

PIC Play

A Serious Game for Alarms in the Paediatric Intensive Care

Bachelorproject

Pim van der Meulen

Dr. F. Cnossen*

January 31, 2016

Abstract

Apart from providing entertainment a serious game also has the goal of educating its users about a particular field. Brinks (2015) reported a lack of knowledge to identify causes of alarms in nurses on a paediatric intensive care unit (PICU). A serious game was created to help educate nurses in providing the most likely causes and treatments in accordance with alarms typically present on the PICU. A number of nurses helped in the development of the project. The design of the game was influenced by additional goals, such as maintaining motivation as well as including visual elements of the PICU. This led to the program being a top-down control game, which uses multiple-choice questions to improve the knowledge of its users. The effects of the game were investigated with the use of an experiment where the game was tested against a version of the program without the inclusion of game-elements. Students played either version followed by a standard written multiple-choice questionnaire. Test-results showed an increase over initial training-results. Furthermore, no significant difference was found between test-results of both groups. It can be concluded that the addition of game-elements does not appear to negatively influence the educational effects on students.

1 Introduction

If asked to name machines typically present in a hospital setting, one might recall “the Miracle of Birth” scene from Monty Python’s “The Meaning of Life”. In it we can see the most expensive machine in the whole

hospital: the machine that goes “ping”. In the paediatric intensive care unit (PICU), you likely hear more than just “ping”. With a wide range of instruments and equipment per patient, with each their own set of alarms with various levels, it should come at no surprise that the sounds will quickly turn into a chaotic mess of “bleeps”, “bloops” and “pings”. This is far from ideal for the medical staff, who, throughout the day, have to decide which alarms to filter out, or interpret as low-priority. Another party that is affected negatively are the patients themselves, since the constant barrage of auditive cues can be disruptive to their rest, sleep-cycle and overall well-being.

The importance of identifying the significance of the alarms is illustrated by the findings in the paper by Lawless (1994), in which it is concluded that “94% of alarm soundings in a pediatric ICU may not be clinically important”.

One of the eight paediatric intensive care units in the Netherlands is located in the Beatrix Children’s Hospital at the Academic Medical Center Groningen (UMCG). The department is divided into three colour-coded sections, with each section having the capacity for up to six or eight patients. The layout of section ‘Blue’ is a combination of two separated single-patient rooms as well as four patient beds that are spread out between the two rooms. When on duty, PICU nurses typically monitor two patients per nurse.

1.1 Previous research

The PIC Play project is built upon the findings of previous research done by Brinks (2015). Brinks (2015) originally sought after a solution to various problems on the PICU at the UMCG, but ultimately shifted fo-

*University of Groningen, Department of Artificial Intelligence

cus to performing an analysis of the entire alarm environment. This led to a list of the most prominent alarm hazards in the PICU, one of which was “a lack of knowledge to identify the cause of alarms”. The failure mode was identified through a risk analysis with PICU nurses and would occur when “there are no more rules in the decision table and a pattern match still has not been found”(p.48). This would entail a personal lack of knowledge of diagnosing an alarm. Brinks provided a solution to this hazard in the form of a knowledge-based system prototype which supported “decision making and training by suggesting diagnoses for patient monitor alarm combinations”(p.3). A knowledge base (KB) was created with the help of PICU nurses, who provided knowledge and rules. The KB contained 20 cases with each case consisting of one or multiple patient monitor alarms as well as the associated cause and treatment.

The alarms that were used in the KB were the most important physiological measurements according to Brinks (2015). These were the blood pressure (BP), the heart-rate (HR) and the oxygen saturation (Sat) of the patient. The values of these variables ranged from high to normal to low. If a measurement of an alarm was classified as normal it would not be present in a case.

1.2 Project goals

The main goal for this project was to create a training system in the form of a game to help educate nurses in providing the most likely cause and treatment for patient monitor alarm combinations. Additional goals were to use visual representations of elements from the paediatric intensive care as a way to provide entertainment, as well as to maintain a level of motivation for the user while playing the game. Overall, the use of a serious game would be well-suited for project requirements, since according to Djaouti, Alvarez, and Jessel (2011) serious games can be defined as “any piece of software that merges a non-entertaining purpose (serious) with a video game structure (game)”. This game-structure could also increase the level of motivation with elements such as high-scores to act as incentives for the user. Since Brinks (2015) already created a KB containing the causes and treatments for one or multiple patient monitor alarms, the database was to be integrated into the game. The project was accepted by the majority of the PICU nurses at the UMCG and a number of nurses provided feedback during the development.

The effects of serious games compared to that of

more traditional teaching methods is a controversial topic. In the paper by Virvou, Katsionis, and Manos (2005) an evaluation study was performed on a virtual-reality educational game. The main focus was “to measure the educational effectiveness of an educational VR-game as compared to educational software that does not incorporate the gaming aspect”(p.2). Their findings were that the effects of the VR-game were “very motivating while retaining or even improving the educational effects on students”. It was also mentioned that the educational effectiveness was “particularly high for students who used to have poor performance in the domain taught prior to their learning experience with the game”(p.1). On the other hand Akl, Pretorius, Sackett, Erdley, Bhoopathi, Alfarah, and Schuenemann (2010) examined the results of large number of papers for effects of educational games. Out of all papers only three suggested but did not confirm a positive effect on the knowledge of medical students. They ultimately concluded that “the available evidence to date neither confirm nor refute the utility of educational games as an effective teaching strategy for medical students”.

1.3 Evaluation

The underlying construct of this project is the efficiency of teaching something to someone. While the creation of the game is very significant to the project it is also important to evaluate its value. The research question for this project: Does the use of a serious game yield better results than the use of a standard learning method in helping to find the most likely cause and treatment for certain combinations of patient monitor alarms in the paediatric intensive care?

In order to measure the effectiveness of the program a pilot study was performed. In it, a group of students played either the game itself (PIC Play) or a stripped-down version of it (NO Play). To truly investigate the influence on the learning performance of the various game elements typically found in a serious game, the alternate version was created by removing these elements, e.g. the high-score, nurse and rewards for saving patients. What remained in the NO Play application were the learning model, KB, and problem-GUI.

Because of the arguable effects of serious games it is hard to hypothesize the impact of the PIC Play version versus that of the NO Play version.

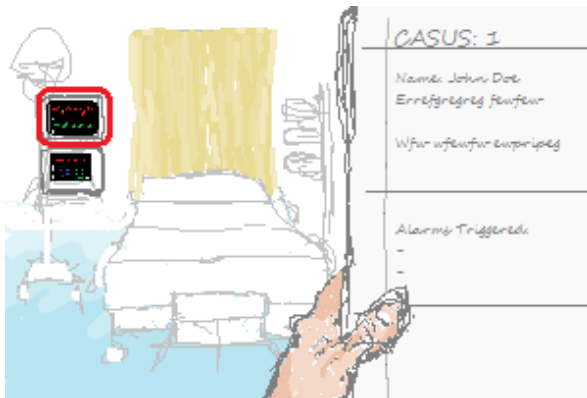


Figure 2.1: Simulation point and click concept

2 Development

In order to ensure that apart from the goals the project also stayed in line with the requirements of the user-base (PICU nurses), a number of PICU nurses were involved in the development cycle of the game. This involvement consisted of two groups of three nurses providing feedback on ideas and implementations during the initial and middle stages of the project.

2.1 Early concepts

One of the first steps in the project was to create a series of preliminary sketches, which would encapsulate various ideas for serious games. These concepts were designed with the following features in mind: overall user-friendliness; resemblance between the game and the actual PICU; level of distinction between simulation and casual game; manner of presenting the learning materials. With regard to the internal structuring of the program, an additional goal was to have the game's framework to be accessible for future additions, i.e. alternative knowledge-bases. In total, three game-concepts were created of which visual sketches can be seen in figures 2.1, 2.2 and 2.3. The first concept was created with a higher level of congruency in mind between the game and the PICU and having less emphasis on game-play elements compared to the other two concepts. The second concept was designed to be a middle-ground between being a simulation and a casual game. The third concept primarily contained elements of a game with a low level of resemblance of the PICU.

The first concept was a patient-simulator game in which nurses were presented with a set of cases with

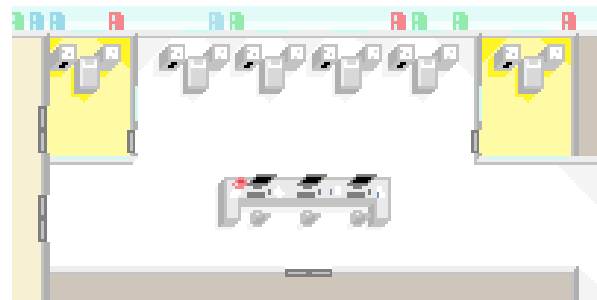


Figure 2.2: Casual top-down control concept

various combinations of alarms (see figure 2.1). The user is able to select monitors to investigate the values of different readings. Based on this information the user can provide the most likely cause and treatment with the use of multiple-choice responses. Incorrect answers would lead to a feedback response in which the correct answer is displayed and further explained. When all cases are completed a review window is displayed containing both the correct and user answers. The GUI of the game involves the view on the patient-bed as well as a virtual notepad containing most of the game-play information and buttons. As the game progresses the view will dynamically change with close-ups of different monitors, patient responses, visual alarms and additional small details such as turning pages of the notepad to depict various information about both patient and alarm readings.

The second concept was centred around a casual top-down control game (see figure 2.2). Instead of focussing on one case per patient, the user has control over a section of the PICU. With the help of a virtual-nurse the user can investigate each of the 6 patients-beds by dragging the nurse from one location to another. This concept also includes a time-cycle, which would trigger new alarms and patients on certain intervals. Based on the responses of the user patients are saved or lost after a number of correct or incorrect diagnoses.

The third concept (see figure 2.3) lacked the visual similarities between the PICU and the game that was present in the previous concepts and instead only contained the visual representation of a slot-machine. After initiating a new draw the user is presented with a combination of patient monitor alarms, e.g. a combination of two instances of 'Sat Low' and one instance of 'BP High'. The user is then able to bet on an option for both the cause and treatment. Depending on the an-

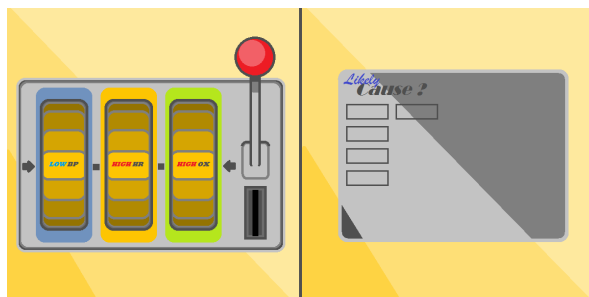


Figure 2.3: Slot-machine concept

swers money is lost or gained, which can possibly provide an incentive to keep playing. One of the underlying ideas of this concept is the ability for users to express their confidence through the manner of betting money on their answer. A learning system would keep track of the confidence level of each case in the knowledge-base and increase or decrease the probability of low and high level cases of appearing more often.

2.1.1 Concept feedback

The three concepts were presented to three nurses of the PICU in the UMCG. Apart from receiving feedback on these ideas, the sessions were also used to gain information about what nurses were looking for in a serious game that was designed for the PICU. The interviews were structured around a feedback form (see feedback form 1 in the appendix). After explaining the underlying problem of the project most nurses indicated that they recognized it. Out of the three concepts, either the second concept or a combination between the first two concepts were favoured. The third concept was seen as the least useful for the project by the nurses. Additional remarks were to incorporate the role of virtual patients more into the game concepts. The feedback led to the second concept being further developed into a prototype, with the addition of the multiple-choice function described in the first concept.

2.2 Prototype

To facilitate changes in the KB or the use of an different model entirely, the flow of the application is centred around two items: a menu section, where the user can choose the KB or start the game; the game section, where the user can play the actual game, pause or return to menu if needed. The game-play is centred

around the user moving the virtual nurse from patient to patient to solve various problems, with each problem containing a case chosen from the KB. A problem is visually represented by a red light located to the left of each patient bed. If a problem is present, the user can initiate a problem window, in which two questions are asked about the cause and treatment for the patient. To test the knowledge of the user, the questions are asked in a multiple-choice setting with four options per question. While the use of open questions would be much more beneficial to the user's understanding of why particular causes and treatments are associated with certain patient monitor alarms, the practicality of it in a program would be considerably lower. If both questions are answered correctly a correct diagnosis is made for the patient. While the original KB was written in Dutch, the prototype database was translated into English to increase the potential user-base for the program. The prototype GUI was also changed to English.

The project was built using a Java SE 6 environment. This was primarily done to improve the usability of the program for possible future development on both PC and mobile-platforms. Since not every computer is updated to the newer SE 7 version of Java, the environment was downgraded to SE 6 version, which more widely used.

2.2.1 Implemented features

After the framework of the game was created, the image in figure 2.2 was used as the background for the game. The various game elements such as the patients, nurse and alarms are drawn at specific locations that were already visually present in the image. The virtual nurse could be dragged to either one of three desk locations, situated in the middle of the screen, as well as next to the right of the patient beds. The patients are drawn at the patient beds in the image, with the possibility of alarms being drawn at the left side of the beds. A game-loop was implemented to keep track of time and used for updating the game-logic (though the majority of the logic was implemented after the prototype). Each in-game hour is set to last two seconds. A drag and drop function was implemented to increase the number dynamic elements in the program and thereby improving the level of immersion of the user. The virtual nurse can be picked up by mouse and dragged to predetermined locations, e.g. a bed or desk.

The problem window (final version shown in figure 2.4) is designed around a separate window, which is di-

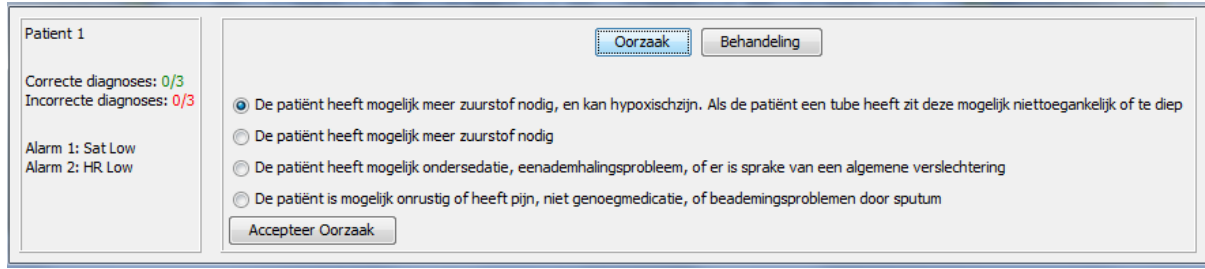


Figure 2.4: Example of a pop-up panel that is displayed when a problem needs to be solved.

vided into an information and diagnosis section. The former contains the patient-ID and the alarms of the case that was chosen for the problem. The latter provides two buttons to switch between the options for either the cause or treatment. The options are shown below the buttons with an addition confirmation button to accept an answer. When both questions are answered another button is added to the diagnosis section that is used to return to the game. For each of the questions, the correct answer is displayed along with the user response at the location of the previous options after the confirmation button is pressed.

2.2.2 Learning model

It was initially planned to use a difficulty setting that the actual nurses could manually alter to increase or decrease various items in the game such as the number of nurses and the probability of new problems and patients appearing per in-game hour. This was ultimately changed in favour of a more elegant learning model based on the Elo rating algorithm described in Schadenberg, Neerinx, Cnossen, and Looije (2015), that apart from providing a difficulty to the game also adjusts the probability levels of individual cases to better match cases that were perceived as easy or hard by the user. This perception is monitored by the system through number of times the case was either answered correctly or incorrectly.

Schadenberg et al. (2015) investigated effects of a rating system to “estimate a child’s skill at playing a game and keeping children motivated to interact with a social robot.” the system was an extension from the Elo rating algorithm originally published in Elo (1978) where it was used to calculate the user-levels of chess-players. The implementation that was discussed in Schadenberg et al. (2015) was used in this project.

$$r_{new} = r_{old} + K(s - E(r)) \quad (2.1)$$

The Elo algorithm updates the levels of both the user and each individual case with the formula 2.1, where r_{new} is the new level (with r itself being a float), r_{old} is the previous value, K is the learning-rate, s is the binary result for the problem (1 for correct answer, 0 for incorrect), $E(r)$ is the expected outcome of the problem. This in turn is found by either formula 2.2 for each case or 2.3 for the user.

$$E(r_{case}) = \frac{10^{r_{case}/400}}{10^{r_{user}/400} + 10^{r_{case}/400}} \quad (2.2)$$

$$E(r_{user}) = \frac{10^{r_{user}/400}}{10^{r_{user}/400} + 10^{r_{case}/400}} \quad (2.3)$$

Here, r_{case} is the level of the case and r_{user} is the level of the user. The value for s depends on whether r is being calculated for the case or for the user. If the user diagnoses a problem correctly the s value for the user is ‘1’ and for the case ‘0’. This leads to the user level increasing each time a correct diagnosis is given, while the case level decreases. The denominator for the r value acts as magnifier function. For every difference value of 400 between the user value and case value, the expected value of one case is increased ten times in comparison with the expected value of the other. The exact value of the denominator can be led back to the original use of the Elo algorithm for chess.

$$P_{case} = \frac{r_{case}}{\sum r_{cases}} \quad (2.4)$$

For the implementation of this program the value of the learning-rate K is locked to 1, for it was assumed that this rate would not have to be dynamic in order to have the learning model function correctly. In order to

be able to use r values to choose a case suited for the player a transformation is required of the r values into probabilities. This is achieved by first transforming the case levels into only positive values. This is accomplished by finding the minimal value below zero of all cases and adding this to every case level. Secondly, the adjusted levels are used to calculate the probabilities according to equation 2.4, where P_{case} is the probability of a case, r_{case} is the case level and r_{cases} are all case levels. The probability of each case being chosen for a problem is calculated through the case value being divided by the total value of all case values combined.

2.2.3 Problem generation

If a new problem is required, a random case is chosen from the KB based on the previously calculated probabilities per case. After a case is chosen the associated cause and treatment are returned (as well as the patient monitor alarms) and each used as the first option out of 4 total that the player can choose as an answer. The remaining 3 are randomly chosen out of other cases in the KB one by one until all options contain unique answers for both types. The order of the options is randomised before presented in the problem panel.

Since the learning model uses probabilities to choose a case for a new problem, it could be possible that some cases would never be chosen. This was addressed by implementing a function in the problem-generation process that ensures that every case in the KB is picked before probabilities are used to choose cases for problems. A list is generated containing all the cases from the KB. Instead of using probabilities to create a new problem, a case is randomly chosen from this list (and also removed from it). This will cause all the cases to be chosen at least once before the game is finished.

2.2.4 Prototype feedback

The prototype was presented to a second group of 3 PICU nurses which provided additional feedback. The interviews were structured around another feedback form (see feedback form 2 in the appendix). The nurses played the prototype for a short period of time (the duration of solving 5 problems) and look at the contents of the KB. There were three main issues that the nurses identified: the locations to drag the nurse to were difficult to locate; the English language was below the intermediate level for some PICU nurses; incomplete causes and treatments for some of the cases in the KB. Over-



Figure 2.6: Example of a separated room containing a patient and a nurse area outline, without overview of a nurse.

all, the majority of the nurses were positive about the prototype.

2.3 Final version

The localization of the areas to drag the nurse towards, such as the desks and patient beds, was improved by the addition of a blue outline to every available location whenever the nurse was picked up by the user (see figure 2.6). The language of the KB, as well as the GUI, were changed back to Dutch to accommodate for PICU nurses with poorer understanding of the English language. The indicated issue with the KB was addressed by having the same expert that helped Brinks (2015) in creating the initial KB to review the feedback and suggest alterations to the database. This resulted in an improved KB (included in the appendix).

The other new additions to the prototype included: a scoreboard; additional game-logic; pause button; pause-screen. The scoreboard contains the high-score of the user, the time that has passed in hours and days, the number of patients that have been saved or lost and the total number of problems that have been completed (both correctly and incorrectly). The elements of scoreboard are divided between the top and bottom of the screen as can be seen in figure 2.5. The high-score is increased by 5 points for each correct question (up to 10 points per problem), as well as by 5 extra points for making a correct diagnosis (answering a problem correctly completely).

The final version also included a new game-play element: if no nurse is present in either a separated room or in one of the desks, the particular room becomes dark, obscuring the view of the visual problem indicator (see figure 2.6). This concept resembles to some extent the situation in the actual PICU, where the beds in the separated rooms are more obscured than others. The im-

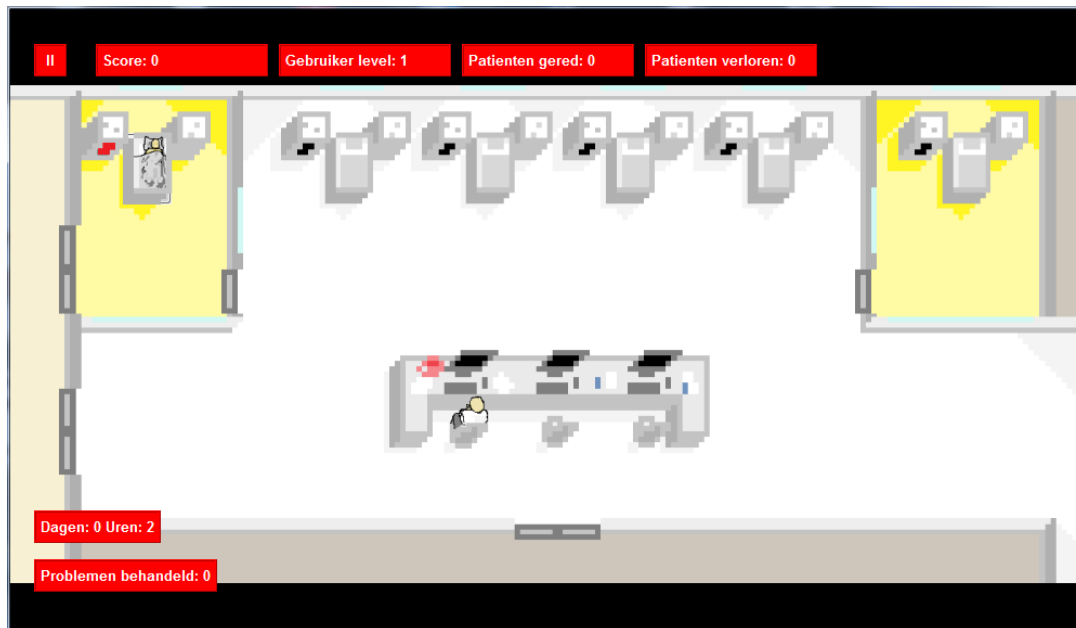


Figure 2.5: The general view of the game.

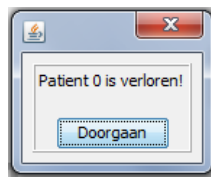


Figure 2.7: Example of a pop-up window that is displayed when a patient is lost.

plementation of patient lives is accomplished by having the results of each problem of a particular patient used for two internal and visual counters. If three correct or incorrect diagnoses are made, the patient is saved or lost respectively. Saving a patient earns the user 50 additional points.

Other features that were implemented in the final version were a patient outcome window (see figure 2.7), patient status pop-up window (see figure 2.8) and log-file storing. The patient outcome window appears when a patient is lost or saved and displays the outcome of the patient as well as the potential bonus points for a save. The window is closed by pressing either a button at the bottom of the panel or the typical close button located at the upper-right of the window. The panel is used as a confirmation tool to ensure that users are aware of the consequences of their actions and thereby

increasing the level of immersion of the user while playing the game. Likewise, the patient status window is used to show the effects of completing problems as well as to help the user keep track of the progress of patients, since the window depicts the number correct and incorrect diagnoses of a particular patient. The window is displayed when the user places a nurse next to a patient bed and clicks on the patient when no problem is present. The log-file storing feature was added to save the results of each play-session. Each file contained the list of items: the problems that were completed; the case-number, the location of the bed from where the problem was called; the patient-ID; the user-level; the results for both the cause and treatment questions (true, false). After the game was terminated, the log file was updated additional items: the high-score; the number of patients that were saved and lost; the number of correct and incorrect problems; the total play-time; the user-level; the individual case-levels.

The problem window was updated with the number of correct and incorrect diagnoses for a particular patient in the information section of the window. An example can be seen in figure 2.4.

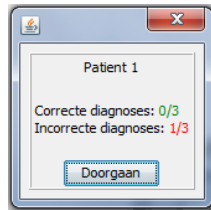


Figure 2.8: Example of pop-up window that is displayed when the user checks the status of a patient.

2.3.1 Playtest feedback

Before the pilot study was run, a one-participant playtest was performed on the final version of the program to calibrate the probabilities of patients and problems appearing per in-game hour, as well as gauge the speed of completing problems by the participants and detect possible anomalies in the game. The participant of the pilot was instructed to play the game until 50 problems were completed (training phase), after which the participant had answer a written multiple-choice test containing all 20 cases from the KB (test phase).

The participant reached the target number in under 40 minutes and reported that throughout the game there were always the maximum number of patients and problems present. After 50 problems, the participant did not feel more familiar with the information that was provided during the game. This was reflected in the resulting low scores for both the training and test phase of the pilot.

The results of the pilot led to a number of changes to the version of the game that was used for the experiment. The probabilities for new patients and problems were reduced from 20% to 1.05% per hour. This new value is increased per type (patient or problem) per hour if no instance of the type is created. This leads a probability of 100% after 4 in-game days. The individual probability values are reset if a new instance is added to the game. The pilot also showed that 50 problems formed a reasonable limit for a stopping criterion for the pilot, although the use of a KB containing 20 cases could be too difficult for participants. Especially for those with limited knowledge on making a correct diagnosis based on patient monitor alarms. This would also lead to most of the cases only be picked a few times, which would decrease the functionality of the learning system. This was addressed by removing the even numbered cases, resulting in a KB with 10 entries. While it was not the result of the pilot study a final alteration

to the program was an automatic termination function that closed the application after 50 problems were completed. This made sure that participants would not accidentally overshoot the threshold by answering more than 50 problems.

2.3.2 NO Play

To test the effects of the game another program the NO Play version was used as a standard. NO Play is created out of the final version of PIC Play through the removal of the various game-elements. While NO Play kept an identical KB and learning model, the problem panel was adjusted to work as a main window. Whenever the user would complete a problem the panel would be updated with a new instance of a problem. The window also included the number of the current problem instead of the patient-ID. Correct or incorrect diagnose counters were replaced with counters keeping track of the number of correct and incorrect problems. All other items, such as the representation of PICU, patients, alarm-indicators, beds, nurse and points, were removed.

3 Pilot Study

3.1 Introduction

The goal of the pilot study was to test if the additions of game-elements improved the learning success of participants. Because the program was constructed around the concept of a serious game, this goal was achieved by comparing the results of PIC Play with that of the NO Play version, which had all game-elements removed from the program. This led to the research question for this pilot study: Does the addition of game-elements improve the results of participants correctly providing the most likely cause and treatment for combinations of patient monitor alarms in the paediatric intensive care?

The pilot consisted of two groups of participants playing either the PIC Play or NO Play version until 50 problems were completed, followed by multiple-choice test containing all the different cases that were used during the training. The variables that were measured were the number of correct questions for both the cause and treatment per patient per condition (PIC Play, NO Play). This was recorded during both the training and testing phases, as well as the individual times to complete the training and test sections of the study.

Since the papers by Cvach (2012) and Akl et al. (2010), discussed in the introduction, illustrated that the effects of serious games are arguable, it is difficult to hypothesize whether an improvement will be present in the results of the PIC Play version in comparison with that of the NO Play version.

3.2 Methods

18 participants participated in the pilot. The participants were students between the ages 18 to 30 years old and had a wide range educational backgrounds. 10 participants played the PIC Play version, while 8 participants played the NO Play version. The results of two participants were removed from the data, due to a combination of extremely low scores and test and training-completion times, which was seen as a suspected lack of interest.

The pilot was performed in a designated experiment room, which included three separated sections. Each section contained one desk and chair, leading to a maximum of three participants per session. During a session each participant was assigned an dual-boot Apple Macbook, on which Windows 7 was loaded. The results of the training were stored in a log-file.

In total there were 3 conditions: the different phases in the study (training or test); the version of the program (PIC Play or NO Play); the type of question (cause or treatment). The study used a between-subject design, where participants played either the PIC Play or NO Play version as training. This design was used to make sure that carry-over effects would be avoided, as well as other sources that could influence the results. The test phase was made up of a written multiple-choice test containing 10 cases (included in the appendix). Each case consisted of two questions: for the cause and for the treatment.

At the start of the pilot the participants were instructed to read the instructions for the training phase. The instructions for both versions (included in the appendix) were presented on screen. After a participant was finished reading they were instructed to close the associated windows. The instructions contained the rules of completion of the game, which were to either solve 50 problems or play for 60 minutes (whichever came first) and the various windows and visual representations that were present in the program, such as the problem window, patients, etc. An overview was also provided on screen of terminology that participants could possibly find difficult to understand. Items

that were included were the different patient monitor alarms (Sat LOW, BP HIGH, etc.) and how to interpret them, but also words such as 'drain', 'sedation' or 'hypoxic'. After the instructions were read and closed, the participant started a game and continued to play until the target-conditions were met. The participant then signalled the surveillant, who provided the written test along with an answer sheet. The instruction for the test phase was presented on screen. A feedback form was included for participants who played the PIC Play version during the training and was used to gauge the responses of the participants on the program. The form was based on the system usability scale (or SUS-scale) created by Brooke (1996). Additional questions were added that asked if the participant had further remarks and if he or she thought that the game helped them with learning. The test instructions, as well as the test itself, the answer sheet and the feedback form are included in the appendix.

As a pre-training measure the first 10 problems were used, because it would show the knowledge of the participant before training.

Because multiple-choice examination was used in the test, the number of correct cause and treatment questions were transformed into grades using formula 3.1, with G being the grade, Q_c the number of correct causes, Q_t the number of correct treatments and T being the total number of questions (in this case 20). Since four options were available for each question, a participant could typically guess 25 percent of the questions correctly, or in this case 5 questions. This number is subtracted from the correct questions in order to compensate for this. A grade was created by multiplying this value with the grade-variable (10 divided by 75 percent of the total, or 15).

$$G = ((Q_c + Q_t) - T * 0.25) * \frac{10}{T * 0.75} \quad (3.1)$$

Since the data constituted the grades and version variables, being continual and categorical respectively, as well as the participant-groups being independent of each other, a Mann Whitney U test was performed to investigate the difference between the grades of both versions.

3.3 Results

Figure 3.1 shows the grades (calculated with the number of correct cause questions (CC) and correct treat-

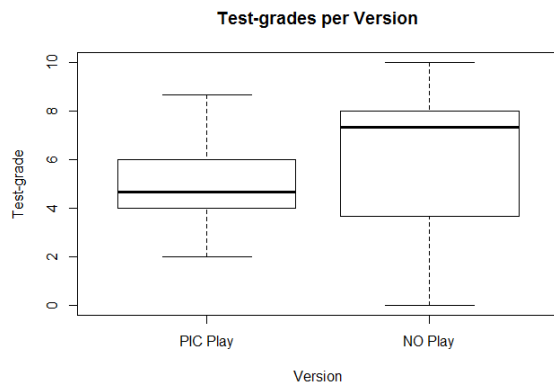


Figure 3.1: Box-plot of test grades for both PIC Play (Mdn = 4.67, n = 9) and NO Play (Mdn = 7.33, n = 7) versions of the program, containing the distribution of the grades including the median.

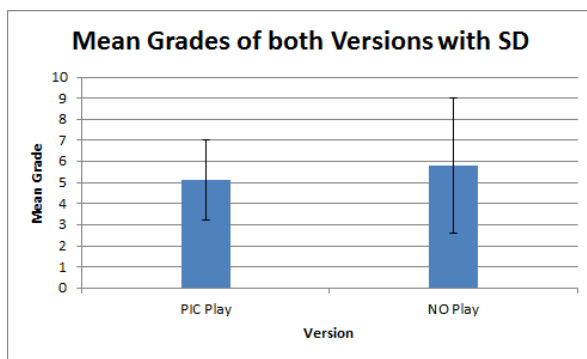


Figure 3.2: Bar chart of the means for both PIC Play (M = 5.11, SD = 1.91) and NO Play (M = 5.81, SD = 3.22).

ment questions (CT) per participant using equation 3.1) for both PIC Play and NO Play versions. The distribution for PIC Play was relative normal. The means of the grades for the PIC Play and NO Play conditions were 5.11 and 5.81 respectively, with a SS of 1.91 and 3.22. To better visualize this, the means and SD are presented in a column chard in figure 3.2 for both conditions and mean grade. The means are visually not extremely different from each other. A Mann-Whitney test indicated that the test grades for participants playing the PIC Play version (Mdn = 4.67) were not statistically significantly higher than that of the participants playing the NO Play version (Mdn = 7.33), $W = 26.5$, $p\text{-value} = 0.633$.

The test and training results for the CC for PIC Play were shown in figure 3.3 and for the CT in figure 3.4.

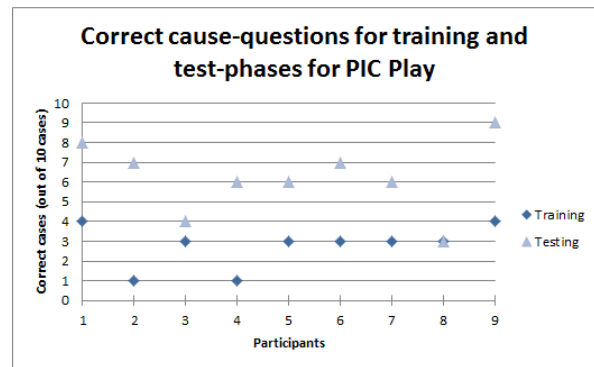


Figure 3.3: Scatter plot of the number of correct cause questions for the training (M = 2.78) and test phases (M = 6.22) for each participant that used the PIC Play version during training.

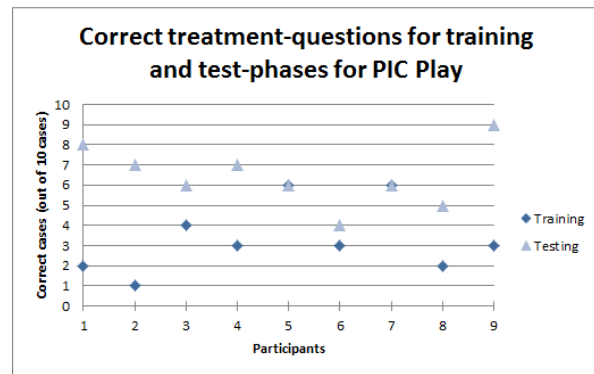


Figure 3.4: Scatter plot of the number of correct treatment questions for the training (M = 3.33) and test phases (M = 6.44) for each participant that used the PIC Play version during training.

The scatter plots show the scores of participants for the training and testing phases. In both figures it can be seen that for each participant the correct number of questions for the testing phase was higher or equal than the correct number of questions for the training phase. Note that there are two instances in figure 3.4 where participants answered 6 out of the first 10 treatment questions correctly for the training phase. Overall, participants improved their scores considerably: M = 2.78 to M = 6.22 for correct cause questions and M = 3.33 to M = 6.44 for correct treatment questions.

For the NO Play variant the test and training results for the CC were shown in figure 3.5 and for the CT in figure 3.6. The scatter plots show the scores of participants for the training and testing phases. It can be seen

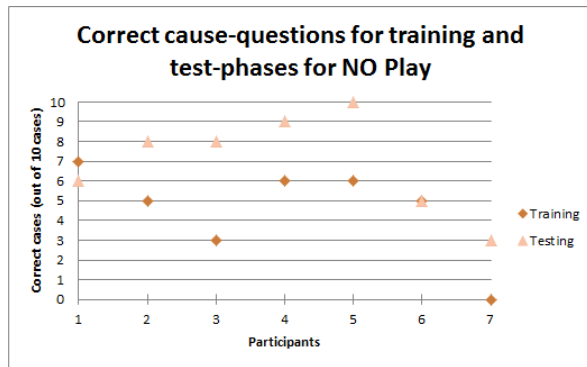


Figure 3.5: Scatter plot of the number of correct cause questions for the training ($M = 4.57$) and test phases ($M = 7.00$) for each participant that used the NO Play version during training.

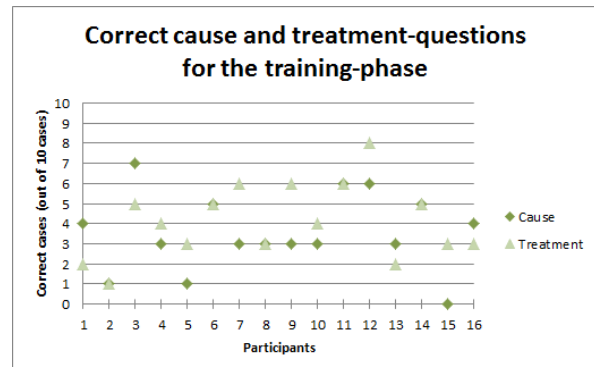


Figure 3.7: Scatter plot of the number of correct cause ($M = 3.56$) and treatment questions ($M = 4.13$) for the training phase for each participant (both PIC Play and NO Play).

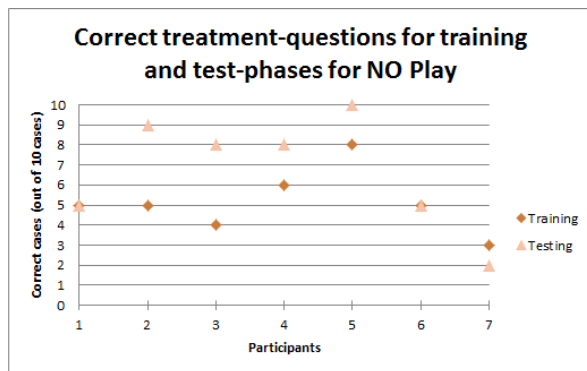


Figure 3.6: Scatter plot of the number of correct treatment questions for the training ($M = 5.14$) and test phases ($M = 6.71$) for each participant that used the NO Play version during training.

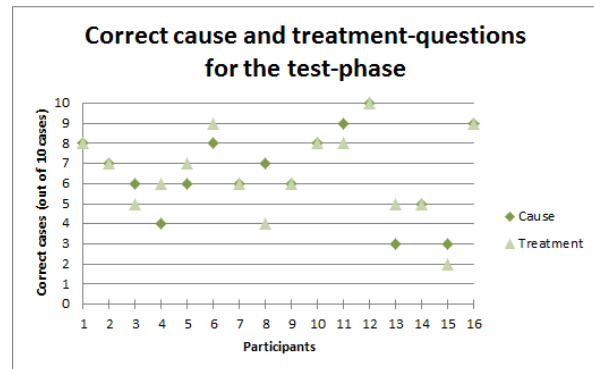


Figure 3.8: Scatter plot of the number of correct cause ($M = 6.56$) and treatment questions ($M = 6.56$) for the test phase for each participant (both PIC Play and NO Play).

that the majority of participants score higher in the testing phase than in the training phase, although there also instances of participants scoring equal or lower than in the training phase. It also became evident that some participants for the NO Play version did extremely well for the training phase, with multiple instances of participants answering 6 to 8 out of the first 10 questions correctly for both the cause and treatment. This is further reflected in a lower increase of results for the test phase over the training phase compared to the PIC Play version: $M = 4.57$ to $M = 7.00$ for the correct cause questions and $M = 5.14$ to $M = 6.71$ for the correct treatment questions.

To illustrate the difference in results for the types of question (cause or treatment), the CC and CT per par-

ticipant for the training phase are shown in figure 3.7 and for the test phase in figure 3.8. The scatter plot in figure 3.7 shows that the correct number of questions for both types ranged between 0 and 8. There were 5 instances of participants answering an equal number of questions correctly for both question-types. Overall, the CT was slightly higher than the CC: $M = 4.13$ for the CT and $M = 3.56$ for the CC. The scatter plot in figure 3.8 shows that the correct number of questions for both types ranged between 2 and 10. There were 8 instances of participants answering an equal number of questions correctly for both question-types, as well as multiple instances of the number of correct questions being almost equal for both question types (for a deviation of 1). Overall, the CT was equal to the CC: $M = 6.56$ for the CT and $M = 6.56$ for the CC.

A scatter plot of the influence of completion times

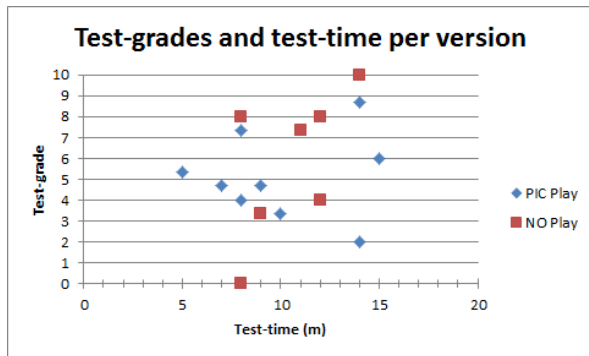


Figure 3.9: Scatter plot of the test grades against the test time (in minutes) per version of the game.

(in minutes) on the test grades can be seen in figure 3.9. The time for completion for PIC Play was between 5 and 15 minutes, for NO Play this was between 8 and 14 minutes. No correlation was visible between the test time and test grade for PIC Play. For NO Play, the figure shows a steep positive correlation between the test grade and completion time.

3.4 Discussion

Since the Mann-Whitney U test failed to indicate a significant difference between the game versions for the test grades, the additions of game-elements do not appear to significantly lower the results of the learning-system. In this regard a serious game does not negatively affect the knowledge of the participants. For this experiment there was a relative low sample-size of 16 participants in total, which influenced the statistical test by having a low power, lowering the statistical conclusion-validity. The other elements of this type of validity, such as fixed conditions and homogeneous participant groups, were achieved through using written instruction, separated experiment rooms and random division. Although the results of the pilot study with the recruitment of students should not be compared to that of a equivalent study with PICU nurses, the choice of student participants is not necessarily unfit for the scope of demonstrating the efficiency of teaching something to someone.

The box-plot in figure 3.1 shows that while the results for the NO Play were higher, the SD was lower for the PIC Play version. The cause for this is not clear based on our data.

As can be seen in figure 3.2 the mean grades for PIC

Play was 5.11 and 5.81 for NO Play. Although it is difficult to find the exact reason for this, a number of explanations can be made.

Since the pool of participants contained students from different levels of education, some individuals could have been inexperienced with participating in experiments, thereby leading to low scores. Because the participant backgrounds were not recorded, this could possibly lead to a point of interest for future research.

Another explanation is that the program is simply too difficult for some of the participants. This could be related to the possibility that subjects have trouble learning the medical jargon that is used throughout the game, but also that the learning system itself is functioning inadequately. One drawback of the game is that it always goes through the entire knowledge base before reverting to randomisation for additional problems. A user could answer some of the initial questions correctly just by guessing, which would lower the probability of those cases to come back at later time. There is no further control to make sure that a user is confronted with sufficient and appropriate cases apart from a random chance. If possible, additional research could be in the implementation of other learning models, or simply improvements on the already existing version.

Participants scored higher on the test phase than the training phase for both versions of the game as can be seen in figures 3.3, 3.4, 3.5 and 3.6. Due to using multiple-choice examination in both versions of the program, the participants could be able to remember unique characteristics of a particular correct answer, without knowing the complete answer. Even though the possible answers are randomized per question some participants would still be able improve their performance without improving their knowledge on the patient-monitor alarms. Because of this the improvement on the test phase results cannot be attributed to an increase in knowledge.

To investigate the relation between the number of correct cause and treatment questions we can look at the scatter plots in figure 3.7 for the training and figure 3.8 for the testing. Whereas the CT and CC per participant for the training phase were not too similar in number, the CT and CC for the testing phase were more closely related to each other. This could be the result of random variation, but also due to the participants using the result of the answer for the cause as a hint for the answer for the treatment question. This was possible since the correct answers were given per question, instead of in the end. Unfortunately we cannot decide which cause

is the most likely based of our data.

There are multiple directions and implementations for future research. To increase the effectiveness of the implemented learning model, the learning rate K could be adjusted to change as the game was played, which would lead to the user level to be more useful in the analyses about the performance of users. This analysis could also be improved by storing more information during the training phase, such as the answers that are given for each question per problem, instead of only recording if the answer were correct or not. The learning model could also be replaced by other algorithms, which could then be compared to each other. For future experiments the participants could be asked about how they felt while playing the game and how this effected their motivation. For this experiment the educational background of each participant was not specifically registered, which made further analysis of this background versus the performance of the participant impossible. If enough participants are included in the study, it could be an area of interest for future studies. The experiment design could also be changed in favour of longer periods of training, which would make the use of the entire KB less of an obstacle.

References

- E. A. Akl, R. W. Pretorius, K. Sackett, W. S. Erdley, P. S. Bhoopathi, Z. Alfarah, and H. J. Schuenemann. The effect of educational games on medical students' learning outcomes: A systematic review: Beme guide no 14. *Medical teacher*, 32(1):16–27, 2010.
- K. Brinks. Alarm hazards on the pediatric intensive care unit. Master's thesis, University of Groningen, 2015.
- J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- M. Cvach. Monitor alarm fatigue: an integrative review. *Biomedical instrumentation & technology, Association for the Advancement of Medical Instrumentation*, 46(4):268–77, 2012.
- D. Djaouti, J. Alvarez, and J. P. Jessel. Classifying serious games: the g/p/s model. *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*, 6: 118–136, 2011.
- A. E. Elo. *The Rating of Chess Players Past and Present*. Arco, New York, 1978.
- S. T. Lawless. Crying wolf - false alarms in a pediatric intensive-care unit. *Critical Care Medicine*, 22(6): 981–985, 1994.
- B. R. Schadenberg, M. A. Neerincx, F. Cnossen, and R. Looije. Improving child-robot interaction by adapting the difficulty of games. Unpublished manuscript, 2015.
- M. Virvou, G. Katsionis, and K. Manos. Combining software games with education: Evaluation of its educational effectiveness. *Educational Technology & Society*, 8(2):54–65, 2005.

4 Appendix

4.1 Feedback Form 1

Feedback Interview No. 1		Interview:
<i>(Na uitleggen probleem)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> Herkent u dit probleem? 		
<i>(Na uitleggen probleem)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> Als u een spel zou moeten maken om dit probleem op te lossen, hoe zou deze er uit zien? 		
<i>(Na presenteren concepten)</i> <u>Vraag:</u> <ul style="list-style-type: none"> Wat zijn u impressies over de concepten? Welk concept zou u als eerste spelen? 		
<i>(Na presenteren concepten)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> Heeft u aanvullingen of verdere opmerkingen? 		

4.2 Feedback Form 2

Feedback Interview No. 2		Interview:
<i>(Na voorleggen spelconcept)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Wat vindt u van het spelconcept? Impressies? 		
<i>(Terwijl gebruiker spel speelt)</i> <u>OBSERVEER & VRAAG:</u> <ul style="list-style-type: none"> • Zijn er elementen die u onduidelijk lijken? • Hoe zouden uw collega's hierover denken? 		
<i>(Terwijl gebruiker spel speelt)</i> <u>OBSERVEER:</u> Waar heeft de gebruiker moeite mee?		
<i>(Na gespeeld te hebben)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Wat vindt u van de rol van de patiënt? • Wat vindt u van de rol van de verpleegkundige? 		
<i>(Na gespeeld te hebben)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Wat vindt u van het gebruik van "levens"? • Heeft u suggesties of alternatieven? 		
<i>(Na gespeeld te hebben)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Wat vindt u van de taal? • Heeft u een voorkeur? • Hoe zouden uw collega's hierover denken? 		
<i>(Terwijl lijst-Koen)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Kunt u zich vinden in de oorzaken en gevolgen hier beschreven? 		
<i>(Na gespeeld te hebben)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Wat vindt u van de naam? • Heeft u suggesties of alternatieven? 		
<i>(Na gespeeld te hebben)</i> <u>VRAAG:</u> <ul style="list-style-type: none"> • Vindt u nog iets missen in het spel? • Heeft u nog opmerkingen? 		

4.3 Knowledge-base 2

Knowledge Elicitation Results Version 2

Casus	Alarms	Cause	Treatment
1	Sat Low	De patiënt heeft mogelijk meer zuurstof nodig	Check de machine en de patiënt. Als het een cardio patiënt betreft, check of de shunt dicht zit
2	Sat High	De patiënt heeft mogelijk minder zuurstof nodig	Check de machine en de patient
3	Sat Low HR Low	De patiënt heeft mogelijk meer zuurstof nodig, en kan hypoxisch zijn. Als de patiënt een tube heeft zit deze mogelijk te diep	Check de machine en de patiënt
4	Sat Low HR High	De patiënt is mogelijk onrustig of heeft pijn, niet genoeg medicatie, of beademingsproblemen door sputum	Check de patiënt
5	Sat Low BP Low	De patiënt heeft mogelijk oversedatie, een ademhalingsprobleem, of er is sprake van een algemene verslechtering	Check de patiënt
6	Sat Low BP High	De patiënt heeft mogelijk ondersedatie, een ademhalingsprobleem, of er is sprake van een algemene verslechtering	Check de patiënt
7	Sat High HR High	Mogelijk is er sprake van een algemene verbetering	Check of het zuurstof percentage of de beademingsdrukken moeten worden afgebouwd
8	Sat High BP High	Mogelijk is er sprake van een algemene verbetering	Check of het zuurstofpercentage of de beademingsdrukken moeten worden afgebouwd. Check of de bloeddruk reëel is.
9	HR High BP High	Er is mogelijk sprake van ondersedatie, de patient heeft koorts, of de patient heeft te veel inotrope middelen.	Check de patiënt.
10	HR High BP Low	Er kan sprake zijn van ondervulling, een vocht tekort	Geef medicijnen zodat de bloeddruk weer op een goed niveau komt. Onderzoek de patiënt voor een volledige diagnose, check of deze een hartritmestoornis heeft
11	HR Low BP High	De patiënt heeft mogelijk last van inklemming of een vagale reactie. Er kan ook een drain zijn	Verlaag de eventuele hersenzwelling met medicijnen, of in het geval van een drain deze af laten lopen.
12	HR Low BP Low	Mogelijk treed een harttamponade op, is er sprake van oversedatie of is er een verslechtering van de patient.	Check de patiënt en check de machine voor hartdruk metingen. Indien pacemaker, check of deze nog goed werkt
13	Sat High HR High BP High	Er is mogelijk sprake van ondersedatie, de patient heeft koorts, of de patient heeft te veel inotrope middelen.	Check de patiënt

14	Sat High HR High BP Low	Er is mogelijk een probleem met de inotrope medicatie of een circulatieprobleem	Check de patiënt
15	Sat High HR Low BP High	De patiënt heeft mogelijk last van sputum	Check of de patiënt moet hoesten
16	Sat High HR Low BP Low	De inotrope medicatie loopt mogelijk niet goed, of er is oversedatie	Check de patiënt
17	Sat Low HR High BP High	De patiënt heeft mogelijk stress, pijn of verdriet. De oorzaak hiervan kan ondersedatie of een beademingsprobleem zijn.	Onderzoek de patiënt
18	Sat Low HR High BP Low	De patiënt is mogelijk in shock	Check de patiënt en de beademingsmachine, check of de inotrope middelen aankomen
19	Sat Low HR Low BP High	De patiënt heeft mogelijk last van inklemming of een vagale reactie of een beademingsprobleem	Verlaag de eventuele hersenzwelling met medicijnen. Indien beademings probleem, check of de patiënt moet hoesten
20	Sat Low HR Low BP Low	Er is mogelijk een circulatie of ventilatie probleem. Het kan ook een probleem met inotrope middelen zijn. Dit kan veroorzaakt worden door een verstopping van de tube.	Check de patiënt en neem de oorzaak van het probleem weg. Geef meer zuurstof, verhoog de beademing wanneer nodig, verhoog de inotropie, check de tubes. Indien dit niet helpt, start reanimatie.

4.4 Protocol Experiment

EXPERIMENT PROTOCOL PICPLAY

VOOR Binnenkomst proefpersonen:

- 1) Boot-up computer.
- 2) Leg toets instructies formulier klaar.
- 3) Leg toets formulier klaar.
- 4) Leg antwoord formulier klaar.
- 5) Leg pen klaar.
- 6) Vul proefpersoon nummer in op antwoord formulier.
- 7) Leg evaluatie formulier klaar (voor PICPlay proefpersonen).
- 8) Laad instructies voor model.
- 9) Laad model.

NA Binnenkomst proefpersonen:

- 1) Welkom proefpersonen.
- 2) Laad proefpersonen registratie formulier invullen.
- 3) Laad proefpersonen consent formulier invullen
- 4) Vertel kort wat er gaat gebeuren.
- 5) Plaats proefpersonen achter computers.

START EXPERIMENT SPEEL FASE

TIJDENS Experiment speel fase:

- 1) Sla starttijden op in het experiment logboek.
- 2) Neem eigenaardigheden op tijdens het experiment in het logboek.
- 3) Los problemen op of beantwoord vragen wanneer nodig.

NA Experiment speel fase:

- 1) Sla eindtijden op in het experiment logboek.

START EXPERIMENT TEST FASE

TIJDENS Experiment test fase:

- 1) Sla starttijden op in het experiment logboek.
- 2) Neem eigenaardigheden op tijdens het experiment in het logboek.
- 3) Los problemen op of beantwoord vragen wanneer nodig.

NA Experiment test fase:

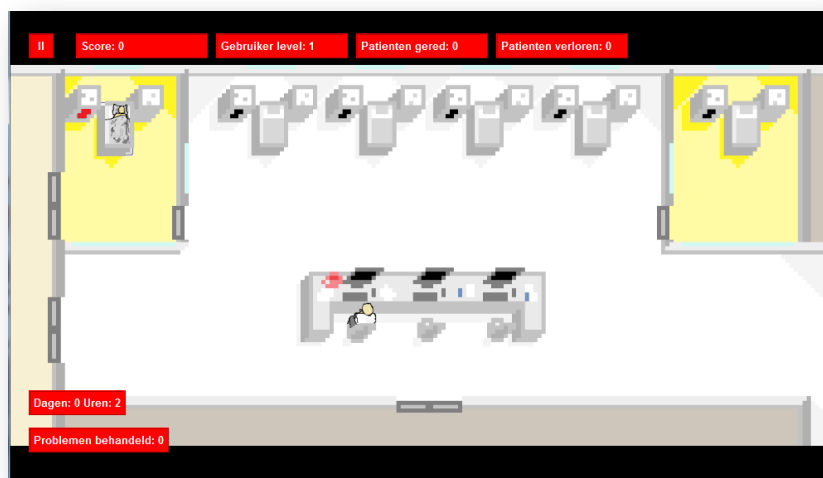
- 1) Sla eindtijden op in het experiment logboek.
- 2) Verzamel papieren.
- 3) Bedank proefpersonen.
- 4) Beantwoord verdere vragen.
- 5) Zie proefpersonen uit.
- 6) Verzamel data van computers.

4.5 Instructions PICPlay

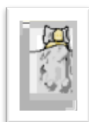
INSTRUCTIES PICPLAY

Dit zijn de instructies voor het leer- en speel-onderdeel van het experiment. Tijdens dit onderdeel is het de bedoeling spelende wijs te leren wat de oorzaak en gevolg van een bepaalde set alarmen zijn. Het onderdeel is afgelopen als je **50 problemen** hebt gehad of langer dan **60 minuten** hebt gespeeld. Werk zo accuraat mogelijk en neem je tijd!

Overzicht



Hierboven is het overzicht van het spel te zien. De kinder-intensive care bestaat uit 6 patiënt-bedden die op een willekeurige tijd een patiënt kunnen bevatten. Naarmate de tijd verstrekt zal je nieuwe patiënten zien verschijnen.



Nieuwe patiënten hebben altijd een nieuw probleem. Oude patiënten kunnen na verloop van tijd een nieuw probleem krijgen. Als je **3 keer** een probleem bij dezelfde patiënt oplost is deze gered (bonus punten). Als je **3 keer** het probleem verkeerd beantwoord is de patiënt verloren (er zal verder niets gebeuren).

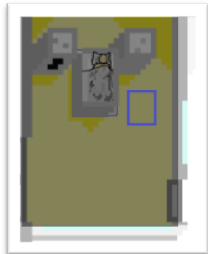


Je kan aan het bed van de patiënt zien of er wel of niet een probleem aanwezig is. Als het rood lampje brandt dan weet je dat er iets mis is met de patiënt. Nadat je het probleem goed of fout hebt beantwoord gaat het lampje weer uit.



Verder bevat het veld een verpleegkundige die je kunt gebruiken om problemen van een patiënt te bekijken en op te lossen. Je kunt de verpleegkundige **oppakken en slepen** naar een bed met je muis. Dit doe je door de muis ingedrukt te houden op de verpleegkundige en dan de muis te slepen naar een **blauw vlak** naast een bed.





Aan weerszijde van het veld staan speciale **quarantaine-kamers** waarbij het zicht, net als bij intensive care afdelingen, beperkt is. Dit betekent dat als er geen zicht is op de kamer (er staat **geen** verpleegkundige bij het **bed** of bij het **bureau**) je niet kan zien of er wel of niet een probleem gaande is bij een betreffende patiënt. Vergeet dus niet om geregeld deze kamers te checken.

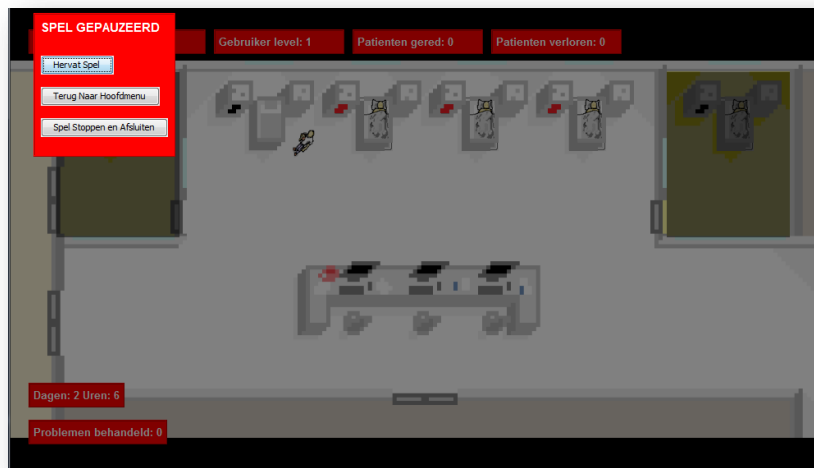
Zoals je kunt zien in het overzicht worden tijdens het spel bepaalde dingen weergegeven. Deze zijn :

- **de score**, die je verhoogd door goede antwoorden te geven en patiënten te redden.
- **het niveau van de gebruiker**, dat veranderd aan de hand van je antwoorden.
- **het aantal** patiënten dat je hebt **gered** of **verloren**.
- **de tijd**, waarbij elk spel-uur gelijk staat aan 2 seconden.

En tenslotte:

- **het aantal problemen** dat je heb behandeld.

Mocht je genoodzaakt zijn om het spel te pauzeren, je kunt linksboven op de pauze-knop drukken.



Probleemvenster

Als je ziet dat een patiënt een probleem heeft kan je dat probleem oplossen door je verpleegkundige te slepen naar het bed van de patiënt en vervolgens op de patiënt zelf te klikken. Dit brengt het probleemvenster naar voren.

In het probleemvenster krijg je een aantal dingen te zien:

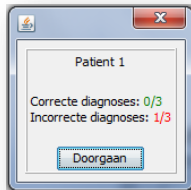
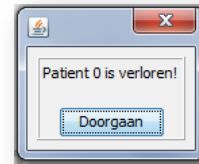
- **het patiënt-ID**, of te wel het nummer van de patiënt.
- **het aantal correct en incorrecte diagnoses**.
- **de set alarmen** die af zijn gegaan.
- **de knoppen** om de **oorzaak- en behandelingsmogelijkheden** weer te geven.

Bij elk probleem is het de bedoeling om met behulp multiple-choice zowel de meest voorkomende oorzaak als behandeling te geven. Een probleem is alleen correct beantwoord als beide vragen correct zijn (hoewel je per goed antwoord nog steeds punten verdient).

Als je een verkeerd antwoord geeft dan krijg je nog steeds het goede antwoord te zien. **Lees** voordat je je keuze maakt goed de mogelijkheden. Sommige antwoorden lijken veel op elkaar.

Als op beide vragen een antwoord is gegeven kan je het venster sluiten door op de nieuwe **return** knop te drukken.

Als je **3 keer** een **correcte** of **incorrecte diagnose** maakt krijg je na het probleemvenster een pop-up te zien op de plek van de patiënt in kwestie. In het venster zie je wat er is gebeurd.



Als je wilt kan je de status van de patiënt zien door als er geen probleem is te klikken op de patiënt waar je verpleegkundige bij staat.

Kennis

Tijdens dit programma kan je te maken krijgen met 3 verschillende alarmen:

- **Sat** – de zuurstofsaturatie van het bloed. Kan gezien worden als de effectiviteit van de ademhaling van de patiënt .
- **HR** – de hartslag.
- **BP** – de bloeddruk.

In het programma kunnen termen voorkomen waarmee je niet bekend bent. De meest voorkomende zijn:

- Drain – een item dat gebruikt wordt om wondvocht op te vangen.
- Hypoxisch – iets is niet voorzien van voldoende zuurstof.
- Inklemming – uitzetting van de hersenen, waardoor deze door de opening in de schedel naar buiten worden gedrukt.
- Inotrope middelen – zorgt ervoor dat de bloeddruk toeneemt.
- Sedatie – verlagen van de staat van het bewustzijn.
- Shunt – gebruikt om een dialysemachine aan te sluiten op de bloedbaan
- Sputum – slijm vermengd met speeksel dat vanuit de luchtwegen wordt opgehoest.
- Vagale reactie – bewusteloosheid door lage bloeddruk.

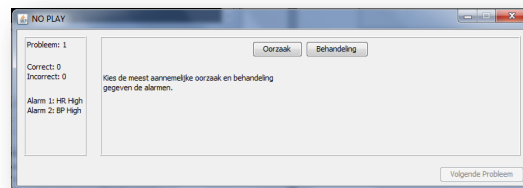
Als je dit allemaal hebt doorgelezen mag je dit document sluiten en een nieuw spel beginnen. Succes!

4.6 Instructions NOPlay

INSTRUCTIES NOPLAY

Dit zijn de instructies voor het leer- en speel-onderdeel van het experiment. Tijdens dit onderdeel is het de bedoeling te leren wat de oorzaak en gevolg van een bepaalde set alarmen zijn. Het onderdeel is afgelopen als je **50 problemen** hebt gehad of langer dan **60 minuten** hebt gespeeld. Werk zo accuraat mogelijk en neem je tijd!

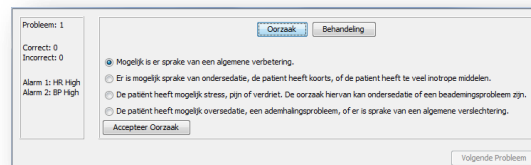
Probleemvenster



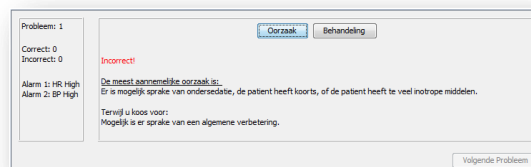
In het probleemvenster krijg je een aantal dingen te zien:

- **het huidige probleem.**
- **het aantal correct en incorrecte diagnoses.**
- **de set alarmen** die af zijn gegaan.
- **de knoppen** om de **oorzaak- en behandelingsmogelijkheden** weer te geven.

Bij elk probleem is het de bedoeling om met behulp multiple-choice zowel de meest voorkomende oorzaak als behandeling te geven. Een probleem is alleen correct beantwoord als beide vragen correct zijn.



Als je een verkeerd antwoord geeft dan krijg je nog steeds het goede antwoord te zien. **Lees** voordat je je keuze maakt goed de mogelijkheden. Sommige antwoorden lijken veel op elkaar.



The screenshot shows a medical simulation interface. On the left, a box contains the text: 'Probleem: 1', 'Correct: 0', 'Incorrect: 0', 'Alarm 1: HR High', and 'Alarm 2: BP High'. In the center, the word 'Correct!' is displayed in green. Below it, the text reads: 'De meest aanmerkelijke behandeling is: Check de patiënt.' At the top right of the main area, there are two buttons: 'Oorzaak' and 'Behandeling'. At the bottom right, there is a button labeled 'Volgende Probleem'.

Als op beide vragen een antwoord is gegeven kan je naar het volgende probleem gaan door op de **Volgende Probleem** knop te drukken.

Kennis

Tijdens dit programma kan je te maken krijgen met 3 verschillende alarmen:

- **Sat** – de zuurstofsaturatie van het bloed. Kan gezien worden als de effectiviteit van de ademhaling van de patiënt .
- **HR** – de hartslag.
- **BP** – de bloeddruk.

In het programma kunnen termen voorkomen waarmee je niet bekend bent. De meest voorkomende zijn:

- Drain – een item dat gebruikt wordt om wondvocht op te vangen.
- Hypoxisch – iets is niet voorzien van voldoende zuurstof.
- Inklemming – uitzetting van de hersenen, waardoor deze door de opening in de schedel naar buiten worden gedrukt.
- Inotrope middelen – zorgt ervoor dat de bloeddruk toeneemt.
- Sedatie – verlagen van de staat van het bewustzijn.
- Shunt – gebruikt om een dialysemachine aan te sluiten op de bloedbaan
- Sputum – slijm vermengd met speeksel dat vanuit de luchtwegen wordt opgehoest.
- Vagale reactie – bewusteloosheid door lage bloeddruk.

Als je dit allemaal hebt doorgelezen mag je dit document sluiten en een nieuw spel beginnen. Succes!

4.7 Instructions Test

INSTRUCTIES TOETS

Dit zijn de instructies voor het test gedeelte van het experiment. Tijdens dit onderdeel is het de bedoeling de kennis te testen die je (hopelijk) hebt opgedaan. Dit onderdeel bestaat uit **10 problemen** met multiple- choice vragen, waarbij je weer de meest voorkomende oorzaak en behandeling moet aangeven. Je hebt **10 minuten** voor de toets.

De antwoorden kan je invullen op het geven antwoorden formulier. Lees de problemen goed door. Succes!

4.8 Feedback Form 3

EVALUATIE FORMULIER PICPLAY

	<i>Sterk mee oneens</i>					<i>Sterk mee eens</i>
1) Ik denk dat ik dit programma graag regelmatig wil gebruiken	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
2) Ik vond het programma onnodig complex	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
3) Ik vond het programma makkelijk te gebruiken	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
4) Ik dat ik ondersteuning nodig heb van een technisch person om dit programma te kunnen gebruiken	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
5) Ik vond dat de verschillende functies in dit programma erg goed geïntegreerd zijn	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
6) Ik vond dat er teveel tegenstrijdigheden in het programma zaten	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
7) Ik kan me voorstellen dat de meeste mensen zeer snel leren om dit programma te gebruiken	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
8) Ik vond het programma erg omslachtig in gebruik	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
9) Ik voelde me erg vertrouwd met het programma	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
10) Ik moest erg veel leren voordat ik aan de gang kon gaan met dit systeem	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
11) Ik vond dat het spel concept een positieve effect had op het leren vermogen	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
	1	2	3	4	5	
12) Zijn er veranderingen, opmerkingen of suggesties voor het programma? (Graag op de achterzijde invullen)						