role
   The task of the soldier is protecting the village by killing the enemy and get hit as little as possible. When they are ordered by the commander to move to another location, they automatically obey. Hereby, the allocation performance of the commander can be measured without also measuring his authority skills.
2. Commander role
   The task of the commander is to assign the soldiers to different locations, according to his estimation of the situation in the field. He is also responsible for the team performance, meaning that a bad team performance will result in a low score on the commander's task performance.
3. Enemy role
   The enemy in this experiment consists of computer-controlled infantry. They are allocated to different locations through spawn points and they will try to kill the soldiers. When they get the opportunity (i.e. when the soldiers offer little resistance), they will attack the village by using long-range weapons: in the game, a bomb will explode on different locations in the village. This will be programmed by using a timer that runs down when enemies are in a region. The more enemies in each region, the faster

the timer goes to zero. The timer will be reset right after an explosion has occurred. This explosion is implemented to induce more stress to the soldiers, both physically (with each explosion, all eight actuators in the vest are activated) and mentally (the noise and clouds of dust).

5.1.2 Environment

Virtual Battlespace 2 (VBS2) is used as the first-person shooter game in this experiment. VBS2 is a simulation environment which is well-known for training and experiment applications when they relate to military themes. A virtual, realistic world is generated where the players can walk around and can choose a wide variety of actions. VBS2 is used particularly in the areas of tactical training and mission rehearsal. The game is provided with a mission editor, which allows creating specific scenarios and environments. More information on the VBS2 game is provided in appendix B.
For this experiment, two scenarios are created. One scenario is used as a test environment to let the participants get used to the controls of the game and to calibrate the stress monitors at the same time. The other scenario is used to gather the actual data. This second scenario is a multiplayer scenario, which means that the soldiers are in the same game and can see each other in the game. In a desert-like environment, a village is situated in the middle of the map. The village consists of a couple of houses and fences, with a road around the village. Soldiers can find cover behind houses, fences, containers, and some scrub or small trees. The village is a good outlook post for spotting approaching enemies, because the direct surroundings offer little possibilities to find cover. Below in figure 11, a screenshot of the village and its surroundings is shown.

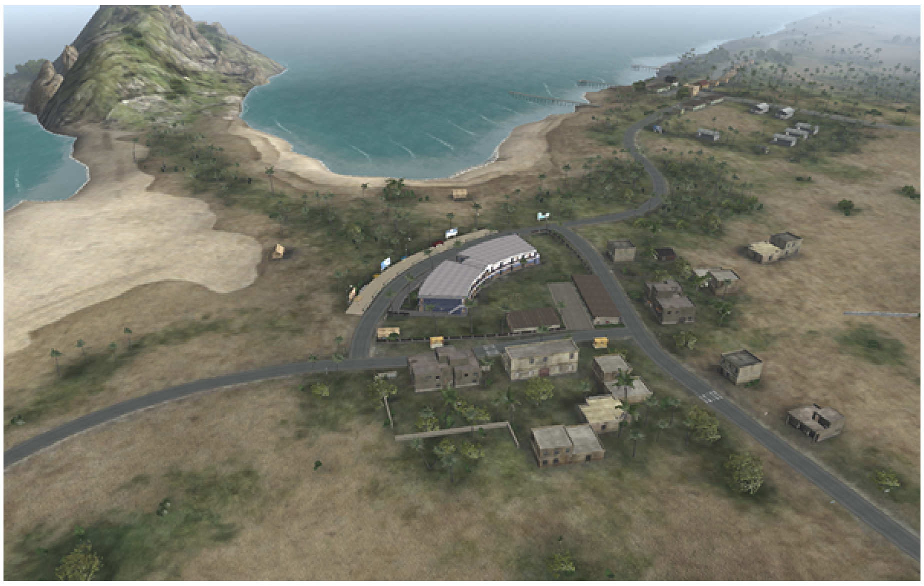


Figure 11: The village and its surroundings in the second scenario in VBS2

The VBS2 scenario spawns 50 enemies on one of three locations. The choice in which location an enemy is spawned, is regulated by a program called the experiment controller (see section 5.7). The soldiers are inside the village, trying to prevent enemies from coming in. There are three fronts (corresponding to the spawn point numbers), to which the commander can assign soldiers. These three regions are considered three different tasks. The location of the spawn points and regions are shown in figure 12.

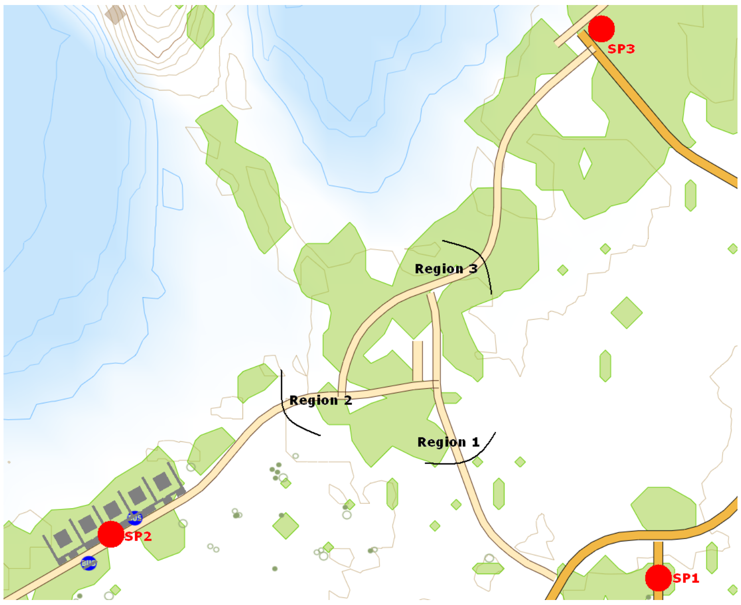
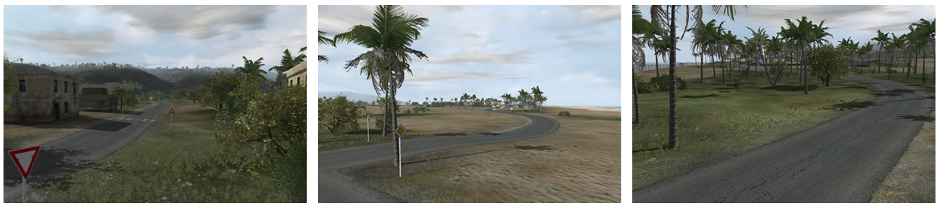


Figure 12: A schematic representation of the map with the spawn points (SP#) and the region names

When an enemy dies, he is spawned on one of the three locations again. This way, the number of enemies in the game always stays the same, independently of the performance of the soldiers. The soldiers can get hit by the enemies, which impairs their ability to shoot accurately (their view is affected), but they can not die. Below in figure 13, the outlook view for each region is depicted. The viewpoint is the soldier looking from the village to the spawn points, where the enemies come from.



(a) Region 1 (b) Region 2 (c) Region 3

Figure 13: Outlook views in all regions

5.2 Decision support

As mentioned in the background section, the three components of a decision support system are:

1. Database: information and knowledge storage and easy access
2. Model: algorithms and reasoning to filter, merge and interpret information
3. Interface: the part of the system the users see and interact with

The decision support system used in this experiment will be described using these three components.

5.2.1 Database

The database of the decision support system consists of input from the stress model and from the VBS2 game. Stress of the team members is monitored by a stress monitor, which calculates the stress value of a team member every 10 seconds. As soon as all stress monitors have passed a value to the decision support model, the support model can start the calculation.
The VBS2 game gives input about the position of the soldiers on the map. This information is updated every second, which always provides the commander with up to date information about the location of all soldiers.

5.2.2 Model

The model is extensively explained in the Decision support model section. For this experiment, the model is built using the programming language C#. The number of soldiers in this experiment is five and the number of tasks is three, so these values are used in the model. An important number to determine is the punishment value when selecting the best array. Because it was not known beforehand what effect the switching of soldiers would have on the game and how bad the disadvantages would be to switch many times, it was difficult to assign a value to the punishment. The maximum difference score of the current distribution is 10; this distribution would be horribly wrong in terms of matching the soldier capabilities to the task demands. When the algorithm of finding the best fitted array would return an array with difference score 0, then the current distribution with difference score 10 should be changed radically. In all other cases, making five changes at a time should be avoided. With this in mind, the punishment value is set on a value of 2. When the difference score of a better array is 0.1, and the current distribution scores 10, the better array would not be chosen if it requires 5 mutations ($0.1 + 2 * 5 = 10.1$). Instead, a temporal solution is chosen which improves the situation but does not change the situation too radically.
A problem arises when a task is not carried out by any soldier. The task demand is now impossible to calculate, because the average stress value at the task is zero and the system will never assign a soldier to this task again. A simple solution to solve this problem is to take the last known average stress value of that region when soldiers were still in that region, and with every cycle of generating decision advice (10 seconds), add 0.01 to this task demand. Eventually, the task demand will be high enough to allocate a soldier to that region and check the situation.

5.2.3 Interface

To carry out the commander task, the commander has to interact with a program interface and he can talk to the soldiers. The program interface has different looks and the model behind the program has different possibilities for supporting the commander, representing different experimental conditions. These conditions are based on steps in the decision making process, and specifically on the OODA loop of military strategic decision making (see figure 3). In this experiment, it is applied to a computer program that can give different types of support based on the phases of decision making. Four conditions are designed, where the program supports the decision maker in increasing amount of steps.

1. The first step is 'Observe', which will correspond to the Basis condition in this experiment. In the OODA loop, this step means observing the situation, but not taking any decisions or processing information about the decision. The system will behave like observing; it will do nothing to help the team leader in decisions making.
   The interface will provide the commander with a graphical representation of the field and positions of the soldiers plotted on this map. No additional support is available in this condition, which makes this condition the control condition.

2. The 'Orient' step corresponds to the Visualization condition. Task-specific information, in this case stress information, is provided to the user to make a more informed decision. The system will support the team leader with the orient step of decision making.
   The interface has the same representation of the field and the soldiers as in the baseline condition. Additionally, the commander will be informed of the stress values of all soldiers. Because the commander has little time to read much and must see at a glance how the situation at hand is developing, the stress values are represented by colored dots. A low stress value is represented by a green dot, a high stress value is represented by a red dot; all values in between are colored between red and green according to the stress value; see figure 14.

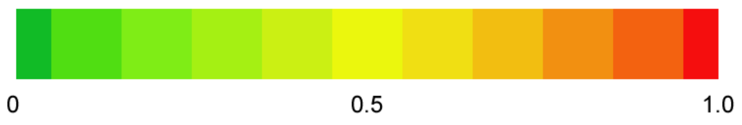
   

   Figure 14: Legend for interpreting the colored soldier dots

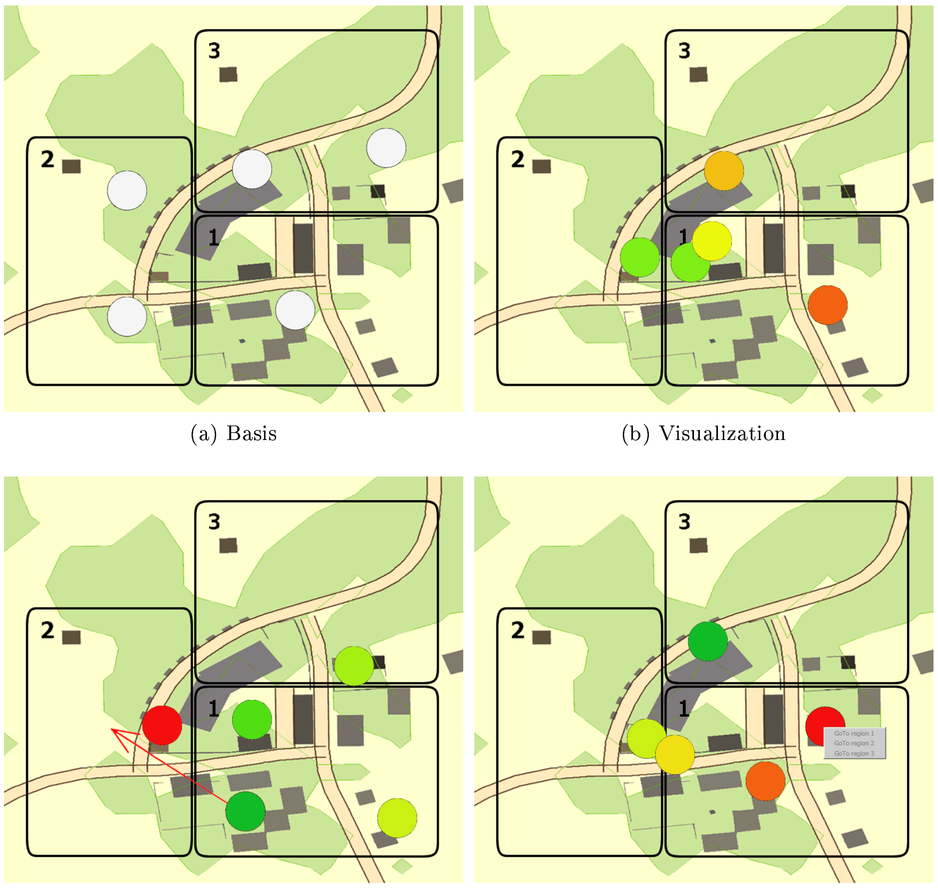
   As in the previous condition, the commander makes all decisions himself and the system does not provide support with the decision making process.

3. In the 'Decide' phase, a choice out of all the alternatives is made. The system calculates possible task allocation solutions and selects the best alternative. By advising the team leader, the system support the team leader in the Decide phase.
   The interface looks the same as in the previous condition. On top of this, a model calculates the perfect distribution of the soldiers based on the stress

values, and presents this to the commander every 10 seconds with arrows. The commander can obey or refuse to change positions, but the model will continue to give advice. Because of the advice component, this step will be called Advice condition in the experiment.

4. The last step in the OODA loop is 'Act', which corresponds to the Automatic condition. The system will support the team leader in the last step, which is acting and execution of the selected choice. Because no real support can be given in execution of the advice, the system will automatically perform this change. The team leader has no control over the situation and the model will be in charge of the decision making process and acting upon it. This condition is created to test the performance of the decision support model without humans intervening in the decision making process. Every ten seconds, the Automatic system will carry out the same calculation to search for the best array as the Advice model does. The only difference is that the Automatic system will not give the advice, but acts on the calculation right away and re-allocates the soldiers according to the newly found best array.
   The interface looks the same as the Visualization condition; the only difference is that team leader can not interact with the interface any more. When clicking on the soldiers in order to re-allocate them, a greyed unclickable pop-up is shown instead of a clickable pop-up in the previous three conditions.

The main difference between the different conditions is the soldier representation in the user interface. The user interface consists of two panels: one panel with the graphical representation of the map and one panel with the command list to execute commands. On the map panel, the soldiers are represented by dots and the regions by black rounded squares with numbers. The commander can zoom in and out within the map panel by using the scroll wheel of the mouse. The map that is presented to the commander in different conditions is shown in figure 15.



(a) Basis (b) Visualization

(c) Advice (d) Automatic

Figure 15: Interface of the commander in different conditions

The basis condition has no stress indication, hence the white color. The visualization condition shows the stress value of the soldier in colors ranging from red to green. The advice condition also shows red arrows when task re-allocation is adviced. The automatic condition shows the stress values of the soldiers; however, the commander can not take actions on this information.
The second panel is the command list with a button 'Send commands'. A screenshot of both panels is shown in figure 16.

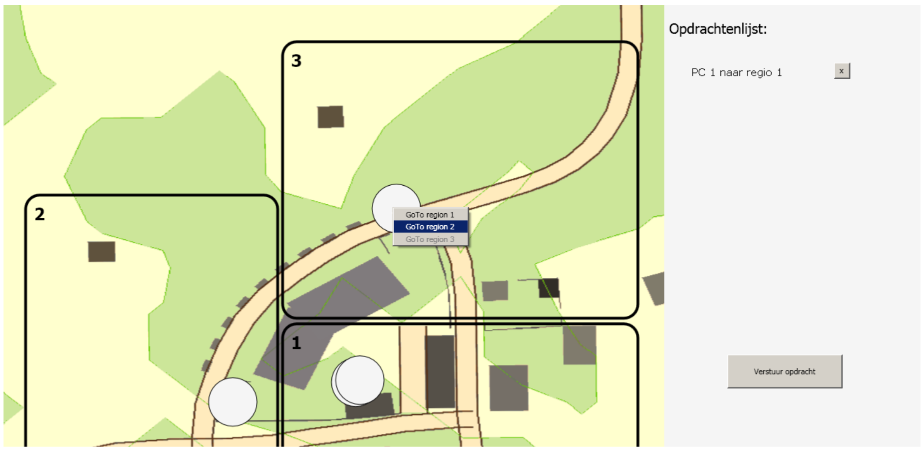
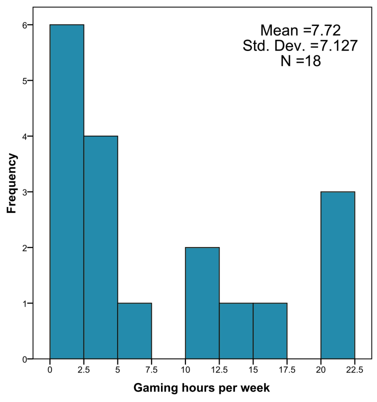


Figure 16: Screenshot of the interface and interaction possibilities

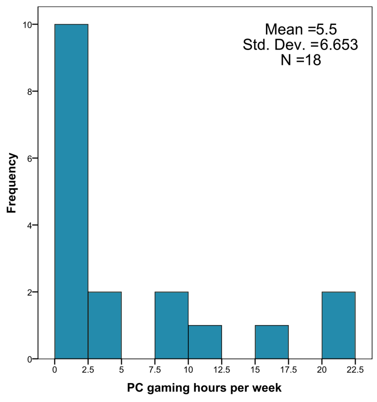
In the second (right) panel, the commander can make a list of commands to perform simultaneously, but once a command is added to the list, he can remove it before sending it. To put a command in the list, the commander has to click on a soldier with the right mouse button. A menu will pop up, which gives options for re-allocating the soldier to another region. In the soldierlist, a maximum of five commands can be gathered: because one soldier can only go to one region, the last command for the soldier is saved in the list. For example, when the commander sends soldier 1 to region 1, and then sends soldier 1 to region 2, only the last command will be in the list and ready to send. After sending the commands, the list is cleared and the commander can see the results of his changes after a few seconds on the map, where the soldiers will change positions. The commander can send the soldiers to three regions, which have a teleport spot that is somewhat sheltered. The commanders can not point to a specific location to send the soldiers to, but he has to send them to one of the three locations. The soldiers will just change position in the game without any warning, even when they were in the middle of an action.

5.3 Participants

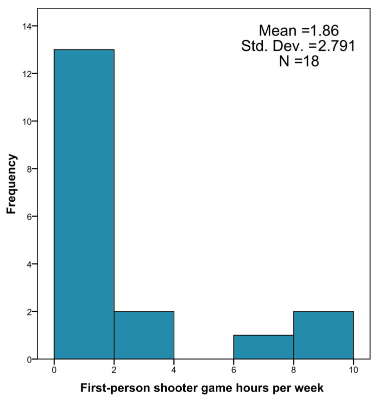
In this experiment, 30 people participated. All participants had some gaming experience with first-person shooter games and were in good health. Because two sessions were not usable retrospectively, the data of 18 participants is used in the data analysis. Of these 18 participants, 4 were women and 14 men. The participants were on average 21 years old ($\sigma = 3.12$). Overall, they reported to play a virtual game for 7.7 hours per week ($\sigma = 7.13$, $Med = 4$, $IQR = 12$), from which 5.5 hours per week on a pc ($\sigma = 6.65$, $Med = 2$, $IQR = 8$). Of the 7.7 hours of virtual gaming, they play 1.86 hours first-person shooter games ($\sigma = 2.79$, $Med = 1$, $IQR = 2$). Histograms of these last three variables are shown in figure 17.



(a) Gaming hours



(b) PC gaming hours



(c) First-person shooter gaming hours

Figure 17: Histograms of the three gaming variables

5.4 Design

Because the soldier-role has to be played by several participants, the experiment is conducted with five soldiers and one commander for each experiment session. The soldiers play the virtual game while the commander has to allocate them to the right tasks. The study design is $4*2$: each team leader will work with all 4 support conditions and within these 4 conditions, the difficulty of the task is manipulated in two ways: easy or hard. Each of these 4 conditions can thus be either easy or hard, which means 8 possible combinations. The commander and the soldiers play the game 28 minutes consecutively, which is called one block. Within each block, four games of each 7 minutes can be distinguished, where each support condition with altering difficulty are presented to the commander. The combination $4 * 2$ is a between-subjects design, because each team leader will be faced with only half of the support condition-difficulty combinations; $4 * 2$ gives 8 possibilities, but the team leader can only have 1 per game, which means 4 possibilities. The soldiers do not notice the change between games; they play 28 minutes in the same game. After each block, one participant who was a soldier becomes the commander. This way, data of 6 commanders and $5*6 = 30$ soldiers can be gathered per session. The design is shown in figure 18.

SESSION								
	Block 1				Block 2			
	Game A	Game B	Game C	Game D	Game A	Game B	Game C	Game D
Commander condition	1	2	4	3	3	4	2	1
Soldiers	Play game				Play game			
	Block 3				Block 4			
	Game A	Game B	Game C	Game D	Game A	Game B	Game C	Game D
Commander condition	3	1	2	4	1	4	3	2
Soldiers	Play game				Play game			
	Block 5				Block 6			
	Game A	Game B	Game C	Game D	Game A	Game B	Game C	Game D
Commander condition	2	4	1	3	2	3	4	1
Soldiers	Play game				Play game			

Figure 18: The experiment design

The white boxes represent the easy condition when varying the difficulty of the commander task and the grey boxes the hard condition. The conditions are as fairly distributed as possible to minimize the possibility that the sequence of difficulties or support conditions influences the results.
The commander and soldiers are sitting in the same room, but the commander can not see the soldiers or the screens where they play the game on. The soldiers were placed in a way they could not see each others screens. The experiment setting is depicted in figure 19.

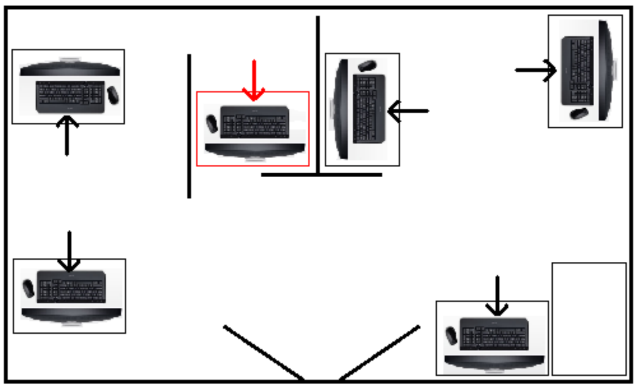
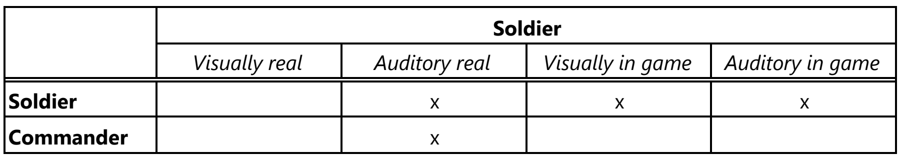


Figure 19: The experiment set-up, with the commander sitting at the desk with the red arrow

Each soldier had their own desk with a laptop that was connected to a monitor, a keyboard, a mouse, the Mobi8 (via Bluetooth), the stress vest and a headset (see section 5.8). The commander was allowed to talk to the soldiers, but on the condition that the commander takes the lead in asking inquiries. The soldiers were only allowed to talk to the commander when they wanted to share urgent and important information. The information flows between soldier and soldier and soldier and commander are shown in figure 20.



	Soldier			
	Visually real	*Auditory real*	*Visually in game*	*Auditory in game*
Soldier		x	x	x
Commander		x		

Figure 20: Interactions between soldier and soldier, and commander and soldier

The soldiers cannot see each other visually in real; they are sitting in the same room, but when they are playing they can not see the facial expression and the screen of the other soldiers. They can hear noises they make or things other soldiers say. In the game, they play a soldier. These soldiers can see each other in the game, and they make sounds occasionally (for example, when a person is hit, he moans a bit). The commander can not see the soldiers in real (see figure 19), but he can communicate with them and he can hear noises they make. However, because the commander is not in the game, he can not see the soldiers visually or hear them in the game. The commander does get some extra information about the soldiers, which depends on the condition. In all conditions, the commander has an overview of the map of the environment and the exact positions of the soldiers. In some conditions, he knows all the stress values of the soldiers.

5.5 Independent variables

An overview of the independent and dependent variables is shown in figure 21.

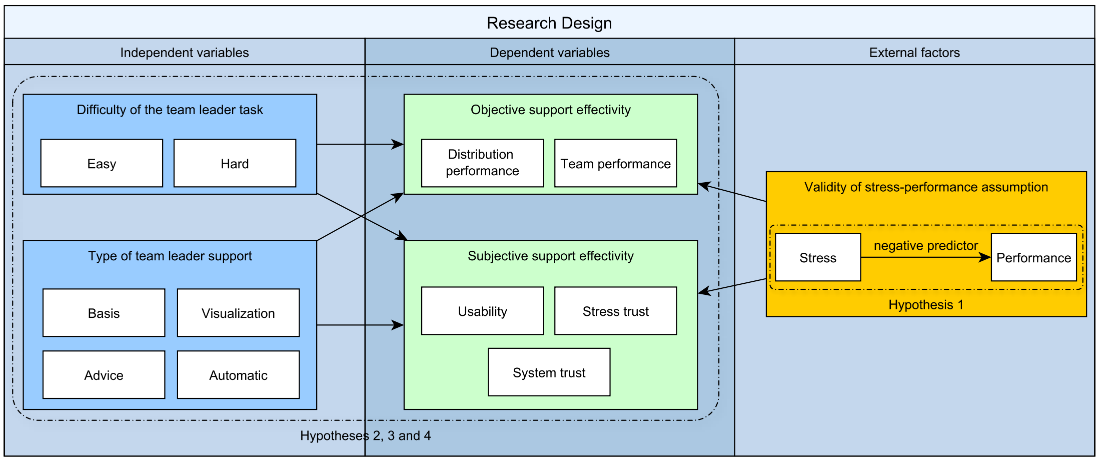


Figure 21: Overview of the research design

The difficulty of the decision task and the type of decision support are manipulated. This manipulation will presumably result in a change in the variables objective and subjective support effectivity, which are the dependent variables. The validity of the assumption that stress has a negative effect on performance influences these dependent variables also, which should be taken into account in the analysis of the results.

5.5.1 Difficulty of the team leader task

The difficulty of the task is manipulated to test whether the decision support system is more helpful with a more difficult team leader task. Per session, data of six commanders is obtained, who each carry out two easy and two hard tasks. The difficulty is manipulated by the frequency with which the spawn points of the enemies are changed. The team leader task is distributing the team members to the appropriate tasks. If the difficulty of the tasks changes frequently, the distribution should be altered more often.
Each time an enemy dies, he is spawned on one of the three spawn points depicted in figure 12. A special program (called Experiment controller, see section 5.7) decides to which spawn point, and takes the difficulty of the decision task into account. This program gives the region to which all died enemies should be spawned, an this region changes with a certain frequency. In the easy condition, a change of region occurs between 60 and 120 seconds. In the hard condition, this change is made between 0 and 60 seconds. Thus, in the easy condition the attack focus changes on average 19 times $((28 * 60)/90)$, and in the hard condition 56 times $((28 * 60)/30)$ per block.

5.5.2 Type of team leader support

We manipulate the type of decision support to find out whether the type of decision support is related to the performance of the commander. In other words: does the commander perform better with one type of decision support in comparison to the others? The experimental conditions explained in more detail:

1. Basis: the commander can communicate with each soldier and has a screen with the positions of each soldier plotted on a map. This is the only way to estimate the 'attack-load' of the soldiers in each area.
2. Visualization: the commander can communicate with each soldier and has a screen with information to support his decision making. For each soldier, he can see the stress value and position plotted on a map.
3. Advice: same as visualization support, except that this system advises the optimal distribution of soldiers by using the stress value as an indicator for the number of enemies that are attacking in the particular area. The advice support is based on the assumption that the stress value is an indicator of the extent to which a soldier needs help. This way, the decision support system calculates an optimal distribution of the soldiers and presents this to the commander. Note that even this support will not give an order to the commander, or automatically make a decision. The system will support the commander with advice, but the commander is free to accept or ignore the advice.
4. Automatic: the decision support system will decide on its own (without commander interference) which actions to undertake. This is merely a comparison condition. The soldiers should not know that the decision

support system takes all decisions, so the commander will have to act like he decides.

Per session, data of 6 commanders is collected in a within-subjects manner; each commander is assigned to all conditions once. The experiment controller regulates when the support condition should be changed (every 7 minutes).

5.6 Dependent variables

By manipulating the aforementioned independent variables, we can measure other variables to find out whether they are related to the manipulated variables.

5.6.1 Objective support effectivity

The performance of the commander is used as a measure for the objective support effectivity. In each support condition, the performance of the commander is calculated using two measures: distribution performance and team performance.
To calculate the distribution performance, the user distribution is compared with an optimal distribution. Because all enemies have the same strength, the number of enemies in each region is used to determine the perfect distribution of the soldiers. The only difference with the calculation of the best array during the game, is the task demand calculation. Instead of using the average stress value in each region, the number of enemies nearby the village is used. This does not mean the number of enemies in sight of the soldiers, but only those enemies that are close enough to the village that they are in the regions depicted in figure 12. For example, the amount of enemies registered in region 1, 2 and 3 are 5, 3 and 5 respectively. After normalization, the task demand per task is 1.92, 1.16 and 1.92. These values are used to calculate the difference score of the user array with these known demands. This difference score is used as the distribution performance score. When the distribution score is 0, the user distribution perfectly meets the unknown task demand (in numbers of enemies). A high distribution score means a low score on agreements between the user array and the perfect array. Because a performance score is supposed to be high with a good performance and low with a bad performance, the distribution score is substracted from the maximum score (10) to get the distribution performance value. This value can range from 0 to 10 and indicates a high performance when the score is high, and a low performance when the score is low.
The second measure is team performance. The commander is responsible for his team. With his communication and allocation possibilities, the commander influences the team performance and it is thus an important measure in evaluation his performance. The team task is to protect the village from the enemies; the number of enemies nearby the village thus indicates how well the team carried out the task. The sum of enemies in all regions is used as the team score measure. For example, in the previous example of 5, 3 and 5 enemies in region 1, 2 and 3, the team score would be $5 + 3 + 5 = 13$. As with the distribution score, a high team score actually means a low score on keeping the enemies outside the village and this score also needs to be translated to a high score for high performance. The number of enemies in the game is fixed and is always 50. Substracting the team score from the maximum score results in a team performance value. This value can range from 0 to 50 and indicates a

high performance when the score is high, and a low performance when the score is low.
When no enemies are in the regions, no perfect distribution can be calculated. The datapoints with no enemies in the regions will have a missing value on distribution performance, but not on team performance.

5.6.2 Subjective support effectivity

After carrying out the commander task, participants are asked to fill in a short questionnaire. The questionnaire consists of three parts:

1. General usability of the support system: questions about satisfaction, learnability and usability of the system
2. Trust in the stress information: questions about the user's view on the relationship between measured stress and perceived stress and their trust in stress as a valuable input for the decision process
3. Trust in the support system: questions about the perceived task effectiveness, behavior, errors, and general trust in the decision support system.

These three parts are quesitoned for each condition. At the end of the questionnaire, the user is asked to fill in his/her favorite system, with a short explanation. The complete questionnaire is shown in appendix C.

5.7 System architecture

To carry out the experiment, different systems need to be connected. The most important components of the architecture are:

1. Computers to play the VBS2 game, including one game host
2. Experiment controller, to ensure that manipulations (such as regulating commander difficulty, switching conditions) are controlled
3. Stress monitors to measure stress
4. Vest managers to activate the pneumatic actuators in the vest at the right time
5. Database system to collect all the data from the VBS2 game, vest managers and experiment controller and provide the link to the support model
6. Support module to support the commander with up-to-date information
7. A network connecting all these components

A visual representation of the system architecture is provided in figure 22.

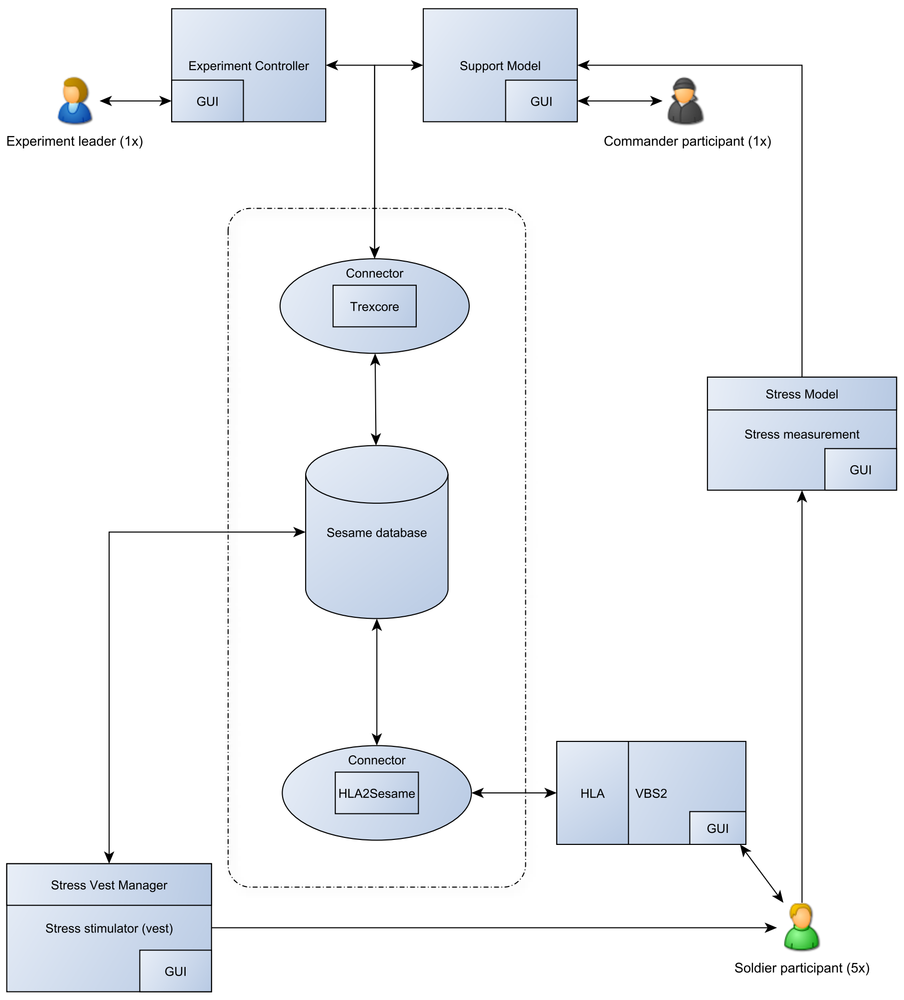


Figure 22: System architecture

The core of the system is the elements within the dotted area. The Sesame database is central to store data to provide other system elements with up-to-date information. The database is connected with VBS2 through a HLA2Sesame connector. This allows the database to store VBS2 data (such as when a soldier gets hit) and to send information to VBS2 (such as changing the enemies attack region). On the other side, the Sesame database has to exchange information with the C# modules at the top of the image. This is facilitated by a Trexcore connector, which allows the database to send information to the modules (such as passing on how many enemies are present in the regions) and also to allow the modules to pass information on to the database (such as when a vest should be activated). More information about Trex and VBS2 is provided in appendix B. On the upper left corner, the experiment controller is depicted. Its task is to make sure that the support model behaves according to the right condition. An XML file is used to match the right condition to the right block and time. When the experiment controller is started, a timer will run down for each game.

When a new game starts, the condition will change automatically. Besides keeping track of conditions and time, the experiment controller also regulates the commander difficulty. Depending on the easy or hard condition, the experiment controller will switch the attack region of the enemies with different frequencies. Next to the experiment controller is the support model. Its interface is presented to the commander and the commander can change the situation in the game through this interface. For example, when the commander re-assigns a soldier, the support model passes this through to the database, which passes it through to the HLA system that changes the position. The soldier sees that his position is changed through his interface.
On the right side, the stress model is shown. It includes a simple interface, where the stress monitor can be activated. The stress monitor calculates the stress every ten seconds and sends it over the network to the support model.
In the lower right corner, the interaction between the soldier and the VBS2 game is depicted. Because the soldiers are in the same game, a network needs to be set up to connect these computers. One of the soldier computers has to host the game.
In the lower left corner, it is shown that the vest manager communicates with the soldier (through the actuators in the vest) and the database. From the communication with the database, the vest knows when to activate the actuators.

5.8 Materials

The participants all had a laptop, mouse, keyboard, headphones and 17" monitor. The team leader used this laptop to interact with the decision support model, and the team members used it to play Virtual Battlespace 2 (version 1.4). To carry out their task, the team members were equipped with a stress monitor and a stressor vest. The stress monitor is a Mobi8 from TMSi for heart rate measurement, which was designed with a fingerclip to clip on the finger of the participant. The stress vest is a TN Games 3RD Space Vest which has eight 'active zones' to give the participant a blow of 30 pounds per square inch (psi) by using a aircompressor. More information about the hardware is provided in appendix B. Figure 23 shows photos to illustrate the materials used.

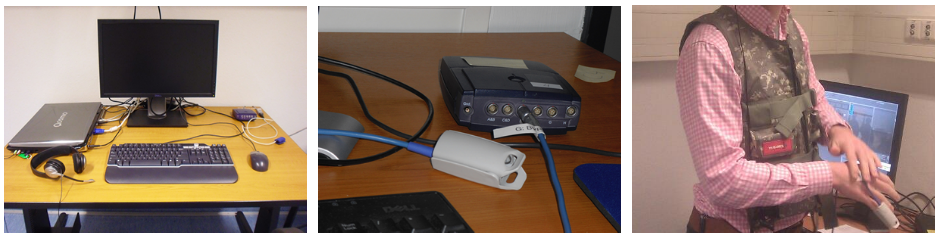

(a) Desk (b) Mobi8 (c) TN Vest

Figure 23: Materials used in the experiment

5.9 Procedure

The participants first fill in a general questionnaire to extract some demographic information. Also, a neuroticism questionnaire is filled in. The neuroticism questionnaire is used to check a person's perceived tendency to neuroticism to compare this with the average stress measured in the experiment. The neuroticism items of the Eysenck Personality Questionnaire (EPQ) are used, which are twelve questions that can be answered with either 'yes' or 'no'. The EPQ neuroticism is a self-reported scale has already been validated: Cronbach's $\alpha = 0.83$ (Sanderman, Arrindeel, Ranchor, Eysenck, & Eysenck, 1995). During the experiment, the Subjective Units of Distress (SUD) questionnaire is used to estimate the stress of the user by asking them to pick a number on a scale between 0 and 10, corresponding to different symptoms of stress. The SUD score has been widely used by counselors in both clinical and research applications and is valid as a measure of stress (Kaplan & Smith, 1995). All questionnaires used in this research can be found in appendix C. The participants are informed about the task descriptions and practical information by reading them (appendix D). After that, the experiment leaders introduce themselves and answer any questions of the participants. Then, each participant takes place behind a computer and is connected to the stress monitor. To let the participant familiarize with the game and the controls, and because the stress monitors need to be calibrated, a training session takes place. The participant have to rest for 1 minute, can try some controls in the next minute and then they will be heavily attacked by enemies for 2 minutes. After the training session, the real experiment session will begin. One of the participants is disconnected from the stress monitor and will play the commander role. The other participants will play the soldier role and play the actual game. The experiment consists of 6 blocks with each 4 games of 7 minutes, with small breaks between them to switch participants and fill in the SUD and the commander questionnaires (appendix C). Afterwards, a short debriefing is facilitated and the participants can go home. The time schema is shown in table 10.

Table 10: Procedure

Activity	Duration (minutes)
Pre-experiment questionnaires	10
Introduction of the experiment	5
Training / Calibration of stress monitors	15
Block 1	$4 * 7$
Interim questionnaire	5
Block 2	$4 * 7$
Interim questionnaire	5
Short break	5
Block 3	$4 * 7$
Interim questionnaire	5
Block 4	$4 * 7$
Interim questionnaire	5
Short break	5
Block 5	$4 * 7$
Interim questionnaire	5
Block 6	$4 * 7$
Interim questionnaire	5
Debriefing	2
Total	**240**

5.10 Data analysis

The data analysis of this experiment consists of two parts. The first part is to check the assumption that more stress causes a decrease in performance. Because the decision support system is based on this assumption, it is important to verify that the participants in the experiment performed less when they encountered more stress. The second part of the analysis is to evaluate the decision support systems. Both the objective and subjective support effectivity are analyzed.

5.10.1 Validation of stress-performance assumption

Hypothesis 1 states that stress is a negative predictor for performance. In order to reject or accept this hypothesis, this assumption must be checked.
Before the comparison between stress and performance is analyzed, it is important to check whether stress is really measured. In previous studies is found that people who score high on neuroticism are more likely to be stressed than people who score low on that scale. This relationship can be found to correlate the average stress value across all blocks of each participant with their neuroticism score. A positive correlation would mean that a higher neuroticism score corre-

sponds to a higher average stress value, which would confirm the assumptions of previous researches.
Another way to check if the measured stress is in accordance with the 'real' stress, self-reported stress of the participants can be compared with the measured stress. It is expected that the self-reported stress roughly corresponds to the measured stress. A positive correlation between the average self-reported stress and the average measured stress across blocks would mean that a higher self-reported stress corresponds to a higher measured stress.
Elevated stress values can originate in high workload or a difficult task. To check this, an anlysis is carried out where the average number of enemies in a region is plotted against the average measured stress value of the team members. A positive correlation indicates that more opposition results in a higher stress value.
Checking the assumption that more stress means less performance can be tackled in two ways: checking the micro effects of stress on performance and checking the macro effects. The performance measures are the same in both analyses:

- Hitcount: number of times the soldiers are hit by the enemies
- Enemieshit: number of times the soldier hit an enemy
- Accuracy: ratio between number of times shot and number of times an enemy was hit by a shot

The micro level analysis is focused on one participant and analyzes datapoints (per 10 seconds) within one block. A cross-correlation analysis is carried out to check whether any pattern emerges when relating stress and the different performance measures on different time lags. For each datapoint that matches a stress value to a performance value, the correlations for 6 datapoints before and after the actual time match are calculated to find any delays in stress reaction. For example, a negative correlation between stress and performance may not be found on the paired datapoints, but a correlation may exist between stress and the performance measure of two datapoints later. When some pattern is found, this means that stress and performance are related, but with some delay. For this analysis, lags between -6 and +6 datapoints are included, therefore calculating all possible correlations of one minute ($6 * 10$ seconds) before and one minute after the actual time match.
At macro level, the average stress value and the average performance per participant are compared every minute. A negative correlation between the stress and the performance measures would mean that stress negatively influences the performance and this would confirm the assumption where the decision support model is based on.

5.10.2 Evaluation of decision support effectivity

Hypothesis 2, 3 and 4 concern the decision support effectivity. Two variables are manipulated in the experiment: the type of decision support and the difficulty of the commander task. As discussed in the previous chapter, the difficulty of the commander task is programmed beforehand. During the experiment, the difficulty manipulation is automatically handled by the experiment controller program. Varying the difficulty for the commander is implemented by varying the frequency of changing the regions from where the enemies attack the village. Two conditions are used: easy, which represents little change of attack regions,

and hard, which represents frequent change of attack regions. Eventually, frequently switching attack regions should result in more required switches of task allocation of soldiers. This would make the task of the commander more difficult.
To verify that this manipulation had the desired effect, the number of switches per block needs to be analysed for both the easy and the hard condition. The data of the perfect distribution (which is calculated originally to calculate distribution performance) is used to determine the number of required switches per game to reach the perfect distribution. Within each block, two games have the easy condition and two games the hard condition. The number of switches is analyzed per block, which means that for each difficulty condition, the average value of number of switches per game is used. A significant difference of the required number of switches between the difficulty conditions would indicate that the manipulation had the desired effect. A non-significant difference would mean that the difficulty conditions can not be distinguished and therefore should be excluded from the rest of the analysis.
The manipulation of type of decision support is used to make a distinction between different types of decision support and their effect on objective and subjective support effectivity. Both variables are analyzed to find differences between the independent variable groups.
The objective support effectivity is analyzed by carrying out an ANCOVA analysis where the two indicators of support effectivity (distribution and team performance) are compared between the four support condition groups. Because some soldier related variables (such as stress and performance) can influence the scores on distribution and team performance, an ANCOVA analysis is used to correct for these covariants. A Sidak test is used to analyze any significant results found in the ANCOVA analysis.
A more in depth analysis is checking whether the number of switches the user made in each condition varies between conditions. The number of switches is the number of times the task allocation changed. Two different types were logged during the experiment: the number of times the user (or the model, in the Automatic condition) re-allocated tasks, and the number of times the perfect allocation of tasks changed. When this allocation would be followed, the highest score on distribution performance would be achieved. This number of changes can be considered the required number of re-allocations to reach the perfect score; note that it does not tell anything about the quality of the re-allocation.
The subjective support effectivity is tested by gathering answers on some open and multiple choice questions, shown in appendix C. For each condition, the participants judged the usability of each system and the degree of trust they had in the system. Also, they rated the trust they had in the stress model; how well do the stress values correspond to the perceived stress value of the team members when the commander asks them? Those three constructs were measured by a number of questions which were expected to reflect these constructs. The coherence of these questions within one construct should be reliable (Cronbach's α should be above 0.7). The differences between the support conditions on these constructs can be analyzed by using an ANOVA test. Any significant differences indicate differences between conditions on one of these factors. On the open questions, some directions or explanations can be extracted to explain differences found in the quantitative comparisons between the support conditions.

6 Results

6.1 Data gathering and processing

From the five sessions that were carried out, three sessions were useful to analyze because of major changes in experiment setting after two blocks. Within these three sessions, due to time constraints, 14 blocks of the expected 18 could be completed. Of these 14 blocks, 3 blocks were not valid for the team leader task. Because one of the stress monitors was broken, the stress value that was passed to the decision support system was not correct. Therefore, the entire system could not be evaluated and these blocks are removed from the dataset. The experiment produced 11 blocks of datapoints on the team leader task.
On the team member task, more results could be obtained to evaluate the stress-performance relationship. Because 11 blocks could be completed where each block involved five soldiers, data of $11 * 5 = 55$ blocks could have been obtained. Some blocks were removed from the dataset, because the stress values were either missing (2 blocks) or not distinctive enough (4 blocks) because of malfunction of the stress monitor. This means that $55-2-4 = 49$ blocks remain to be analyzed. For each block, the number of datapoints will approximate 56 (two datapoints per minute, for 28 minutes), but some logfiles did not capture every possible datapoint. In total, 609 datapoints could be analyzed for the team leader task, and 2717 datapoints for the team member task.
All bar graphs used in this analysis always include error bars. These error bars represent 95% of the Confidence Interval.

6.2 Validation of stress-performance assumption

Because the decision support model is based on the assumption that more stress will decrease the performance, both stress and performance were measured to check this assumption.

6.2.1 Measured stress versus neuroticism score

To verify whether the concept of stress was measured in the experiment, the scores of the participants on the neuroticism scale and the average stress values of the participants were compared. The higher the neuroticism value, the more a person should be reactive to stressors. This comparison is visualized in the scatterplot of figure 24.

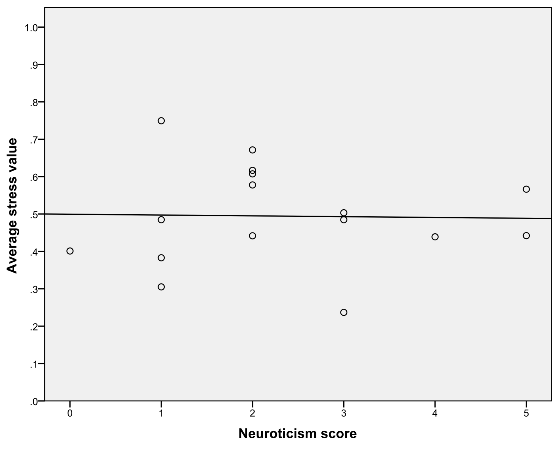


Figure 24: The average measured stress values plotted against the score on the neuroticism questionnaire

A relationship between the measured stress in the game and the neuroticism scores is not found ($\tau = -0.009$, $p = 0.963$).

6.2.2 Measured stress versus self-reported stress

After each block, the participants filled in a Subjective Units of Distress (SUD) form (appendix C). These scores indicate how much stress the participant experienced during each block. The mean scores on the SUD are plotted against the measured stress values and shown in figure 25.

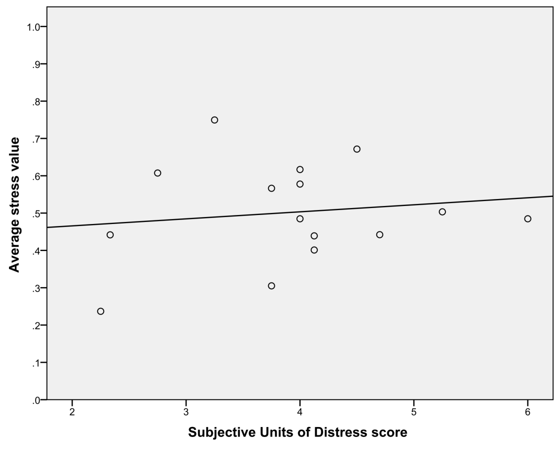


Figure 25: The average measured stress values plotted against the mean scores on the Subjective Units of Distress (SUD)

No relationship between the self-reported stress and the measured stress was found, $\tau = 0.069$, $p = 0.727$.

6.2.3 Measured stress versus average number of enemies

Because the number of enemies in each region was logged, the enemies in the region of the participant could be calculated across all datapoints. For each participant, the average number of enemies he faced in the region was calculated and plotted against the average stress, see figure 26.

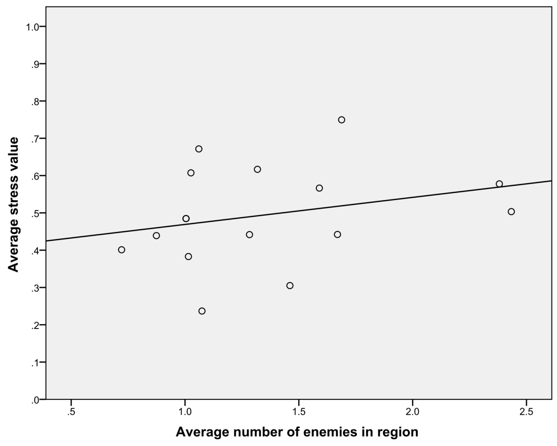


Figure 26: The average number of enemies a participant had to face against the average stress value

The correlation between these two variables is positive, though not significant ($\tau = 0.259$, $p = 0.162$). Facing more enemies is assumed to be more difficult, so a positive significant result would substantiate the decision support model reasoning.

6.2.4 Micro and macro effects of stress on performance

First, the small fluctuations of stress and its influence on the different performance measures are described, also called micro effects. Secondly, more long-term relationships of stress and performance are analyzed, which are called macro effects.

The micro effects of stress can be analyzed by looking at the values of stress every 10 seconds, and the values of the performance measures. For every participant and every block, the stress values and the three performance measures were used in a cross-correlation analysis. An example of such a cross-correlation analysis is depicted in figure 27.

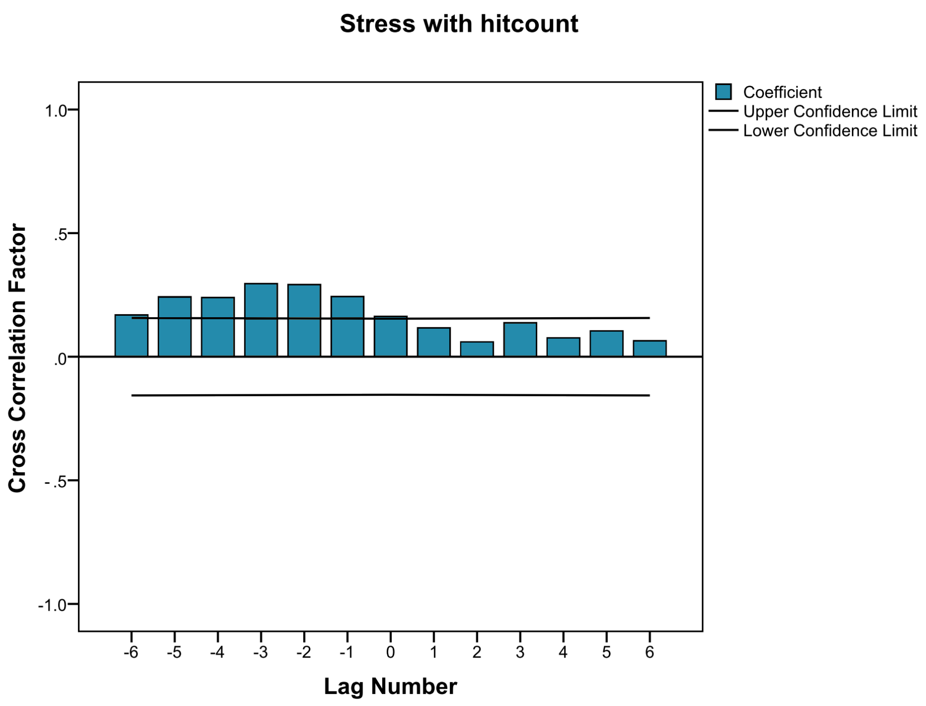


Figure 27: An example of one participant in one block, where the cross-correlation between stress and hitcount is showed

This figure shows the relation between stress and hitcount. On the left (negative) side, the influence of hitcount on stress is showed. The bar with zero lag represents the 'normal' correlation. On the right (positive) side, the influence of stress on hitcount is presented. On lag -2, the correlation is the most strong (0.302). From this graph can be deduced that on average, 20 seconds after this participant is hit by an enemy, the stresslevel of this participant increases. Unfortunately, not all participants produce these results. The variety of both negative and positive correlations and both negative and positive lags show that participants in this experiment did not show a uniform pattern of responding to stress and the influence of events on stress and stress on performance in a short timewindow.

For the macro effects analysis, scores of stress and performance were logged every minute. The stress value represents the average stressvalue in that minute, and the performance measures represent the added performance in that minute. The participants played several blocks during which their stress and performance was measured. To see whether participants adjusted to their task during the experiment, stress data of all participants are ordered whether it was their first, second, third of fourth game.

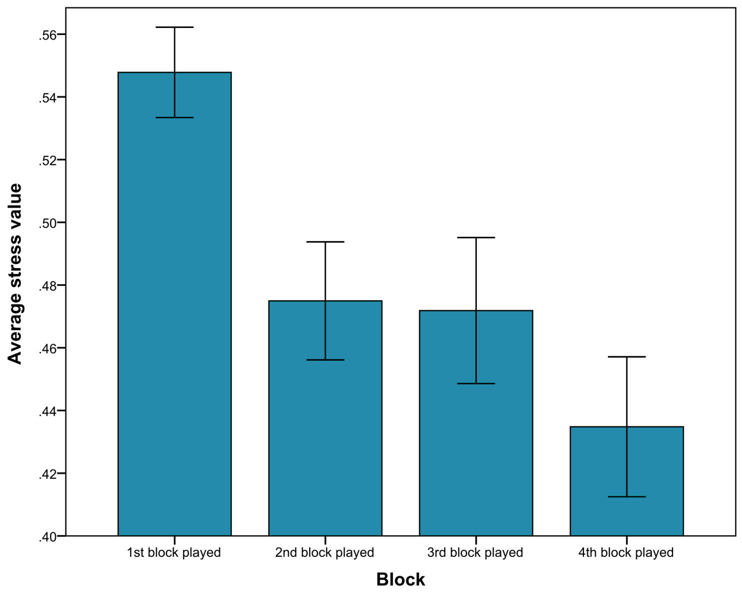
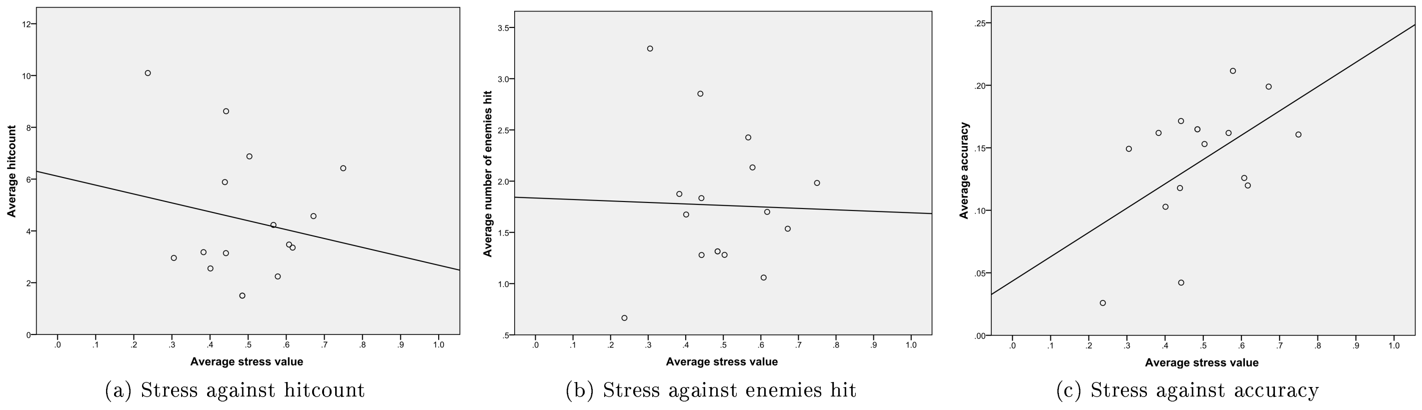


Figure 28: The average stress values plotted per block

Both this figure and an ANOVA test $F(3, 2645) = 28.83$, $p < 0.001$ show that less stress is measured when participants perform the task more often. Using the Games-Howell procedure, significant differences were found between the first block and all other blocks ($p < 0.001$) and between the second and fourth block ($p = 0.035$).
To evaluate performance, the three performance measures are plotted against stress individually, which is shown in figure 29.

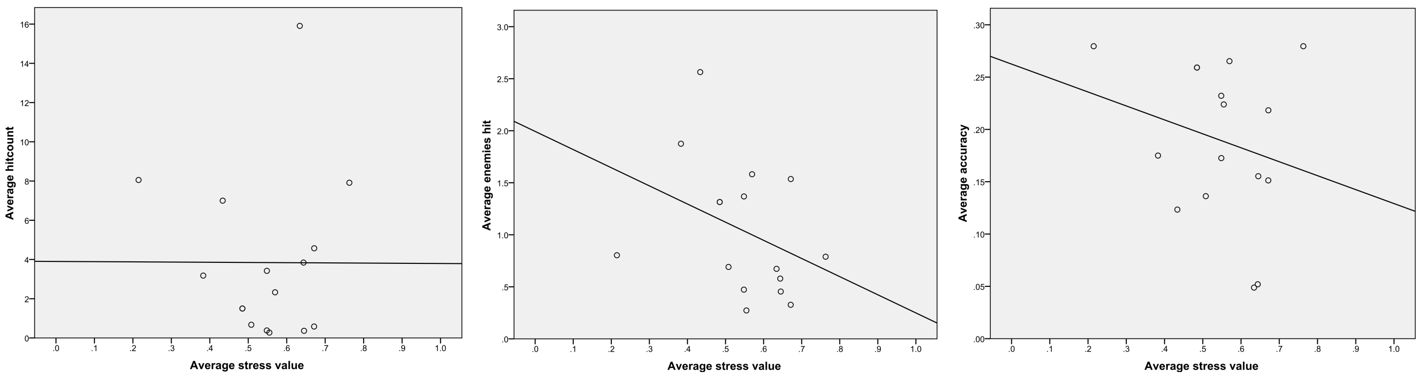


(a) Stress against hitcount (b) Stress against enemies hit (c) Stress against accuracy

Figure 29: Relationships between stress and performance measures across all blocks

The correlations between stress and hitcount and stress and number of enemies hit are not significant ($r_s = -0.063$, $p = 0.816$ and $r = -0.029$, $p = 0.916$ respectively). However, there is a correlation between stress and accuracy, $r =$

0.520, $p = 0.039$. This indicates that team members with high stress were more accurate in shooting enemies than team members with low stress.
Because figure 28 showed that stress values in block 1 were significantly higher than in the other blocks, the analyses for the relationship between stress and the three performance measures are repeated when using only data of soldiers when they played the game for the first time.



(a) Stress against hitcount (b) Stress against enemies hit (c) Stress against accuracy

Figure 30: Relationships between stress and performance measures of the first block played

These results are more consistent with the expectations, although the correlations are not significant ($r_s = -0.003$, $p = 0.990$; $r_s = -0.361$, $p = 0.169$; and $r_s = -0.236$, $p = 0.380$ respectively). When comparing these results to figure 29, the performance seems to be influenced by stress in the way we expected, but only when stress is high enough. In the discussion, the differences between both figures will be further discussed.
From these analyses, small indications point in the direction that stress might be a negative predictor for performance when stress is high enough. However, these results were not significant. The analyses across all blocks indicated that only the relationship between stress and accuracy yielded significant results and the direction of this relationship was positive (where a negative relationship was assumed). Therefore,

> Hypothesis 1: The stress value of the team members is a negative predictor for their performance

can not be accepted.

6.3 Evaluation of decision support effectivity

The two independent variables that may influence the support effectivity are commander task difficulty and support condition. To verify that the (automated) task difficulty manipulation actually took place, the number of switches made by the perfect distribution are analyzed in both the easy and hard condition. In figure 31, an error bar graph is plotted which shows the average number of switches per level of difficulty.

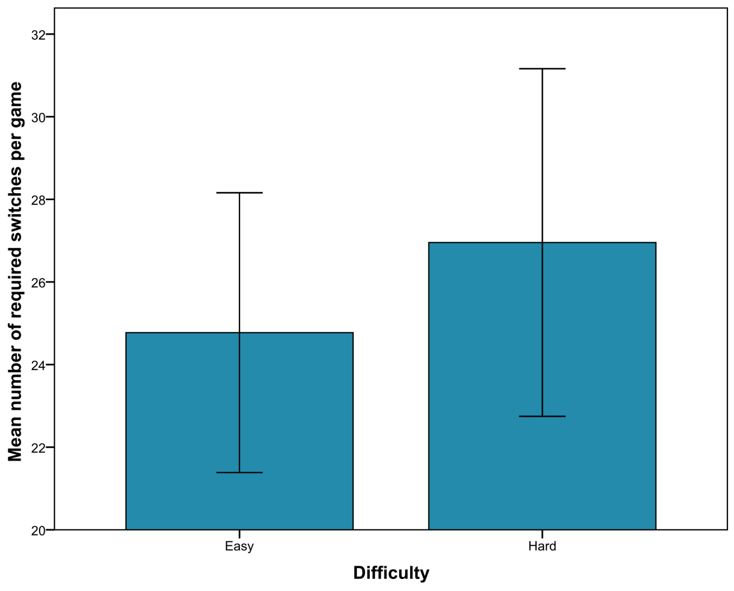


Figure 31: Error bar graph of commander difficulty

An ANCOVA test is used because the number of datapoints for the difficulty condition is not equal and an effect of the commander support condition might influence the comparison between these unequal groups. The results show no difference between difficulty scores, $F(1, 42) = 0.773$, $p = 0.384$ and *partial* $\eta^2 = 0.12$. This result shows that manipulating the frequency of switching attack regions did not result in the need to change the soldier distribution. In the rest of the analysis, these difficulty conditions will not be analyzed separately any more.

To evaluate the decision support effectivity, two analyses can be carried out: based on the data logged during the experiment, and based on the subjective user input that was gathered after the commander carried out his task.

6.3.1 Objective support effectivity

Objective support effectivity consists of distribution performance and team performance. These two performance scores are depicted in figure 32.

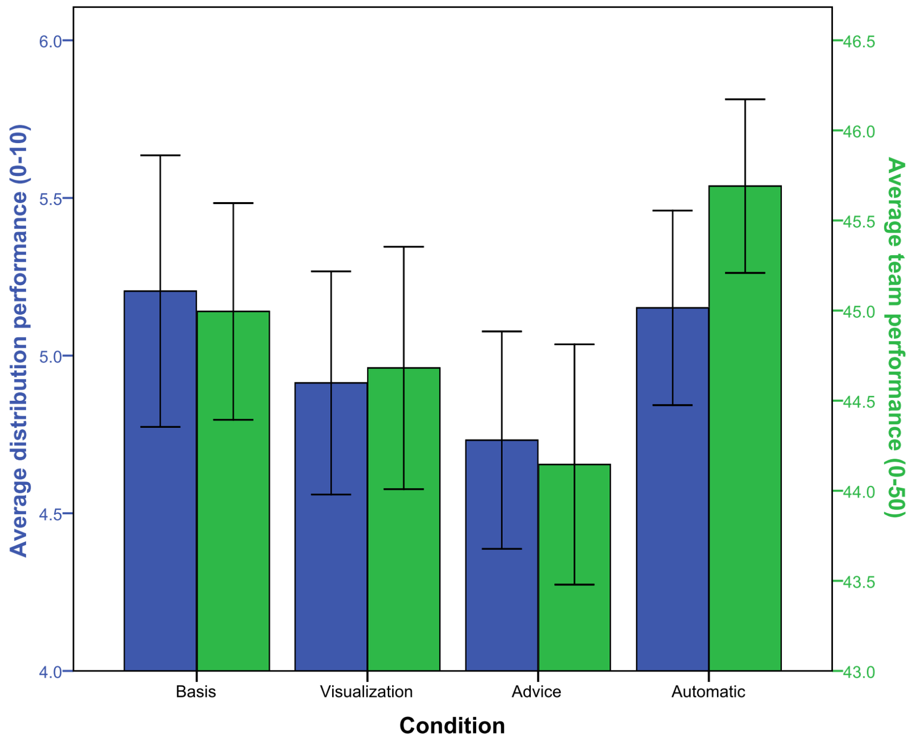


Figure 32: Average distribution performance (y1) and team performance (y2), calculated for each condition

From this figure, a trend can be seen where commanders in the advice condition seem to perform worse than commanders in the other conditions. To test whether these differences are significant, an ANCOVA analysis is carried out for both distribution performance and team performance. On distribution performance, average enemieshit ($F(2, 543) = 8.71$, $p = 0.003$, $r = 0.13$), hitcount ($F(2, 543) = 13.25$, $p < 0.001$, $r = 0.15$) and accuracy ($F(2, 543) = 6.24$, $p = 0.013$, $r = 0.11$) were significantly related to the commander's distribution performance. There was no significant effect of commander condition on commander distribution performance after controlling for the effect of the variables mentioned earlier, $F(3, 543) = 2.36$, $p = 0.071$, $partial\ \eta^2 = 0.013$.
On team performance, average triggercount ($F(2, 543) = 4.36$, $p = 0.037$, $r = 0.089$), enemieshit ($F(2, 543) = 4.06$, $p = 0.045$, $r = 0.086$) and hitcount ($F(2, 543) = 4.42$, $p = 0.036$, $r = 0.090$) were significantly related to the commander's team performance. There was also a significant effect of commander condition on commander team performance after controlling for the effect of the variables mentioned earlier, $F(3, 543) = 5.16$, $p = 0.002$, $partial\ \eta^2 = 0.028$. Sidak's test shows that only the advice and automatic condition significantly differed from each other, $p = 0.001$; the other conditions were not significantly differing from each other.
These results indicate that the model in the Automatic condition performs better than commanders in the Advice condition on team performance.

The results of the analysis on the differences between the number of switches the user made in each condition are shown in figure 33.

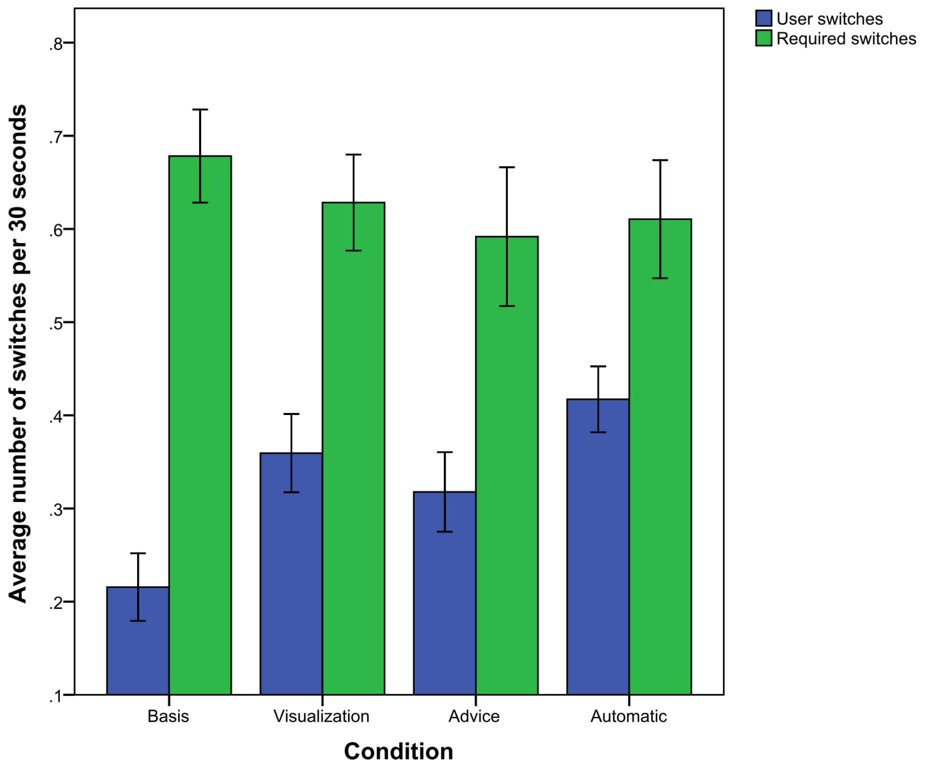


Figure 33: Mean number of switches of the user or model (blue) and number of switches required to reach perfect distribution (green), calculated for each condition

An ANCOVA is used to check differences between groups for both variables and correct at the same time for covariants. On the number of switches of the perfect task allocation, no difference between groups is found, $F(3, 192) = 2.078$, $p = 0.105$, *partial* $\eta^2 = 0.15$. This means that the required number of switches to reach the perfect task-allocation is not dependent on the condition.
On the number of switches by the user or model, hitcount ($F(2, 191) = 5.32$, $p = 0.022$, $r = 0.15$) is significantly related to the number of user switches. There was a significant effect of commander condition on the number of user switches after controlling for the effect of the covariant hitcount, $F(3, 192) = 21.491$, $p < 0.001$, *partial* $\eta^2 = 0.26$. Sidak's test shows that the Basis condition differs significantly from Visualization ($p < 0.001$), Advice ($p < 0.001$) and Automatic ($p < 0.001$) on the number of switches by user or model. Also, the conditions Automatic and Advice differ significantly, $p = 0.004$. The other conditions do not differ significantly from each other. This means that the user made significantly less switches in the Basis condition than in the Visualization, Advice and Automatic conditions, and the commanders in the Automatic condition also made significantly more switches than the Advice condition.

6.3.2 Subjective support effectivity

The participants filled in a questionnaire after they carried out the commander task. This questionnaire is shown in appendix C.
On the multiple choice questions, the coherence between the constructs had sufficient reliability: usability had good reliability, Cronbach's $\alpha = 0.862$ and trust in the system and trust in the stress model both had acceptable reliability, Cronbach's $\alpha = 0.797$ and Cronbach's $\alpha = 0.784$ respectively.
The rating of the commanders on these constructs is depicted in figure 34. The Basis and Automatic condition miss some bars: the Basis condition did not have any stress input, so the stress trust can not be measured. The Automatic conditions misses both usability and stress bars, because the commander could not interact with the system in this condition.

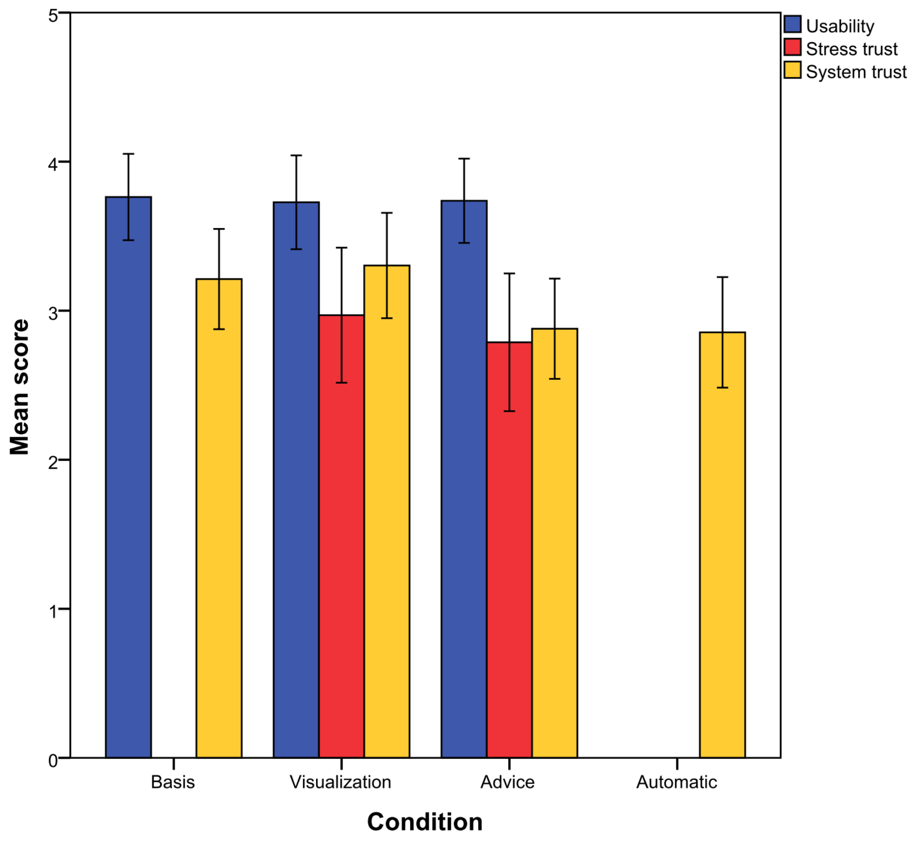


Figure 34: Subjective rating on usability, stress trust and system trust for each condition

No significant differences were found on the subjective ratings between conditions; usability $F(2,30) = 0.019$, $p = 0.981$, stress trust $F(1,20) = 0.392$, $p = 0.538$, and system trust $F(3,40) = 2.130$, $p = 0.112$.
The last question of the commander questionnaire yielded new insights though. In this question, the commander was asked to choose his favorite system, with a short explanation why he had chosen that system. Out of 11 responses, 7

chose the advice system and 4 the visualization system. All the responses on the open questions are shown in appendix E (in Dutch). These responses are summarized and translated. This is shown in table 11.

Condition	Mentioned advantages	Mentioned disadvantages
Basis		• Without any indication of the situation of soldiers, you can't ask specific questions • Stress is considered a valuable input, which is not present in this condition
Visualization	• Useful to see stress values, as an inducement to ask inquiries • Best overview of the situation • Feeling of control over the team	• Stress values did not always correspond to self-report of team members
Advice	• Getting as much information as possible benefits the decision making • I can compare the information of the pc and my own opinion	• Advice was confusing, because it sometimes recommended illogical actions • Model behind the advice is not reliable, so the advice is of little value
Automatic		• Unpredictable and re-allocated tasks randomly and too frequent • The system should announce his decisions (= task re-allocation) to the team members • When stress values are the only source to base decisions upon, the decisions can't be good

Table 11: Summary of open question responses

The participants prefer the advice system over the visualization, baseline or automatic system. They think that more information will lead to a better informed decision. Also, they report that the automatic system is "unpredictable" and "works randomly"; they think that the automatic system can not make good decisions. However, the data analysis shows that the automatic system performed significantly better on team performance than the advice condition. The subjective user input provided some leads on explaining the quantitative results. This will be discussed in the Discussion section.

The hypotheses composed in section 4 can now be evaluated.

> Hypothesis 2: Communicating stress values of team members will increase the team leader performance compared to the situation where no decision support information is provided.

No significant differences were found between the visualization condition and the basis condition. This hypothesis can not be accepted.

> Hypothesis 3: Execution of the team leader task by the decision support model without human team leader interference will outperform the human team leader without decision support information.

No significant differences were found between the automatic condition and the basis condition. This hypothesis can not be accepted.

> Hypothesis 4: Combining human and computer reasoning will lead to the best team leader performance.

The team leader performance consists of both distribution performance and team performance. On the distribution performance, no significant differences were found. On team performance, the automatic condition performed better than the advice condition, meaning that this hypothesis should be rejected.

7 Discussion

Strengths of this research are the insights gained by measuring both real-time stress and performance, testing different decision support types in a within-subject way, and the holistic approach of setting up the experiment. First of all, the relationship between stress and performance is studied with a new type of stressor: the gaming vest, which responds according to actions of the participants in the game. Using a virtual game to explore the relationship between stress and performance is an easy task set-up, because performance can be measured directly in the game and the real-time stress measurements facilitated easy comparison between these two variables. The second strength of the study is testing different decision support types in a within-subject way. Many studies divide decision support in various levels, where the most elementary division is information support (information acquisition and analysis) on one side, and decision selection and action implementation on the other (Rovira et al., 2007). These two were compared with no support (control condition) and fully automated support (essentially also a control condition). Because the design was developed to confront each team leader with each condition, the results of this comparison have more statistical power. Finally, with a broad research purpose of finding new technologies to support military commanders on a mission, a holistic approach is particularly suited. A weakness of this approach is the possibility that the results are not fully explicable because of large manipulations and many interdependencies (because the real world situation is simulated). However, this type of research offers the possibility to sketch the problem space and makes the application of the technology, in this case decision support during combat situations, easily understandable and imaginable. Also, it provides many open ends which facilitates new research ideas.
The results show two major points: stress is more likely to be a positive than a negative predictor of performance, and collaboration between a military commander and decision support system yield worse results than decision making by the decision support model individually. The discussion of the results is split based on these two findings.

7.1 Relationship between stress and performance

Previous research shows that a high neuroticism score is related to the likelihood to show emotional or physical reactions to a stressful event. The Eysenck Personality Questionnaire (EPQ) used in this experiment, measures the neuroticism construct with good reliability. Previous studies show that heart rate variability is a good measure of the amount of stress a person is experiencing. In this study, no relationship is found between the neuroticism score and the measured stress in this experiment. Because of time constraints and the absence of a comment in the EPQ manual about filling in the questionnaire for one personality trait, we chose to ask only the neuroticism questions. Retrospectively, this might have influenced the reliability of the test. Because the other items of the personality questionnaire were not asked, the context of the neuroticism items was absent and this could have led to an unbalanced questionnaire.
The measured stress (using heart rate variability) and the user reported stress were analyzed. As a measure for user reported stress, the SUD scale was used, because it is a fast and accurate way of estimating the stress a person is ex-

periencing (Kaplan & Smith, 1995). The values of measured stress and user reported stress are expected to correlate positively. This study found a correlation of $\tau = 0.069$, which can not be considered a positive trend. We used the SUD scale because it is widely used and validated for estimating the amount of stress experienced by a person. A possible explanation for the absence of a positive trend is the distinctiveness of the SUD scale for the range of stress that was experienced during this study. The levels on the SUD scale are developed to distinguish between totally no stress (experienced in a deep sleep) and the maximum anxiety, panic and fear you can possibly imagine. These two extremes, and also some adjacent scales, were not filled in by the users in this first person shooter task. As a result, the answers of the participants were clustered around the middle levels (3, 4 and 5) and maybe the short description of each of those levels did not provide enough detail to distinguish between them.

The positive trend on the relationship between the number of enemies a participant had to face and the measured stress confirms statements from previous studies that an increase in taskload elevates the experienced stress (Driskell & Salas, 1996). The fact that this relationship is not significant, is probably due to the small sample size of this experiment.

Concerning the relationship between stress and performance, the micro and macro effects were analyzed. During the session, the participants seemed more relaxed as the session progressed and the stress measurements confirm this observation (see figure 28). Also, the expected effects of stress (negative relationship with both number of enemies hit and accuracy) was seen as a trend at the first block played (when the measured stress was highest, see figure 30), but actually turned around as the session progressed. Across all blocks, the shooting accuracy of the participants was higher when more stress was measured, which might indicate positive stress after the initial stress of the first block played. Prior to the experiment, we expected the team member task to be stressful enough. Pilots confirmed that we could distinguish playing sessions from rest sessions using the stress monitor. However, we did not specifically test distinguishing low and high taskload using the stress values, which retrospectively should have been tested prior to the experiment. Whereas the prospected system will be used in the battlefield, we could not use such an environment for this experiment. Instead, the realistic serious battlefield game Virtual BattleSpace 2 (VBS2) was used. Major differences are the lack of physical effort needed for virtual gaming and the absence of combat stress. The combat stress is very specific, because it involves dealing with major threats like getting injured or fear of death. In this experiment, the combat stress was replaced by trying to induce severe stress by making the task difficult and using stress inducing gaming vests. By using the gaming vest, we expected to elevate the experienced stress during the game. Despite our efforts, the overall team member task might have turned out to be not stressful enough and aimed at the left part of the transition (between bored and moderate stress) instead of on the right part of the transition (between moderate and severe stress). This way, the performance measured over the entire session does not decrease when stress increases, because the stress is not severe enough to impair physical and cognitive processes.

Also, the stress measurement needs to be critically evaluated. As explained in section 3, the calibration of the stress monitor is critical for the stress measurement. The calibration can only be successful when the participant is confronted with both low and high stress situations. Retrospectively, the gaming environment in which the participants were forced to get familiar with the controls and gameplay, might have been too free and did not guarantee that all the participants were exposed to the same amount of high stress situations. When observing participants during the calibration session, we saw that some participants found good hiding places where they were not confronted with large numbers of enemies, while others were violently attacked in the open field. This might have caused irregularities in calibration, which had impact on all the stress measurements in the rest of the experiment. Despite our efforts to provide all participants with the same calibration situation, it is not guaranteed that they actually had the same exposure to stressful events. In future research, the calibration session could be more automated to increase the likelihood that all participants are going through the exact same procedure in this crucial phase. Finally, when generalizing this research to the military domain, the representativeness of the participants used in the experiment is a discussion issue. The system is targeted to be used in the military domain and will investigate stress values of soldiers and decision making skills of commanders. We have chosen to invite regular participants for both the soldier and commander role due to constraints of time and resources. Real soldiers probably react differently to stressful events than regular persons.

7.2 Support effectivity

The second part of the analysis concerns the decision support effectivity. Manipulating the difficulty of the commander task could have provided interesting insights or explanations on differences between support conditions. Maybe advice would be more effective when carrying out a hard task than (for example) only visualization, while with an easy task the visualization might be more appropriate. However, from the results section, it became apparent that the differences between the two difficulty levels could not be found in the data. This can be explained by the fact that the attack regions were expected to switch approximately 56 times per block in the hard condition, where this would be 19 in the easy condition. Unfortunately, the number of times the attack region was switched was not logged. Switching tasks only has effect when the number of enemies that were shot (and died) between those switches is sufficient. For example, when the attack region would change within 20 seconds, and no enemies were shot in that 20 seconds, this attack change can not be found in the data. From pilot experiments, we found that the number of switching between 56 and 19 would be appropriate. In the real experiment, the number of enemies that were shot was much less than in the pilot experiments, therefore making more switches without any effect. This could be because the participants in the pilot experiments were less serious about hiding from the enemies; they were hit more, but they also shot more enemies. When this experiment would be repeated, the easy condition should change the attack region even less times, and the hard condition around 20 times a block. The differences between the two conditions will be more apparent that way.
The analyis of the objective support effectivity shows that the model performed better than the advice supported commanders on team performance. All other relationships between support effectivity and conditions were not significant.
A possible explanation for the results is that the commander, despite the color codes, could be overwhelmed with information without knowing what to do with

it. People are not used to interpret stress information, and the participants had to find out themselves how reliable and usable that extra information actually was. This could have caused decreased performance in support effectivity.
Another explanation can be found in an often suggested cause when human and system perform worse when working together: the complacency effect. This effect concerns the uncritical reliance of the human on the proper function of an automated system without recognizing its limitations and the possiblities of automation failures (Bahner, Hüper, & Manzey, 2008). Although the complacency effect is strongest where a human and an automated system have to work together, this phenomenon can also be found in automatically generated advices. The complacency effect results in humans not seeking for confirmatory or disconfirmatory information, and directly following the automatically generated advice without cross-checking its validity against other available information. Although the complacency effect can explain the lower performance in the advice condition to some extent, it is likely that this effect is not the main cause. Complacency occurs most likely in interaction with automated systems that are perceived as highly and constantly reliable (Parasuraman et al., 2000), which is not the case considering the comments of the commanders about the reliability of the model (see table 11 and appendix E). On top of that, if the human commanders would be highly complacent in the advice condition, the support effectivity would be close to the support effectivity of the automatic condition. Because the advice and automatic condition differ significantly from each other, the influence of the complacency effect can be considered limited.
Finally, the difference on team performance is maybe due to the fact that the model had a built-in mechanism that sometimes sends soldiers to empty areas where no enemies were present to check the situation. A frequently heard complaint about the advice system was that the system adviced to send soldiers to a region where no soldier was present at the moment; this advice was labeled by the commanders as 'illogical'. The participants were maybe more reluctant to listen to that advice because they did not trust it, but that might have caused enemies to walk into the village through unmanned regions. Although the participants made many negative comments on the advice system, the majority of the participants chose the advice system as the system they would prefer to use. They reported that they prefer as much input as possible for the decision making process. Because the participants would probably not choose the advice system when it would provide totally irrelevant information, they must think that stress information of team members is important information in the decision making process of task allocation.
On the number of user switches, it was found that commanders in the basis condition changed their task allocation less often than commanders in the other conditions. The model in the automatic condition significantly made more switches than the commanders in the advice condition. The participants reported that in the advice condition, the arrows were quite overwhelming and that they wanted to be able to remove the arrows. This might indicate that the advice condition could not display the most relevant information for the user in a correct way, which can impair the performance of the user working with that system. The commanders could maybe not fully concentrate on the task and made less changes. However, making less changes in task allocation does not automatically mean that the support effectivity is low. The commanders in the basis condition made significantly less switches than all the other condi-

tions, but this did not result in less performance on support effectivity (except when comparing with the automatic condition). The low support effectivity of the advice condition can thus not be attributed to making less changes due to concentration loss.
In general, the unability to find sigificant differences on most of the analyses can be due to the small sample sizes: for the commander task, eleven results could be evaluated because of time constraints and some technical problems. The team member task had data of 18 participants to analyze, although the analyses on stress and performance were focused on comparisons between blocks, which resulted in a sufficient number of datapoints (49).
The discussion points provide recommendations for future research. The concluding section will elaborate more on this topic.

8 Conclusion

The aim of this research was to find out whether the decision making of a military team leader can improve when he would know more about the condition of the members of his team. The research question was formulated as:

> *Does decision support, based on an estimation of stress of team members, increase the performance of the team leader in allocating team members to a set of tasks?*

This research question is answered with an experiment in a military test environment. Different decision support types and the difficulty of the task were manipulated. The influence of these two indepedent variables were tested on two depedent variables: objective support effectivity, which consists of two types of team leader performance measures, and subjective support effectivity, which is a questionnaire filled in by the commander. External to these variables, an important assumption was that the stress of the team members is a negative predictor for the performance of the team members.
Based on previous research, it was expected that stress is a negative predictor of performance (hypothesis 1, see section 4). On the support types, it was hypothesized that visualizing information about stress values of team members would increase the quality of decision making compared to the control condition (no support), and advice on interpreting the stress values by a decision support model would yield the best quality of decision making. The model was also individually tested by including an automatic condition.
In total, 18 participants completed the experiment, from which 609 datapoints for the commander task and 2717 datapoints for the team member task were distilled. The analyses show that the measured stress (using heart rate variability) did not correlate with either the validated neuroticism scale from the Eysenck Personality Questionnaire (EPQ), or the self-reported stress measured on the SUD scale. Both micro and macro effects between stress and performance measures (number of times hit, number enemies shot, and shooting accuracy) were analyzed, which resulted in a positive correlation between stress and shooting accuracy and no correlation between the other two performance measures and stress. Because a negative predictor role was expected, hypothesis 1 was rejected. However, the analysis showed that in the first 30 minutes of the experiment, the stress of the participants was significantly higher than in the rest of the experiment. Analyses on the relationship between stress and performance in this first 30 minutes showed results that were more consistent with hypothesis 1, although these results were not significant. This finding might indicate that a more stressful task is needed to make stress predict performance negatively.
The objective support effectivity measures showed that the model in the automatic condition performs better than the human commanders in the advice condition. All other relationships between objective support effectivity and conditions were not significant. The subjective support effectivity evaluation provided an interesting perspective, because the commanders liked the advice support system more than the other systems. The users prefer a system that provides more information, because they reported that they think more information will yield a better result. Both support effectivity measures indicate that none of the hypotheses 2, 3 or 4 posed in section 4 can be confirmed.
This study was originally triggered by the idea to develop new tools for the mil-

itary domain to increase information superiority (competitive advantage) and efficiency. Because the decision support model is developed with the aim to serve more applications, follow-up research does not have to involve a virtual game task. Due to the flexibility of the algorithm, the number of tasks and team members can also be altered. There are roughly four fields were subsequent research could arise: psychology, military research, serious gaming and decision support systems.
In the field of psychology, this research contributed by providing an example of a real-time stress measurement which is used within the experiment. The real-time stress measurements can also be used in civil applications; there is software available for athletes that can estimate whether you should exercise or rest based on heart rate sensor input.
These kind of applications can also be found in the military research domain, where the measurement of stress could be deployed when soldiers have to perform difficult tasks (not necessarily combat, but also physically difficult missions). During these missions, the team leaders can keep an eye on the condition of their soldiers. With this information, the decision of deployment of particular soldiers in next missions can be more informed.
Because this experiment was conducted in a serious gaming environment, this research has provided some answers to questions about the relationship between stress and performance in games. In this serious gaming environment, a stressor vest and a highly skilled opponent were not enough to severly stress the participants. Maybe these effects will be apparent when a game is more frustrating or more immersive. Future research could provide answers to that.
Finally, the decision support system can be altered and improved by adding more functionalities. Some improvements were suggested by the participants of this experiment. First of all, when re-allocating the tasks, the system should provide a warning to the team members. Secondly, the model can be made more flexible by empowering the user to ignore advices or remove information from the screen. The whole system can also be changed into a learning system in which system and user really work together. The experiment involved a system that gave advice to the user, but gave the same advice over and over again, even if the user thought that advice was not useful. A more collaborative approach is possible in which the system and user give each other feedback and reach a decision together. The military perspective of this research approach did not allow for extensive thinking and feedback during the commander task, but for other tasks, a learning system might be an option.
In summary, the research question should be answered with 'no', because the performance of the team leader did not increase. It is not clear whether the decision support system would provide valuable input for the team leader. According to the subjective user input, the stress values of team members and advice of the system would be valuable input for decision making. This study revealed a discrepancy between the objective and the subjective results of support effectivity. A follow-up research could investigate this 'mismatch' and focus more on a substantiated explanation for this phenomenon.

References

Aldwin, C., & Werner, E. (2009). *Stress, coping, and development: An integrative perspective.* New York, USA: Guilford Publications.

Allport, G. (1961). *Pattern and growth in personality.* New York, USA: Holt, Rinehart and Winston.

Ammann, K., Bourdon, L., Buller, M., Florence, G., Freund, B., Gunga, H., et al. (2010, July). *Real-time physiological and psycho-physiological status monitoring* (Tech. Rep. No. RTO-TR-HFM-132). Research and Technology Institution (RTO) and North Atlantic Treaty Organisation (NATO).

Bahner, J., Hüper, A.-D., & Manzey, D. (2008). Misuse of automated decision aids: Complacency, automation bias and the impact of training experience. *International Journal of Human-Computer Studies*, *6*.

Barnes, M. (2003). *The human dimension of battlespace visualization: Research and design issues.* Aberdeen Proving Ground: Army Research Lab.

Berntson, G., & Cacioppo, J. (2004). *Heart rate variability: Stress and psychiatric conditions.* New York, USA: Blackwell/Futura.

Boyd, c. J. A. (1987). *A discourse on winning and losing.* (Unpublished briefing)

Carver, C. S., & Connor-Smith, J. (2010). Personality and coping. *Annu Rev Psychol*, *61*, 679-704.

Costa, P., & McCrae, R. (1992). *Revised neo personality inventory (neo pi-r) and neo five-factor inventory (neo-ffi).* Lutz, USA: Psychological Assessment Resources.

Delahaij, R., & Gaillard, A. W. K. (2008). Individual differences in performance under acute stress. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, *52*, 965–969.

Delaney, J. P., & Brodie, D. A. (2000). Effects of short-term psychological stress on the time and frequency domains of heart-rate variability. *Percept Mot Skills*, *91*(2), 515–524.

Diedrichsen, L. (2000). Command & control: Operational requirements and system implementation. *Information & Security*, *5*.

Dishman, R. K., Nakamura, Y., Garcia, M. E., Thompson, R. W., Dunn, A. L., & Blair, S. N. (2000). Heart rate variability, trait anxiety, and perceived stress among physically fit men and women. *International Journal of Psychophysiology*, *37*(2), 121–133.

Driskell, J., & Salas, E. (1996). *Stress and human performance.* Mahwah, USA: Lawrence Erlbaum Associates.

Friedman, B. H., Thayer, J. F., & Tyrrell, R. A. (1996). Spectral characteristics of heart period variability during cold face stress and shock avoidance in normal subjects. *Clinical autonomic research official journal of the Clinical Autonomic Research Society*, *6*(3), 147–152.

Gohm, C., Baumann, M., & Sniezek, J. (2001). Personality in extreme situations: Thinking (or not) under acute stress. *Journal of Research in Personality*, *35*(3), 388-399.

Hancock, P., & Warm, J. (1989). A dynamic model of stress and sustained attention. *Human Factors*, *13*(5), 519-537.

Hjortskov, N., Rissén, D., Blangsted, A. K., Fallentin, N., Lundberg, U., & Søgaard, K. (2004). The effect of mental stress on heart rate variability and blood pressure during computer work. *European Journal of Applied Physiology*, *92*(1-2), 84–89.

Kamphuis, W., Gaillard, A., & Vogelaar, A. (2011). The effects of physical threat on team processes during complex task performance. *Small Research Group.*

Kaplan, D., & Smith, T. (1995). A validity study of the subjective unit of discomfort (sud) score. *Measurement and Evaluation in Counseling and Development*, *27*.

Kavanagh, J. (2005). *Stress and performance: a review of the literature and its applicability to the military.* Santa Monica, California: Rand Corporation.

Kreibig, S. D. (2010). Autonomic nervous system activity in emotion: a review. *Biological Psychology*, *84*(3), 394-421.

Lieberman, H., Bathalon, G., Falco, C., Morgan 3rd, C., Niro, P., & Tharion, W. (2005). The fog of war: decrements in cognitive performance and mood associated with combat-like stress. *Aviat Space Environ Med*, *76*(7 Suppl), C7-14.

Lucini, D., Di Fede, G., Parati, G., & Pagani, M. (2005). Impact of chronic psychosocial stress on autonomic cardiovascular regulation in otherwise healthy subjects. *Hypertension*, *46*(5), 1201-1206.

Macklin, C., & Dudfield, H. (2001). Campaign assessment visualisation techniques for command teams. In *Human interfaces in control rooms, cockpits and command centres.* Stevenage, United Kingdom: Institution of Engineering and Technology.

Ntuen, C. A., Park, E. H., McBribe, M. E., Marshak, W., Mountjoy, D. N., & Yarbrough, P. L. (1998). Collective asset display for commander's decision making. In *Systems man and cybernetics 1998 1998 ieee international conference on* (Vol. 2, p. 1302-1305).

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, *30*, 286-297.

Power, D. J., & Sharda, R. (2009). Decision support systems. In *Handbook of automation* (p. 1539-1548). Berlin, Germany: Springer Verlag.

Prinzel, L. J., Freeman, F. G., Scerbo, M. W., Mikulka, P. J., & Pope, A. T. (2000). A closed-loop system for examining psychophysiological measures for adaptive task allocation. *The International journal of aviation psychology*, *10*(4), 393–410.

Rhee, C., & Rao, H. R. (1990). Evaluation of decision support systems: The three-faceted approach to dss evaluation. *Decision Support Systems.*

Rovira, E., McGarry, K., & Parasuraman, R. (2007). Effects of imperfect automation on decision making in a simulated command and control task. *Human Factors*, *49*, 76-87.

Salmon, P. M., Walker, G. H., Ladva, D., Stanton, N. A., Jenkins, D. P., & Rafferty, L. (2007). Measuring situation awareness in command and control: comparison of methods study. In *Proceedings of the 14th european conference on cognitive ergonomics: invent! explore!* (pp. 27–34). New York, USA: ACM.

Sanderman, R., Arrindeel, W., Ranchor, A., Eysenck, H., & Eysenck, S. (1995). *Het meten van persoonlijkheidskenmerken met de eysenck personality questionnaire: een handleiding.* Groningen, Netherlands: Noordelijk Centrum voor Gezondheidsvraagstukken.

Sauter, V. (2010). *Decision support systems for business intelligence.* New York, USA: John Wiley & Sons.

Schneider, T. (2004). The role of neuroticism on psychological and physiological stress responses. *Journal of Experimental Social Psychology*, *40*.

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems*, *33*(2), 111–126.

Simon, H., & Zieve, D. (2009). *Stress - the body's response.* University of Maryland Medical Center (UMMC).

Staal, M. A. (2004). *Stress, cognition, and human performance: A literature review and conceptual framework.* NASA Johnson Space Center, Houston, USA: Ames Research Center, NASA/TM-2004-212824.

Yerkes, R., & Dodson, J. (1908). The relation of strenght of stimulus to rapidity of habit formation. *Journal of Comparative Neurology and Psychology*, *18*, 459-482.

Zhu, B., & Chen, H. (2008). Information visualization for decision support. In *Handbook on decision support systems 2* (p. 699-722). Berlin, Germany: Springer Verlag.

A Code

Listing 3: Find best bitarray

```
private double punishment = x;

private BitArray FindBestArray()
{
      //The best fitting solution is returned by the bestArray
          BitArray
      BitArray bestArray = new BitArray(s * t);
      //List of BitArrays that have exceeded the threshold
      List<BitArray> listBetterArrays = new List<BitArray>();
      //Score of the difference of the current BitArray
      double listPossibilityDifference = 0;
      //Variable to calculate the totalScore and assess which
          array has the lowest score (which is best)
      double totalScore = 0;
      //CalculateArrayDifference of the currentArray
      double differenceCurrentArray =
          CalculateArrayDifference(currentDistribution);

      foreach (BitArray item in legal)
      {
            //Calculate the difference value of each item in
                the legal list
            listPossibilityDifference =
                CalculateArrayDifference(item);
            //Check if any item (BitArray) from the legal list
                has a better (=lower) score than the current
                distribution. If an item is, add to
                listBetterArrays
            if (differenceCurrentArray -
                listPossibilityDifference > 0)
            {
                  listBetterArrays.Add(item);
            }
      }

      //Calculate which item in listBetterArray is best
          without too many mutations in distribution
      double previousScore = 99999;
      if (listBetterArrays.Count == 0)
      {
            return currentDistribution;
      }
      else
      {
         listBetterArrays.Add(currentdistribution);
```

```
            foreach (BitArray array in listBetterArrays)
            {
                  totalScore = CalculateArrayDifference(array);
                  totalScore += NumberOfMutations(array) *
                      punishment;
                  if (totalScore < previousScore)
                  {
                        previousScore = totalScore;
                        bestArray = array;
                        smallestDifference = totalScore;
                   }
                  totalScore = 0;
            }
            return bestArray;
      }
}
```

Listing 4: Number of mutations calculation

```
//Method that calculates the number of mutations needed for a
    particular solution by comparing it with the current
    distribution
private int NumberOfMutations(BitArray bits)
{
      int differences = 0;
      for (int i = 0; i < bits.Length; i++)
      {
            if (!bits[i].Equals(currentDistribution[i]))
            {
                  differences++;
            }
      }
      //Divide by 2, because one mutation always requires two
          differences found in the previous calculation
      return
          (System.Convert.ToInt32(Convert.ToDouble(differences)
          / System.Convert.ToDouble(2)));
}
```

B Software and hardware used

This appendix section provides some extra information about the hardware and software used in this experiment.

B.1 Software

B.1.1 Virtual Battlespace 2

Virtual Battlespace (VBS) 2 is a runtime environment that is developed by Bohemia Interactive. It offers an immersive virtual world to assist with learning and practicing military-related tasks in either an individual or collective manner. Developers can easily create new levels or adjust levels to specific needs.
The game is designed to simulate a real training, so all aspects of modern battlespace ((unmanned) vehicles, vessels, aircrafts) are included.

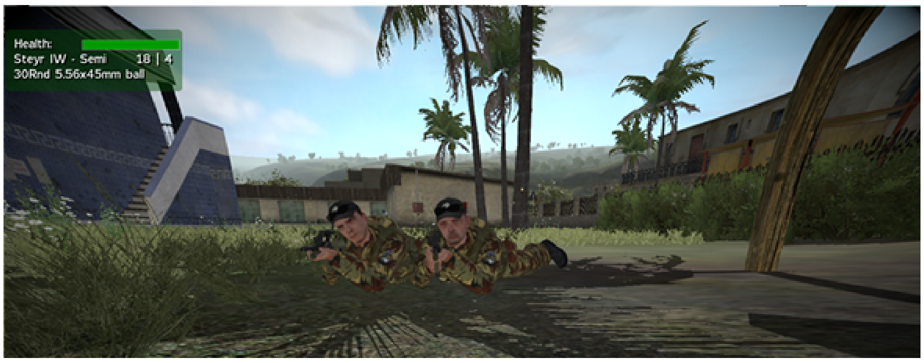
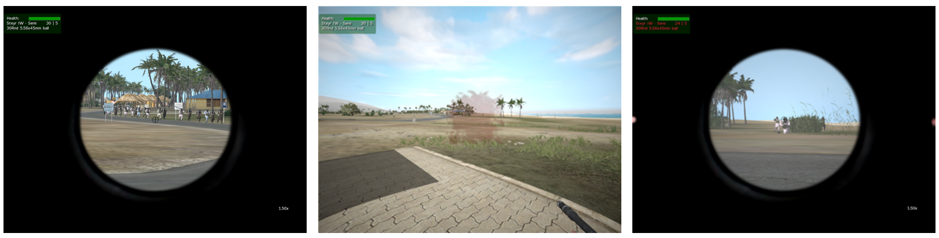


Figure 35: Soldiers in VBS2

The variety of actions for soldiers is more elaborated than in commercial first person shooter games, in order to enhance the possibilities of training. Also, the reactions of the virtual soldiers are more realistic than in commercial first person shooter games. When the soldier gets shot, he is not able to fire right back and he is also more likely to die than in commercial games.



(a) Enemies spotted in zoomed mode (b) Shot by an enemy (c) Zooming to shoot

Figure 36: VBS 2 screenshots on user actions

For more information about this software package, see the official website[1].

[1]Bohemia Interactive VBS2 website:`http://armory.bisimulations.com/products`

B.1.2 Trex

Trex stands for Tagging-based Real-time Exhibitor. It can be seen as a query engine and is characterized by the following properties:

- Semantic: it can be used to query RDF data which may be distributed over multiple networked computers
- Real time: it can be used to query dynamic data. The result of the query can change, when the underlying data pool changes
- Visual: results are presented visually

The goal of Trex is to enable humans and computers to do collaborative tagging. Data is produced in large quantaties, but is made searchable through the use of meta data. To enable humans to consume large quantities of data, query results are visualized. It is designed to visually explore and alter large quantities of data in a semantic web repository. Trex is designed to connect to a Sesame repository to store and load data from. The Sesame Repository is a Java applet, which will run on an Apache Tomcat server. In order to use Trex, these programs need to be installed additionally. Trexcore is the underlying framework for Trex.

B.2 Hardware

B.2.1 Mobi8

The Mobi system is developed by TMSI (Twente Medical Systems International B.V.). It is a multichannel system that is designed to register various (electro)physiological signals in domains like kinesiology, rehabilitation, ergonomics, motion analysis, sports and telemedicine. The measurement possibilities of the Mobi system includes EMG, EEG, ECG, positions, forces, movements, temperature, respiration and galvanic skin response. It can measure these signals both ambulatory or stationary. With stationary use, the data can be send directly to a computer using BlueTooth telemetrics. For ambulatory use, the memory card within the Mobi can save data up till a maximum of 1 GigaByte. Two AA batteries provide the Mobi with enough power to register for a maximum of 24 hours. The Mobi system is displayed in figure 37.

Figure 37: Mobi 8 system from TMSI website[2]

For more information about the TMSI Mobi system, see the official website[2].

-overview

[2]TMSI Mobi official website:http://www.tmsi.com/?language=nl&id=5

B.2.2 TN Gaming vest

The TN Gaming vest is developed by TN Games and is called the 3RD Space Vest. Its goal is to create a more immersive gaming experience by adding a third dimension to gaming: physical presence. The vest is equipped with air-pressurized spots, which can poke the user that wears the vest when he is shot. This air pressure is generated by an air compressor, which is connected to a wall outlet (power source) and a connector leading into a tube that connects the air compressor to the vest. The vest is attached to the computer by a USB cable to send a signal to the compressor when the user is hit in the game.
The vest is closed with a zipper and fork-clips and can be adjusted with straps on shoulders and abdomen to create the best fitting. Figure 38 shows the TN 3RD Space Vest and the compressor.

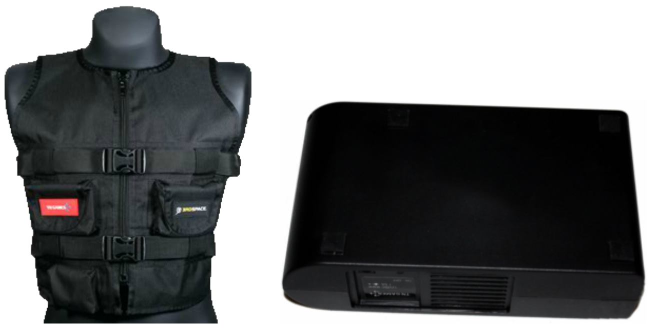

(a) TN 3RD Space Vest (b) Compressor

Figure 38: TN Gaming vest materials[3]

In the vest, eight 'active zones' are present, four on the front and four on the back. The direction and force of bullet fire, explosions, and environmental effects can be synced with the number and position of the active zones giving a signal. For example, when the user is shot by a pistol, only one active zone gives a signal in the corresponding place, but with a big explosion, all eight active zones are activated.
For more information about the TN 3RD Space Vest, see the official website[3].

[3]TN 3RD Space Vest official website: `http://tngames.com/products`

C Questionnaires (Dutch)

In this experiment, four questionnaires were used:

1. General questionnaire to collect demographic data, which includes filling in the subjective units of distress questionnaire before the experiment started and questions about the gaming vest, which were answered after the experiment
2. Neuroticism questionnaire (short Eysenck Personality Questionnaire)
3. Subjective units of distress questionnaire, which the participants filled in between the blocks
4. Commander questionnaire, which the commander filled in after the end of the block

These questionnaires are included in the aforementioned sequence.

Intake vragenlijst **PPN:**

Geslacht: m/v

Leeftijd:

Hoogst afgeronde opleiding: lagere school / lbo / mbo / mavo / havo / vwo / hbo / universiteit

Geeft hieronder aan of de volgende factoren bij u een rol spelen, kruis de bolletjes aan en vul eventueel een waarde (schatting) in op de stippellijnen.

	Ja	**Nee**
Rookt u?	**O**	**O**
Zo ja, hoeveel sigaretten gemiddeld per dag:............		
Drinkt u koffie?	**O**	**O**
Zo ja, hoeveel kopjes gemiddeld per dag:		
Drinkt u alcoholische dranken?	**O**	**O**
Zo ja, hoeveel glazen gemiddeld per week:............		
Vindt u dat u gisternacht goed heeft geslapen?	**O**	**O**
Sport u?	**O**	**O**
Zo ja, hoeveel uur gemiddeld per week:		
Bent u naar TNO komen fietsen?	**O**	**O**
Heeft u ooit (of momenteel) hartklachten gehad?	**O**	**O**
Neemt u medicijnen?	**O**	**O**
Speelt u computer spelletjes?	**O**	**O**
Zo ja, hoeveel uur gemiddeld per week:		
Zo ja, hoe veel tijd daarvan speelt u pc-spellen:……….		
En, hoeveel tijd daarvan speelt u een first person shooters:		
Heeft u wel eens een TN GamingVest aangehad?	**O**	**O**

Subjective Units of Distress **PPN:**

Deze subjectieve stress schaal, is een handige meetmethode om aan te geven hoe hoog de discomfort op een bepaald moment is. De schaal heeft 11 punten (0 t en met 10). Geeft met een kruisje op de lijn aan, hoe hoog uw ervaren stress momenteel aanvoelt:

0: Complete ontspanning. Diepe slaap, totaal geen stress.

1: Wakker, maar heel erg ontspannen; in slaap vallend. Uw gedachten dwalen af net zoals vlak voor u in slaap valt.

2: Ontspannend aan het strand, ontspannen thuis voor een warm vuurtje op een winterse dag, of een vreedzame wandeling door het bos.

3: De hoeveelheid stress en spanning die nodig is om je gedachten niet te laten dwalen, om je hoofd omhoog te houden etc. Deze spanning is niet onplezierig; het is ‘normaal’.

4: Milde stress zoals milde lichamelijke spanning, lichte zorgen of angst. Een beetje onplezierig maar makkelijk te verdragen.

5: Milde tot gemiddelde stress. Duidelijk onplezierig maar te weinig om veel lichamelijke symptomen te produceren.

6: Gemiddelde stress. Erg onplezierige gevoelens van angst, boosheid, zorgen, vrees en/of duidelijke lichamelijke spanning zoals hoofdpijn, buikklachten. Duidelijke onplezierige maar draaglijke sensaties; u kunt nog helder nadenken. Meestal omschreven als een slechte dag, maar het werken, rijden, converseren wordt niet belemmerd.

7: Gemiddeld hoge stress die het concentreren moeilijk maakt. Tamelijk intense lichamelijke stress.

8: Hoge stress. Sterke gevoelens van angst, zorgen, vrees en/of lichamelijke spanning. Deze gevoelens zijn niet lang vol te houden. Nadenken en het oplossend vermogen zijn verzwakt. Lichamelijke stress is wezenlijk. Werken, rijden, converseren etc. wordt bemoeilijkt.

9: Hoge tot extreme stress. Denken is wezenlijk belemmerd.

10: Extreme stress, gevoelens van paniek of angst. Extreme lichamelijke spanning. Het maximale gevoel van angst, paniek en vrees dat je mogelijk acht.

0 1 2 3 4 5 6 7 8 9 10

Deze vragen kunt u pas na het afronden van het experiment in vullen. Geef het formulier voorlopig weer terug aan een van de proefleiders.

Vragen over de Gaming Vest **PPN*:***

Wat voor gevoel gaf het vest u als deze afging?				
Aangenaam				Onaangenaam
O	O	O	O	O

Hoe stressvol hebt u het vest ervaren?						
Helemaal niet stressvol						Heel erg stressvol
O	O	O	O	O	O	O

Kunt u in uw eigen woorden het gevoel dat het vest u gaf beschrijven:

Kunt u hieronder aangeven of u het idee had, dat het vest uw beleving van het spel vergrootte?

Heeft u verschillen gemerkt in de werking van het vest? Reageerde het vest de ene keer anders dan de andere keer?

Vragenlijst vooraf Proefpersoonnummer: _______

Wil je iedere vraag beantwoorden door 'ja' of 'nee' te omcirkelen? Er zijn geen goede of foute antwoorden en geen strikvragen. Het is niet nodig dat je erg lang over de vragen nadenkt.

Wil je erop letten alle vragen te beantwoorden?

1.	Gaat je stemming dikwijls op en neer?	ja	nee
2.	Voel je je wel eens 'gewoon miserabel' zonder dat daar reden voor is?	ja	nee
3.	Raak je snel geïrriteerd?	ja	nee
4.	Ben je nogal gauw in je gevoelens gekwetst?	ja	nee
5.	Komt het nogal eens voor dat je schoon genoeg hebt van de dingen?	ja	nee
6.	Vind je jezelf een zenuwachtig (nerveus, gespannen) iemand?	ja	nee
7.	Vind je jezelf een piekeraar (tobber)?	ja	nee
8.	Vind je jezelf een gespannen persoon?	ja	nee
9.	Als je in een pijnlijke situatie bent geweest, zit dat je dan nog lang dwars?	ja	nee
10.	Lijd je aan nervositeit?	ja	nee
11.	Voel je je vaak eenzaam?	ja	nee
12.	Word je vaak gekweld door schuldgevoelens?	ja	nee

Subjective Units of Distress **PPN:**

Deze subjectieve stress schaal, is een handige meetmethode om a ante geven hoe hoog de discomfort op een bepaald moment is. De schaal heeft 11 punten (0 t en met 10).

0: Complete ontspanning. Diepe slaap, totaal geen stress.

1: Wakker, maar heel erg ontspannen; in slaap vallend. Uw gedachten dwalen af net zoals vlak voor u in slaap valt.

2: Ontspannend aan het strand, ontspannen thuis voor een warm vuurtje op een winterse dag, of een vreedzame wandeling door het bos.

3: De hoeveelheid stress en spanning die nodig is om je gedachten niet te laten dwalen, om je hoofd omhoog te houden etc. Deze spanning is niet onplezierig; het is ‘normaal’.

4: Milde stress zoals milde lichamelijke spanning, lichte zorgen of angst. Een beetje onplezierig maar makkelijk te verdragen.

5: Milde tot gemiddelde stress. Duidelijk onplezierig maar te weinig om veel lichamelijke symptomen te produceren.

6: Gemiddelde stress. Erg onplezierige gevoelens van angst, boosheid, zorgen, vrees en/of duidelijke lichamelijke spanning zoals hoofdpijn, buikklachten. Duidelijke onplezierige maar draaglijke sensaties; u kunt nog helder nadenken. Meestal omschreven als een slechte dag, maar het werken, rijden, converseren wordt niet belemmerd.

7: Gemiddeld hoge stress die het concentreren moeilijk maakt. Tamelijk intense lichamelijke stress.

8: Hoge stress. Sterke gevoelens van angst, zorgen, vrees en/of lichamelijke spanning. Deze gevoelens zijn niet lang vol te houden. Nadenken en het oplossend vermogen zijn verzwakt. Lichamelijke stress is wezenlijk. Werken, rijden, converseren etc. wordt bemoeilijkt.

9: Hoge tot extreme stress. Denken is wezenlijk belemmerd.

10: Extreme stress, gevoelens van paniek of angst. Extreme lichamelijke spanning. Het maximale gevoel van angst, paniek en vrees dat je mogelijk acht.

PPN:

Geef na elk blok, met een kruisje op de lijn aan, welke score (Subjective Units of Distress) op u van toepassing is. De proefleiders zullen aangeven wanneer je weer een kruisje mag zetten.

BLOK:

1

0 1 2 3 4 5 6 7 8 9 10

2

0 1 2 3 4 5 6 7 8 9 10

3

0 1 2 3 4 5 6 7 8 9 10

4

0 1 2 3 4 5 6 7 8 9 10

5

0 1 2 3 4 5 6 7 8 9 10

6

0 1 2 3 4 5 6 7 8 9 10

Vragenlijst commandant Proefpersoonnummer: _______

1. Basis ondersteuning

Deze vragen gaan over het type ondersteuning waarbij je alleen witte bolletjes op het scherm zag en de verdeling moest maken op basis van mondelinge communicatie met de soldaten.

De eerste acht vragen gaan over de interface (uiterlijk) van het systeem. Het gaat hierbij dus niet om of het systeem bijvoorbeeld een goed advies geeft, maar of je gemakkelijk met het systeem kon omgaan en of het deed wat je verwachtte als je een bepaalde actie (bijv. klikken met de muis) deed.

Vragen over besturing en interface	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem is gemakkelijk te gebruiken	0	0	0	0	0
2. Het is leuk om het systeem te gebruiken	0	0	0	0	0
3. Het systeem werkt niet zoals ik wil dat het werkt	0	0	0	0	0
4. Het is moeilijk om het systeem te <u>leren</u> gebruiken	0	0	0	0	0
5. Het systeem doet wat ik wil in zo min mogelijk stappen	0	0	0	0	0
6. Het is moeilijk om te onthouden hoe ik het systeem moet gebruiken	0	0	0	0	0
7. De interface ziet er mooi uit	0	0	0	0	0

8. Andere opmerkingen over de besturing of interface kun je hieronder kwijt (optioneel)

____

____

____

De volgende vragen gaan over het gebruik van het systeem en de informatievoorziening.

Vragen over de ondersteuning	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem gaf correcte informatie	0	0	0	0	0
2. Het gedrag van het systeem was in alle situaties hetzelfde als wat ik ervan verwachtte	0	0	0	0	0
3. Communiceren met soldaten is een goede manier om beslissingen voor allocatie op te baseren	0	0	0	0	0
4. Ik heb optimaal gepresteerd met dit systeem	0	0	0	0	0

5. Andere opmerkingen over de ondersteuning kun je hieronder kwijt (optioneel)

____

____

____

2. *Visualisatie ondersteuning*

Deze vragen gaan over het type ondersteuning waarbij je alleen gekleurde bolletjes op het scherm zag en geen pijlen, en waarbij het systeem niet de beslissingen nam.

De eerste acht vragen gaan over de interface (uiterlijk) van het systeem. Het gaat hierbij dus niet om of het systeem bijvoorbeeld een goed advies geeft, maar of je gemakkelijk met het systeem kon omgaan en of het deed wat je verwachtte als je een bepaalde actie (bijv. klikken met de muis) deed.

Vragen over besturing en interface	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem is gemakkelijk te gebruiken	0	0	0	0	0
2. Het is leuk om het systeem te gebruiken	0	0	0	0	0
3. Het systeem werkt niet zoals ik wil dat het werkt	0	0	0	0	0
4. Het is moeilijk om het systeem te leren gebruiken	0	0	0	0	0
5. Het systeem doet wat ik wil in zo min mogelijk stappen	0	0	0	0	0
6. Het is moeilijk om te onthouden hoe ik het systeem moet gebruiken	0	0	0	0	0
7. De interface ziet er mooi uit	0	0	0	0	0

8. Andere opmerkingen over de besturing of interface kun je hieronder kwijt (optioneel)

____

____

____

De volgende vragen gaan over het gebruik van het systeem en de informatievoorziening.

Vragen over de ondersteuning	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem gaf correcte informatie	0	0	0	0	0
2. Het gedrag van het systeem was in alle situaties hetzelfde als wat ik ervan verwachtte	0	0	0	0	0
3. Communiceren met soldaten is een goede manier om beslissingen voor allocatie op te baseren	0	0	0	0	0
4. De stressinformatie is goede informatie om beslissingen voor allocatie op te baseren	0	0	0	0	0
5. De stressinformatie kwam vrijwel altijd overeen met de informatie die de persoon via de headset gaf	0	0	0	0	0
6. Ik vertrouw de stressinformatie meer dan de informatie die de persoon via de headset gaf	0	0	0	0	0
7. Ik heb optimaal gepresteerd met dit systeem	0	0	0	0	0

8. Andere opmerkingen over de ondersteuning kun je hieronder kwijt (optioneel)

____

____

____

3. Advies ondersteuning PP nr: _______

Deze vragen gaan over het type ondersteuning waarbij je pijlen op het scherm zag.

De eerste acht vragen gaan over de interface (uiterlijk) van het systeem. Het gaat hierbij dus niet om of het systeem bijvoorbeeld een goed advies geeft, maar of je gemakkelijk met het systeem kon omgaan en of het deed wat je verwachtte als je een bepaalde actie (bijv. klikken met de muis) deed.

Vragen over besturing en interface	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem is gemakkelijk te gebruiken	0	0	0	0	0
2. Het is leuk om het systeem te gebruiken	0	0	0	0	0
3. Het systeem werkt niet zoals ik wil dat het werkt	0	0	0	0	0
4. Het is moeilijk om het systeem te leren gebruiken	0	0	0	0	0
5. Het systeem doet wat ik wil in zo min mogelijk stappen	0	0	0	0	0
6. Het is moeilijk om te onthouden hoe ik het systeem moet gebruiken	0	0	0	0	0
7. De interface ziet er mooi uit	0	0	0	0	0

8. Andere opmerkingen over de besturing of interface kun je hieronder kwijt (optioneel)

____

____

____

De volgende vragen gaan over het gebruik van het systeem en de informatievoorziening.

Vragen over de ondersteuning	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem gaf correcte informatie	0	0	0	0	0
2. Communiceren met soldaten is een goede manier om beslissingen voor allocatie op te baseren	0	0	0	0	0
3. De stressinformatie is goede informatie om beslissingen voor allocatie op te baseren	0	0	0	0	0
4. De stressinformatie kwam vrijwel altijd overeen met de informatie die de persoon via de headset gaf	0	0	0	0	0
5. Ik vertrouw de stressinformatie meer dan de informatie die de persoon via de headset gaf	0	0	0	0	0
6. Het gedrag van het systeem was in alle situaties hetzelfde als wat ik ervan verwachtte	0	0	0	0	0
7. Ik heb het systeem op geen fout kunnen betrappen	0	0	0	0	0
8. Ik heb sterk geloof en vertrouwen dat het systeem zijn taak effectief uitvoert als ik dit niet zou kunnen controleren	0	0	0	0	0
9. Het systeem was goed in staat om zijn taak (advies geven) effectief uit te voeren	0	0	0	0	0
10. Ik heb optimaal gepresteerd met dit systeem	0	0	0	0	0

11. Andere opmerkingen over de ondersteuning kun je hieronder kwijt (optioneel)

____

____

____

4. Automatische ondersteuning

Deze vragen gaan over het type ondersteuning waarbij het systeem de beslissingen nam.

Bij de volgende vragen ga je het automatische systeem beoordelen.

Vragen over de ondersteuning	Erg mee oneens	Mee oneens	Neutraal	Mee eens	Erg mee eens
1. Het systeem was goed in staat om zijn taak (allocatie van soldaten) effectief uit te voeren	0	0	0	0	0
2. Het gedrag van het systeem was in alle situaties hetzelfde als wat ik ervan verwachtte	0	0	0	0	0
3. Ik heb het systeem op geen fout kunnen betrappen	0	0	0	0	0
4. Ik heb sterk geloof en vertrouwen dat het systeem zijn taak effectief uitvoert als ik dit niet zou kunnen controleren	0	0	0	0	0
5. Het systeem heeft optimaal gepresteerd	0	0	0	0	0

6. Op welke manier(en) zou de prestatie van het automatische systeem verbeterd kunnen worden?

____

____

____

____

____

7. Andere opmerkingen over de ondersteuning kun je hieronder kwijt (optioneel)

____

____

____

5. Eindvraag

Als je de commandant taak nog één keer zo goed mogelijk uit zou moeten voeren, welk type ondersteuning zou je dan kiezen?

0 De baseline ondersteuning

0 De visualisatie ondersteuning

0 De advies ondersteuning

0 De automatische ondersteuning

En (in het kort) waarom?

____

____

____

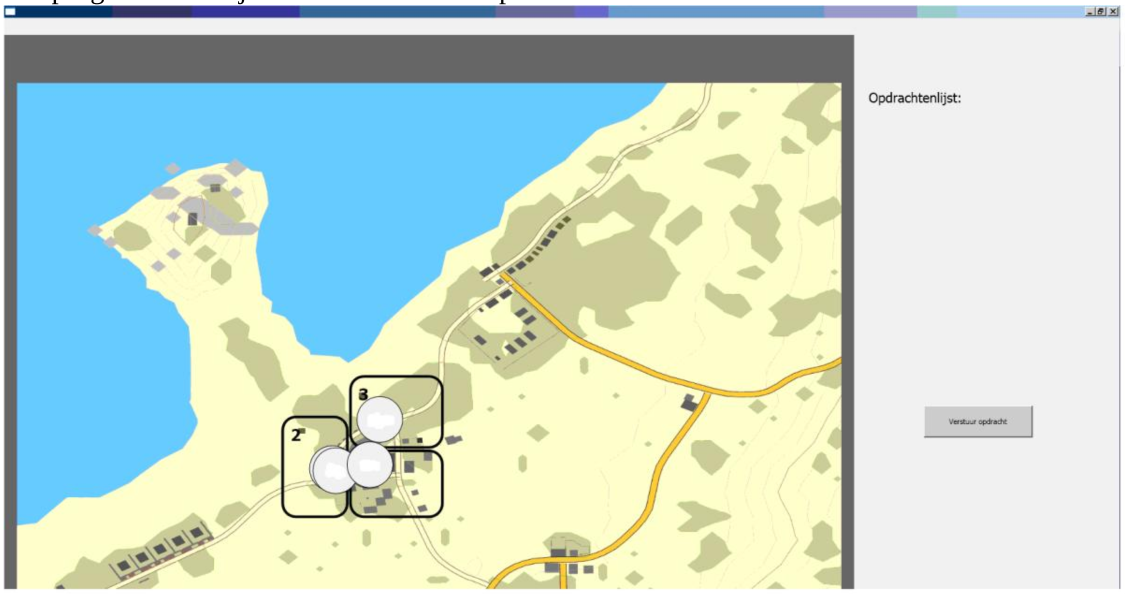
D Task Description (Dutch)

The task description includes the commander documentation, soldier documentation, and general information about the experiment. All these texts were printed and the participants red them before the experiment started.

Commandant documentatie

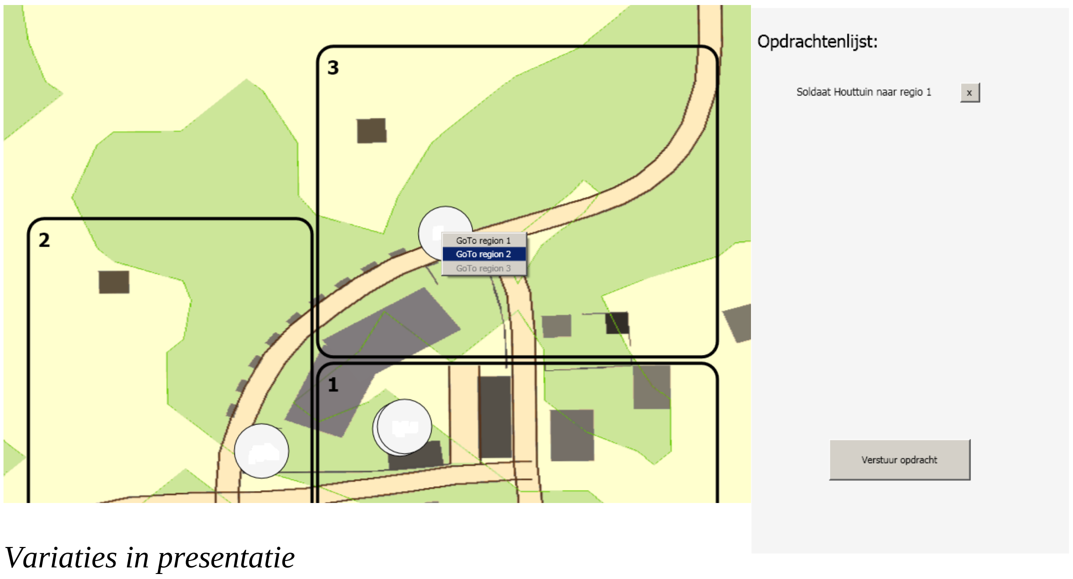
Taak
Als commandant heb je de taak om de vijf manschappen op de <u>beste manier te verdelen</u> over het aangevallen dorpje. Dit kan door middel van een kaartje van de omgeving (weergegeven) en besturing met de muis (combinatie van links- en rechtsklikken). Normaal gesproken neem je deze beslissing op basis van gezond verstand en communicatie met je soldaten, maar nu is er een nieuw systeem ontwikkeld dat kan helpen bij de besluitvorming. Als commandant moet je dus – naast je taak uitvoeren – ook beoordelen in hoeverre dit systeem bruikbare informatie geeft om een betere verdeling van soldaten te maken.

Interface interactie
Het programma wat je ziet bestaat uit twee panelen:



Aan de linkerkant een kaart van het gebied met vijf rondjes (deze representeren de soldaten), 3 afgeronde vierkanten (deze representeren drie regio's). Binnen de kaart kun je in- en uitzoomen met het scrollwiel van je muis.
Aan de rechterkant staat een opdrachtenlijst met onderaan een button 'Verstuur opdracht'.
Het is de bedoeling dat je als commandant in de gaten houdt waar je manschappen lopen en op basis van communicatie en informatie van het systeem ze een nieuwe positie geeft als je de huidige verdeling van de soldaten niet goed vindt. Het geven van een nieuwe positie werkt als volgt:
Klik met de rechtermuisknop op het cirkeltje van de soldaat die je wilt verplaatsen. Hier zie je twee opties (je kunt hem niet verplaatsen naar de regio waar de soldaat nu in staat). Zodra je dit gedaan hebt, komt in het paneel rechts jouw opdracht te staan. Deze opdrachtenlijst is een soort wachtrij; de opdracht wordt nog niet uitgevoerd. Als je meerdere opdrachten tegelijkertijd wilt uitvoeren, kun je ze onder elkaar verzamelen in de lijst. Je kunt ze ook altijd nog wegklikken met het kruisje naast de opdracht. Omdat een soldaat maar naar één regio kan gaan, wordt alleen de laatste opdracht onthouden. Als je bijvoorbeeld 'Soldaat 1 naar regio 2' in de opdrachtenlijst zet, en daarna 'Soldaat 1 naar regio 1', zal de opdracht niet erbij komen, maar veranderen in de laatstgenoemde. Er kunnen dus maximaal vijf opdrachten in de lijst komen. Als je zeker bent van

je lijst met opdrachten, kun je deze versturen met ‘Verstuur opdracht’. Hierna verdwijnen de opdrachten uit de lijst en worden de soldaten verplaatst naar hun nieuwe regio. Na enkele seconden is dit ook te zien op de kaart.



Variaties in presentatie
In totaal krijg je vier verschillende type interfaces in willekeurige volgorde. Deze wisselen elkaar automatisch af. Er is altijd een kaartje te zien, maar er zijn ook een aantal verschillen.

1. “Basis”: je ziet witte bolletjes. Je krijgt informatie over de positie van de soldaten, maar geen informatie over hun stress status
2. “Passief”: je ziet gekleurde bolletjes. Je krijgt hiermee informatie over zowel de positie als de stress status van de soldaten
3. “Advies”: je ziet gekleurde bolletjes en (soms) pijlen. Je krijgt hiermee informatie over zowel de positie als de stress status van de soldaten, en een model berekent steeds welke verdeling ideaal zou zijn. Als zijn ideale verdeling verschilt met de jouw verdeling, geeft hij advies wat hij zou veranderen door middel van rode pijlen
4. “Automatisch”: je ziet gekleurde bolletjes. Je krijgt hiermee informatie over zowel de positie als de stress status van de soldaten, maar je kunt geen aanpassingen doen aan de posities. Dit doet het systeem automatisch zelf.

Advies redenatie
Bij het presentatietype “Advies” krijg je suggesties van het systeem. Deze worden berekend door een model. De volgende punten worden meegewogen in deze berekening:

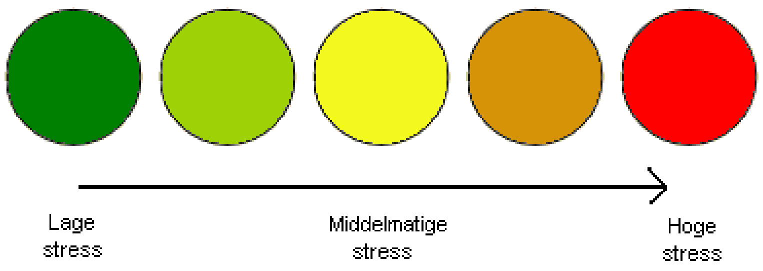
- Het model maakt een inschatting van de taakbelasting per regio op basis van de stressniveaus van de soldaten die in elke regio staan
- Het model maakt een inschatting van de ideale verdeling door de soldaten die het minst last hebben van stress in te zetten bij een hoge taakbelasting. Anders gezegd, het model volgt de redenering:
  “Hoge stress zorgt voor minder goed presteren in de toekomst”

Houd goed in de gaten dat deze redenering misschien niet altijd geldt. Je moet zelf beslissingen nemen en je bent als commandant verantwoordelijk, dus denk goed na welke informatie je gebruikt en of deze geldig is.

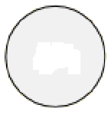
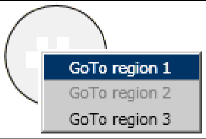
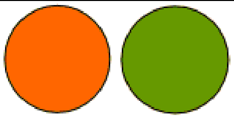
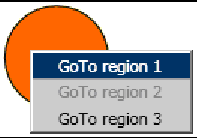
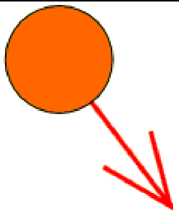
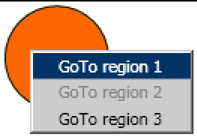
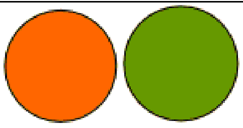
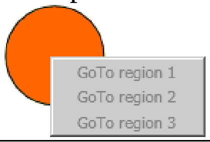
Jouw taken:

1. Verdelen van soldaten naar juiste posities zoals jij denkt dat de beste verdeling is. Je mag hierbij gebruik maken van:
   a. Informatie die via de interface gegeven wordt
   b. Informatie die je via communicatie van de soldaten krijgt
2. Communiceren naar betreffende soldaten, in het bijzonder als jij (of het systeem, in de automatische versie) een wijziging gaat doen (voorbereiden)
3. Als soldaten te ver uit de regio lopen, terugplaatsen in een regio

Betekenis van kleuren:



Basis besturing van de interface:

Presentatietype	*Soldaat representatie*	*Verplaatsen van soldaten*
1. "Basis"		
2. "Passief"		
3. "Advies"		 OF Klikken op de pijl
4. "Automatisch"		Je kunt de soldaat niet verplaatsen: 

Je krijgt de verschillende presentatietypen in willekeurige volgorde.

Soldaat documentatie

Taak
Je bent als soldaat in een hinderlaag terecht gekomen. Terwijl je met een beperkt groepje soldaten de omgeving aan het verkennen was, komen aanvallers plotseling het dorpje waar jullie toevallig zijn, belagen.Als soldaat heb je de taak om ervoor te zorgen dat aanvallers van buiten het dorp niet het dorp binnen gaan. Je doet dit door tegenstanders met kogels te raken.
Daarnaast heb je ook een persoonlijk doel: zo goed mogelijk presteren. Dat wil zeggen: <u>zo vaak mogelijk raak schieten en zo min mogelijk geraakt worden</u>.
Tijdens het spel kun je plotseling verplaatst worden; dit heeft te maken met de taak van de commandant. In dit geval moet je gewoon zo goed mogelijk verder spelen.

Interface interactie
Het spel bekijk je vanuit het perspectief van jouw eigen soldaat. De basis besturing werkt als volgt:

Toets	**Functie**
Toetsenbord	
ASWD	Lopen in bepaalderichting
Shift + ASWD	Rennen in bepaalderichting
X	Hurken (nog een keer drukken is rechtopstaan)
Z	Liggen (nog een keer drukken is rechtopstaan)
R	Herladen van wapen (linksboven zie je hoeveel kogels je nog hebt)
V	Veel inzoomen (nog een keer drukken is uitzoomen)
Muis	
Linker muisknop	Schieten
Rechtermuisknop	Eenbeetjeinzoomen
Scrollknop indrukken	Menu voor wapens / acties kiezen (nog een keer drukken laat het menu verdwijnen)
Scrollen	Door menu van wapens/acties lopen (alleen als je eerst hebt ingedrukt)

Explosieven
De tegenstander heeft ook explosieven bij zich om het dorpje te vernietigen. Als jullie (als team) teveel tegenstanders in de nabije omgeving van het dorpje laten, zal er een bom afgaan. Het voorkomen van die explosies is dus alleen mogelijk door de tegenstanders terug te dringen buiten het dorpje.

Prestatiemetingen
Jullie individuele- en teamprestatie zullen bijgehouden worden. Dit zal afhangen van een aantal factoren:
- Hoeveel explosies er geweest zijn
- Hoe vaak je iemand hebt geraakt
- Hoe vaak je zelf geraakt bent
- Je precisieratio (aantal keer dat je een tegenstander hebt geraakt gedeeld door het aantal schoten)

Het is de bedoeling om in elk spel zo goed mogelijk te presteren. Denk hieraan terwijl je aan het spelen bent.

Praktische informatie over het experiment

Experimentopbouw
Het experiment zal bestaan uit een aantal stappen:

0. Kalibratie
1. Blok 1
2. Blok 2
   Pauze
3. Blok 3
4. Blok 4
   Pauze
5. Blok 5
6. Blok 6
   Debriefing

Bij de kalibratie worden de stressmeter gekalibreerd en leer je omgaan met de controls binnen de game. Na de kalibratie start blok 1. Dit blok zal voor de commandant bestaan uit vier games van elk 7 minuten. Voor de soldaten zal het voelen als een lange sessie van 28 minuten. Na elk blok wordt één soldaat gewisseld met de commandant. De ingeroosterde pauzes zijn ongeveer 10 minuten. Na 6 blokken is het experiment klaar; je zult hierna nog enkele vragenlijsten moet invullen alvorens je naar huis kunt gaan.

Stressmeting
Tijdens het spelen van het spel als soldaat heb je altijd een stressmeter op een vinger van je rechterhand. Je mag in principe zelf weten op welke vinger van je rechterhand je dit wilt doen, maar waarschijnlijk is je pink (of anders je duim) hier het meest geschikt voor. Je moet tijdens het spel immers nog wel goed kunnen klikken met de muis. Het is belangrijk dat het draadje dat naar het vingerhoedje loopt, vastgeplakt zit op je hand. Daarnaast is het belangrijk om tijdens de metingen niet met je hand te zwaaien of je vinger zodanig te buigen dat het vingerhoedje (ver) open gaat.

Vest
Tijdens het spelen van het spel als soldaat heb je altijd een vest aan. Het is belangrijk dat je het vest goed strak aantrekt, bij voorkeur niet over een dik kledingstuk zoals een trui. In het vest zitten acht luchtkamertjes die vol geblazen kunnen worden met lucht en weer leeglopen. Er wordt in het experiment gevarieerd met het geven van deze prikkels, dus het vest zal niet altijd op dezelfde manier werken.
In het vest zitten twee kabels: één gaat naar je pc en één gaat naar de compressor. De (USB) kabel die naar de pc gaat zit niet echt 'vast' aan het vest. Let dus goed dat deze kabel 'vrij' loopt en niet op spanning staat, en dat deze goed in het vest blijft zitten.

Communicatie via headsets
Tijdens het experiment kan er gecommuniceerd worden via de headsets. In principe is het een open verbinding tussen alle deelnemers van het experiment. Om te voorkomen dat iedereen hierin door elkaar gaat praten, heeft de commandant altijd het initiatief. Hij of zij roept soldaten op als er iets gevraagd of doorgegeven moet worden. Als een soldaat iets wil zeggen tegen de commandant, roept hij/zij eerst de commandant op, waarna je een boodschap kunt doorgeven. Om het overzichtelijk te houden is het dus niet de bedoeling om onderling tussen soldaten te praten via de headsets. Je kunt wel steeds horen wat de andere soldaten zeggen als zij met de commandant praten, maar de commandant heeft altijd het initiatief in de gesprekken.

E Questionnaire: open question responses (Dutch)

The responses of the participants in the team leader role, sharing their opinions about the four conditions.

Antwoorden op open vragen commandant-vragenlijst

1 = Baseline
2 = Visualisatie
3 = Advies
4 = Automatisch

Gekozen voor:	Waarom?	Andere opmerkingen
3	Ik heb het idee dat je zelf de eindbeslissingen moet maken, maar wel zoveel mogelijk informatie krijgen	
3	Dan kan ik de informatie van de pc en van mezelf vergelijken	(over interface) Je kon alleen naar regio's verplaatsen, niet preciezer
2	Dan zou ik meer het gevoel van controle hebben	
3	Deze is het handigst	
3	Wel handig om de pijlen te zien, voor suggestie, maar wel zelf beslissen wat er met de soldaten gebeurt. Het stressniveau vond ik ook erg nuttig, omdat je dan via communicatie extra informatie aan de soldaat kan vragen over de situatie	(over interface) Ik vond het soms lastig om te zien welke PC welk bolletje was (omdat de bolletjes elkaar overlappen). Misschien een optie om het aangeklikte bolletje op de voorgrond te stellen? (over baseline) Het is wel handig om het stressniveau te zien. Hierdoor kan je vragen hoe de situatie daar is. Bij de witte bolletjes kan je dat niet zien, waardoor je 'random' vraagt hoe het gaat (over visualisatie) Handig om stressniveau te zien. Hierdoor kan je vragen hoe de situatie is (meestal wordt er dan veel geschoten) (over advies) Ik vond de pijlen eigenlijk niet erg nuttig. Uit de communicatie met de soldaten bleek dat er bv. op 2 plekken beschoten werd. De pijlen gaven aan de soldaten ergens anders heen te sturen, dus dat heb ik niet gedaan. Eerder extra soldaten naar de cruciale gebieden gestuurd, dan de soldaten naar de 'lege' (niet bedreigende) gebieden

3	Waarschijnlijk omdat je geen geheel overzicht hebt	(over advies) Pijltjes leken soms niet klikbaar
2	De toevoeging van stresslevel is nog net een toegevoegde waarde, maar het systeem (model) is nog niet betrouwbaar, dus het advies is van weinig waarde	(over automatisch) Lastig als er geen verdere input van de gebruiker is. Stresslevel is wellicht onbetrouwbaar / onnauwkeurig als enige te gebruiken
3	Automatisch was niet effectief, maar de informatie en advies waren wel nuttig	(over de interface) Duidelijk, effectief en intuïtief (over de baseline conditie) Communicatie was zeer belangrijk (over automatisch) Minder snel verplaatsen, communicatie toevoegen voordat iemand wordt verplaatst!
2	Beste overzicht, meeste controle over je team	(over automatisch) Meer naar de stress levels kijken en daarop reageren. Niet random gebruikers verplaatsen
2	Automatisch onvoorspelbaar en niet prettig. Advies soms verwarrend, omdat het dingen adviseert soms die onlogisch lijken. Stresslevel kan een toevoeging zijn, dus visualisatie ondersteuning geprefereerd boven baseline	(over visualisatie) Stresslevels leken niet altijd overeen te komen met wat soldaten zeiden te voelen (over automatisch) Aankondigen wat er gaat gebeuren zodat dat aan de soldaten gecommuniceerd kan worden
3	De adviezen waren goed, ik moet ze alleen beter leren negeren, anders blijf je de soldaten verplaatsen, wat te veel desoriënteert	(over interface) Je zag niet goed wanneer de fases overgingen (over baseline) Ik wist nog niet goed hoe de anderen het spel zagen, omdat ik als eerste was. Daarom was het soms onduidelijk (over advies) Ik stuurde te vaak mensen weg, de communicatie met de soldaten was niet voldoende van mijn kant (over automatisch) De gebruiker moet toch beslissingen kunnen overrulen. Ik werd een aantal keren om hulp gevraagd, maar ik kan niets