ZooSmart: Personalizing learning through optimizing player flow in a hybrid interactive math game

Laure Y. Smits
Department of Industrial Design
Eindhoven University of Technology
l.y.smits@student.tue.nl

Nick M. van Geenen
Department of Industrial Design
Eindhoven University of Technology
n.m.v.geenen@student.tue.nl

Wisse J.J. Raaijmakers
Department of Industrial Design
Eindhoven University of Technology
w.j.j.raaijmakers@student.tue.nl

Ivy G.C. van Dongen
Department of Industrial Design
Eindhoven University of Technology
i.g.c.v.dongen@student.tue.nl

ABSTRACT

This study aims to address the need for innovative educational games that are able to provide personal development. Current primary school education lacks the time and support to provide the needed individual support for students aged 8-11. We present ZooSmart, a hybrid innovative educational math game that aims to provide children with personalized learning through artificial intelligence. Learning algorithms are used to determine the players' flow state and help them stay in the flow of learning. Incorporating Computer Vision allows us to take a hybrid approach to educational games, creating a digital and physical part that students interact with. Play-testing showed a great potential for the game as enjoyment was high and learning was experienced as motivational. The evaluation of the personalization AI showed an accuracy of 62%, showing room for improvement but validating the ability to label flow states correctly for most of the data points. Future work needs to focus on innovating the game set-up as well as the personalisation AI. In this way, a higher accuracy can be achieved as well as better educational results.

Keywords

Artificial Intelligence, Personalized Learning, Educational Games, Computer Vision, Tangible Design, State of Flow and Explainable Artificial Intelligence (XAI)

1. INTRODUCTION

1.1 Problem Definition

Due to the constant increasing shortage of teachers in the Netherlands, the average size of classes keeps increasing . This makes it harder for teachers to give attention to every student within such a class (Ekelschot, 2020). Giving individual feedback to each student is also more difficult in this case, possibly resulting in students with remaining questions about certain topics, which were already clear for other students (Pedder, 2006).

Next to this, in the Netherlands there were already worries about the basic abilities of students, especially in the subject of math (Inspectie van Onderwijs, 2021). However due to the corona pandemic, and therefore online education, an extra delay in the development of math skills has been formed, which has not been brought back to the level before (Inspectie van Onderwijs, 2022). The Dutch Inspectorate of Education, states that the focus on the basic skills, especially math, is important in the upcoming period (Inspectie van Onderwijs, 2022).

As the shortage of teachers is not going to disappear soon, there is a need for solutions that can give the individual guidance and feedback needed for every student, when the teacher is not able to, especially for the topic of math.

To address this need, ZooSmart was created. ZooSmart is a hybrid educational game with a Zoo theme, that provides personalized learning with the use of (explainable) artificial intelligence within a class environment. It functions as an extra tool that can be used by teachers that sometimes cannot give the individual attention needed, due to the large classes, to let students learn independently in a high-quality way.

This paper first presents the related works, followed by the methods and materials, final concept, discussion and lastly a conclusion.

1.2 Related Work

This chapter presents the related works within 3 topics: Games in education, (Explainable) Artificial Intelligence in Education and Flow Theory.

1.2.1 Games in Education

Videogames are functional tools for acquiring knowledge, learning specific strategies and equipping children for the culture of the information society by introducing them to the practices of computer literacy (Gros, 2007). They are able to stimulate problem solving skills (Gros, 2007), which are required to be able to tackle math questions, and can contribute by using everyday words to describe position (McFarlane, 2002).

Serious games, designed for a specific goal instead of for entertainment only, receive increasing interest from the education field (De Gloria et al., 2014) and their effectiveness has been proven (Connoly et al., 2012). There is a want for engineering tools and methods that can achieve effective building of games and can push for effective learning experiences (Greizer et al., 2007). Generations of new serious games should use advanced technologies, such as Artificial Intelligence, and should study the right balance between educational and entertainment goals (De Gloria et al., 2014).

Fun within gamification or child-computer interaction is often seen as an essential element of learning and has a significant and positive indirect effect on both the students attitude towards the subject and learning development (Tisza, 2021).

1.2.2 (Explainable) Artificial Intelligence in Education

There has been an increase in applications of artificial intelligence within the education sector. As the system of education involves, educational experience is being enriched, complex issues are being dealt with and teaching methods are customized for individual students (Chen et al., 2020).

Knowledge discovery is the core of machine learning (Chen et al., 2020) and can help teachers gain understanding of the skill within a certain concept (Kucak et al., 2018). One of the functions artificial intelligence can have in education, surrounding the topic of learning, is that it can reveal learning shortcomings of students and address them early in education (Chen et al., 2020). As can be understood from the Introduction, this can be especially valuable with the increasing class sizes.

1.2.3 Flow Theory

The flow concept is one of the most popular constructs to express playing experiences (Procci et al., 2012) and it is argued that games are most engaging and successful if they can produce flow circumstances (Kiili, 2005). Flow can be described as a mental state in which a person is completely absorbed by a specific activity (Csikszentmihalyi, 1991). Csikszentmihalyi (2002) has introduced nine flow dimensions or mental states in terms of challenge and skill levels, including flow, anxiety and boredom. This opens up the possibilities to further explore the concept of flow states in a broader sense.

2. Methods and Materials

2.1 Approach and Method

The first step taken in defining the scope of the project was determining where we could apply AI/ML in an appropriate social context where its use would add real value to the design. Research shows that the domain of education still contains a handful of challenges that we as designers could help with (Leicht et al., 2018). By taking a closer look, we realized that especially primary schools struggle with providing and facilitating enough personal development for the students (Knauder & Koschmieder, 2019). In classes of 20-30 students it can be hard for one teacher to provide each individual student with the time and care they need in order to learn the best. We believe that we could intervene here with a design that allows teachers to better spend their time and efforts in class as well as allow students to learn in a fun way. The target audience was then defined as primary school students aged 8-11.

After the initial scoping, further explorations were done to create an initial concept of our design. Gamification has shown great potential in the context of education (Dicheva et al., 2015; Caponetto et al., 2014; Nah et al., 2014) and was therefore deemed very suitable for this project as well. Gamification allows sometimes boring topics to be presented in a way where children are more motivated to learn and achieve better results (van Roy & Zaman, 2018). Adding gamification to our concept not only makes the fundamentals of a design intervention more solid, but also opens up the possibilities for adding personalized learning through AI/ML. Taking a hybrid approach in this project made it possible to integrate a personalized learning AI but also a tangible interaction for the children to play around with. Tangible designs have shown great potential and allow children to still enjoy the benefits of a digital solution without fully immersing them into a digital learning environment where everything is done via screens (Garcia-Sanjuan et al., 2018).

We believe that our design should not take over the role of a teacher but should merely be used as a tool. By providing the teachers with insights on the students progress in the game, it would open up the possibility to dynamically direct the curriculum towards what is needed for the student and improve on the students' deficiencies in an efficient way. Moreover, the role of explainable AI could help with communicating to the student and the teacher on the in-game progress. Showing therefore the benefit and value of adding AI/ML to this concept and context.

2.1.1 The Zoo Concept

After defining the design challenge, we iterated to figure out the exact application we would be creating. We immediately had the idea of setting the game in a zoo which would allow us to cycle through the zoo with math exercises. After further exploration, we decided that a hybrid approach (physical and digital design) would be perfect if children were to take on the role of a zookeeper. In this way, children can interact with real physical objects that represent different foods that animals in the zoo need which makes the game more immersive and engaging.

ZooSmart allows students to become a zookeeper and help the other zookeepers with feeding all of the animals. The zookeeper that the students will help is called Sam. They introduce the student to every new animal and provide them with the exercises. After completing the exercises, the mascot of the zoo called Froggie (a red-eyed tree frog), will comment on the students' progress and give insight into what is being personalized for the next round of exercises.

2.1.2 The Role of AI/ML

By adding artificial intelligence to the ZooSmart concept, we open up the possibilities for personalized learning. The goal is to help each student with their individual progress in a way that makes it fun and engaging for them to practice and learn math. By incorporating a learning algorithm that dynamically changes the game based on the students progress, we aim to achieve the aforementioned goal. By approaching the concept as a hybrid set-up both a physical and digital component will be created. Creating a design that combines a dynamic personalisation AI with a Computer Vision AI, the concept would achieve its goal.

2.2 Learning Algorithm and Need of XAI

2.2.1 Computer Vision

The player interacts with the digital component of ZooSmart by placing wooden chips on the playing board. Therefore, the game requires a system to recognize these different physical objects, each object representing different input information for the game.

The decision was made to use Computer Vision (CV) as a method for recognizing the physical objects, because it would allow for the most freedom for the player to place the objects wherever they like on the playing board. Other solutions for presence detection (e.g. weight, RFID, light) would require integrating electronics into the playing board making the design not as suitable for the educational environment. In the current concept, the physical elements are kept simple to ensure children understand the purpose of each element and the game is suitable for diverse environments (e.g. classrooms). Additionally, we had no previous experience with CV and we set out to learn about working with

CV, because it is a good fit as an additional learning goal for this course.

The setup uses a webcam that is pointed at the playing board. There are different methods within CV possible to achieve the end goal of recognizing physical objects with a webcam. The initial plan was to train a neural network on images of our objects, but it was decided to revert to color-based object detection, as this was less complicated and a better fit for the scope and purpose of our concept.

The playing board is painted white, and each illustration on the objects is painted in a distinctive bright color. The system isolates these colors by masking out the defined color ranges into binary image feeds using the OpenCV python library (Bradski, 2000). From here, a border following algorithm (Suzuki & Abe, 1985) that is included in the OpenCV library, is used on these binary images to find the contours of the playing board and the different illustrations (figure 1). The system checks if the contours of the colored illustrations fall within the area of the playing board. If so, they are counted up and sent to the game when the player submits their answer. The counted amount is compared to the correct answer from the database in unity, resulting in a *correct* or *incorrect* score.

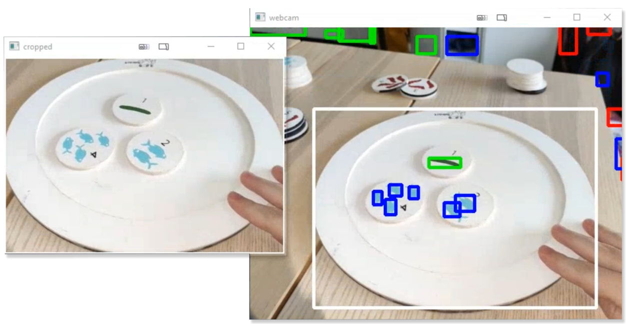


Figure 1: Contour Detection

2.2.2 Conceptualization of the personalisation AI

The related work emphasizes the role of flow theory in educational games. A clear-cut model for applying flow in design is the 'flow channel' (figure 2). It provides the designer with two variables: *challenge* and *skill*. Because this theory describes variables that can be measured and expressed in a 2D graph, it was considered to be portable to the domain of machine learning by interpreting *challenge* as the '*difficulty level*' of the exercise, and *skill* as the '*score*' of the player.

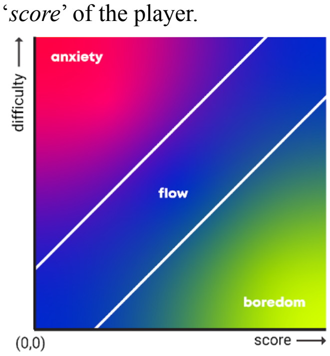


Figure 2: Flow Channel

As we were introduced to supervised learning, we determined its appropriateness for classification algorithms. By classifying the flow state of the player in either one of the three categories *boredom*, *flow*, or *anxiety*, the algorithm is able to adjust the game to get the player to the desired state of *flow.*

The concept uses a dynamic system that predicts the player's flow state at the end of each set of exercises. By calling the personalisation algorithm once every a set of three exercises, the system tries to strike a balance between accuracy and dynamic effect on the gameplay.

We hypothesize that interpreting the flow state after every single exercise could create input data that is too divergent, resulting in a low accuracy for determining the right flow state. On the other hand, the dynamic difficulty adjustment should have a noticeable effect on the game experience as it is the leading mechanic of the concept together with the tangibility.

As the learning algorithm was conceptualized, it was quickly learned that defining a score that accurately describes the player skill is a rather complex composite variable that is expressed by combining multiple variables like time and correctness. Instead of assigning a weight to these variables ourselves, it was found that the most accurate way to do this would be to let the algorithm decide this for us by incorporating the time and correctness as separate variables; resulting in the model using 3 distinct features. Through the scikit-learn library we selected the SVM (Support Vector Machine) algorithm to be used for classification because it is a well-documented algorithm that is able to achieve the task at hand (Scikit-learn, 2022). The training and application of the learning model is described in chapter 3.2.

2.2.3 The Role of XAI

We see great value in adding explainable artificial intelligence to our concept and have decided that for the children, a personification of the XAI is a suitable way to realize this. At the ages of 8-11, it might be difficult to understand XAI, so having this personification allows for a direct interaction that adds value to the students' learning process.

From the player's perspective, the main explainable AI implementation within ZooSmart is represented by the frog character called Froggie. Froggie is the personification of the Personalisation AI; it shares its perception of the player's state in a simple and relevant manner: *"That did not go so well, but don't worry; tomorrow will be a bit easier"*. They adjust their tone based on the state of the player and give insight on how the difficulty of the next set of exercises will be adjusted.

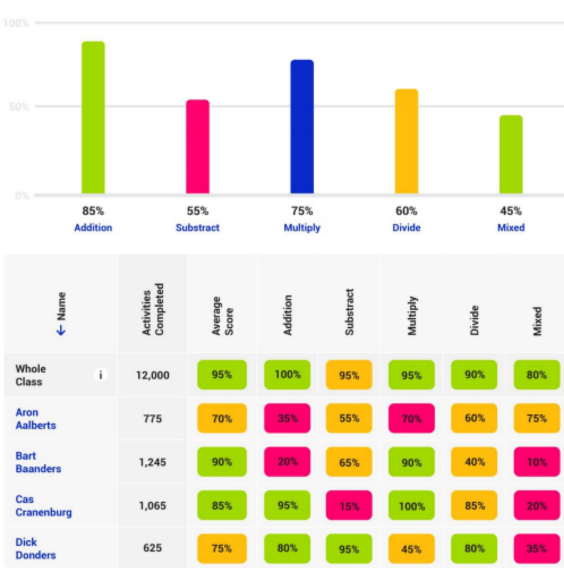
Additionally, Sam the Zookeeper is the personification of the Computer Vision (CV) system. Sam observes the work that the player is doing and controls if the answer is correct or not, giving feedback on the input they registered and the correct input for each exercise. It is crucial that the player and the CV system (Sam) are in agreement about the amount of food that will be fed to the animals.

We believe that representing these two AI-systems as characters in the game allows for insight into the actions and motivations of these intelligent systems, in a way that is understandable and relatable for the students playing the game. Froggy, who personifies the player state, is deliberately separated from Sam, who gives feedback on the correctness of the answers (figure 3). This is done with the aim of positioning Froggie as a friendly character who gives the player positive reinforcement during their learning process which can be very effective in inclusive educational contexts (Morin, 2007).

From the teachers perspective, the implementation of explainable AI would be more direct; we envision a teacher-dashboard giving the teacher insights into the progression of the math education of their students (figure 4). The personalization AI will summarize and process the raw player data in a way that provides insightful information for the teacher to use in their classes. It could show the proficiencies and deficiencies of students, but more importantly their associated flow state. Based on this, teachers are able to tailor their teaching approach to individual students and work more efficiently.

The realization and implementation of the teacher dashboard is outside of the scope of our concept and is therefore only conceptualized. However, it is an important element that is relevant to the social context that the concept is designed for, as it could allow teachers to better support their students and also reduce the amount of students that fall behind on their math development.

Figure 3: Froggie (XAI) and Sam (Computer Vision)



Name	Activities Completed	Average Score	Addition	Substract	Multiply	Divide	Mixed
Whole Class	12,000	95%	100%	95%	95%	90%	80%
Aron Aalberts	775	70%	35%	55%	70%	60%	75%
Bart Baanders	1,245	90%	20%	65%	90%	40%	10%
Cas Cranenburg	1,065	85%	95%	15%	100%	85%	20%
Dick Donders	625	75%	80%	95%	45%	80%	35%

Figure 4: Teacher Dashboard Concept

3. Final Concept

This chapter will elaborate on the final concept by describing the design of the interaction, intelligent behavior & embodiment of the concept and the testing & analysis procedure.

3.1 Design of the interaction

This section will focus on explaining the interaction and the associated scenario(s).

3.1.1 Context of Interaction

As described in the Introduction, there is a need for educational math games that can give personal challenges and guidance when the teacher is not able to do so, due to various reasons. The game is meant to function as an assisting tool for teachers working at a primary school, especially with big classes with a large variety in ability of the students. The direct target group of ZooSmart is children between the ages of 8- 11 and the indirect target group are the teachers responsible for the development of these children, as they can track the progress in the teacher interface (figure 4).

3.1.2 Scenario of Interaction

ZooSmart consists of a digital computer game and physical wooden board together with multiple wooden chips containing illustrations of cucumbers, berries and fish (figure 5). Once the game has started, the player is asked to turn on the webcam and place the feeding board within the frame. After which the first exercise appears as a series of three. In each exercise, Sam the Zookeeper asks the player to feed one of the animal species (red panda, crocodile and turtle) the correct amount of food, which is the answer of a math question in the form of a descriptive sentence. These exercises are imported from a database of math questions with corresponding answers, difficulty level (on a scale of 1 to 100) and the custom sentences. These questions are based on existing math educational modules to ensure accurate difficulty labels.

Figure 5: ZooSmart Set-Up

The player reads the question and calculates the amount of food needed off the top of his head. After which, he tries to match the correct type of food chipswith the animal and place a correct total amount of food on the feeding board. Every chip has an amount of food ranging from one to five, which adds another dimension to the math questions (figure 6). When the player is done placing the chips, the any button on the keyboard is pressed to confirm, so that the computer vision can take a snapshot and analyze the amount of food. Immediately it is communicated if the answer is correct and what the computer vision ‘sees’ on the feeding board, to take away any possible worries about the accuracy of the system.

Figure 6: Wooden Chips

This process cycles again for two more times and following is a classification of the mental state (boredom, flow or anxiety), based on the factors: time, correctness and difficulty. Within the game, Froggie (personification of XAI) then gives feedback, based on the classified perceived flow state that is understandable for the young player. Next to that the starting point for the next exercises, in difficulty, is also based on this classification (figure 7).

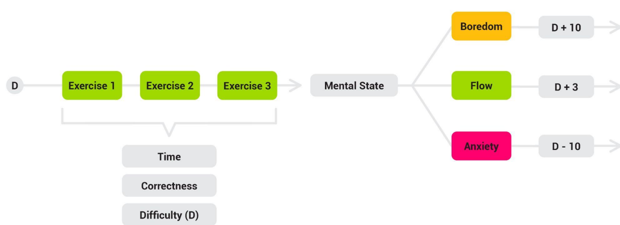


Figure 7: Visualization of Data Exchange of the Exercise Set

3.1.3 Data Exchange within Interaction

The Unity game engine is the center point of the system where the intelligent systems and interaction with the player come together.

Real-time connections are made between the game engine and Python. The CV data is constantly communicated to the game engine, and the personalisation AI is called only after each set of exercises to dynamically change the difficulty.

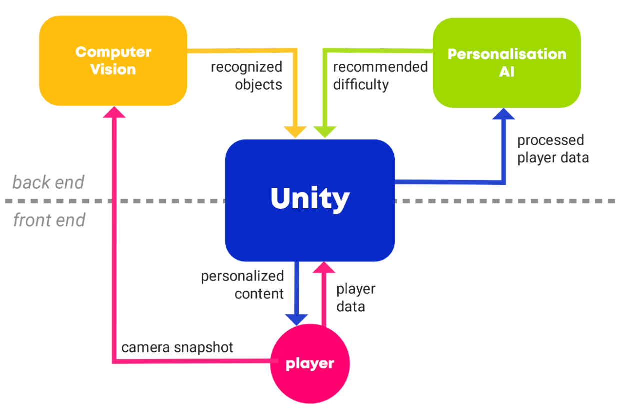


Figure 8: Visualization of Data Exchange within the Game

The game engine uses a network of scripts to manage the loading, displaying, and checking of the exercise answers. Furthermore, the game engine loops the process of the set of exercises (figure 8). At the start of each loop it uses the newly provided difficulty to display the right dialogue and animal amount (figure 9).

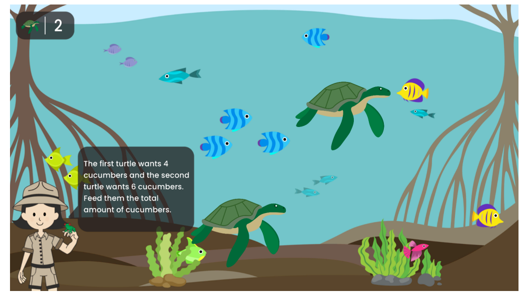


Figure 9: Dialogue Example in Game

3.2 Intelligent Behavior and Embodiment

As planned at the start of the design process, the collecting of training data was started halfway through the process, before the fully playable game was realized. In parallel to the game development, the physical artifacts and the dataset of exercises was finished. With these two components at hand the researchers were able to individually simulate the game, and collect labeled data as follows:

1. Start stopwatch
2. Read Sam's math question as a sentence
3. Place the right amount of food items on the board
4. Manually asses the correctness of the answer
5. Repeat step 2,3,4 for all three exercises
6. Stop stopwatch
7. Record perceived flow state

In total, data on 100 exercise sets was collected (example in figure 10. Figure 11 and 12 visualize the training data as a table and in 3D space respectively. This data was preprocessed by standardizing all data and imported in Python as a CSV file and used for training the SVM algorithm using the code attached in Appendix 7.2.

Difficulty (1-100)	Time (0-180)	Correctness (0-3)	Label (0-2)
10	21	3	0
20	29	3	1
23	25	3	1

Figure 10: Example Data from Training Data

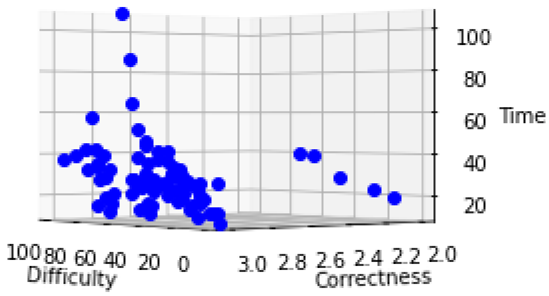


Figure 11: 3D Visualization of Training Data

The SVM algorithm finds the optimal separating hyperplane to divide the data points in two categories. It does this by maximizing the distance between the hyperplane and the two categories. In 3D space, this hyperplane is a 2D plane.

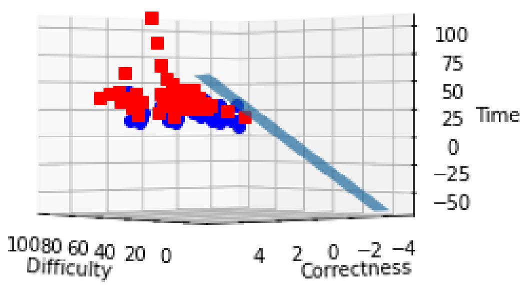


Figure 12: 3D Classification Visualization

The labeled data had a distribution of 38 instances labeled *boredom*, 48 labeled *flow*, and 14 labeled *anxiety*. The classifications of this data are better shown in 2D plots. Figures 13-15 show the mutual relation between the *difficulty*, *time*, and *correctness* respectively. In all of these plots, the algorithm classifies these data points in three colored categories: *anxiety* as red, *flow* as gray, and *boredom* as blue.

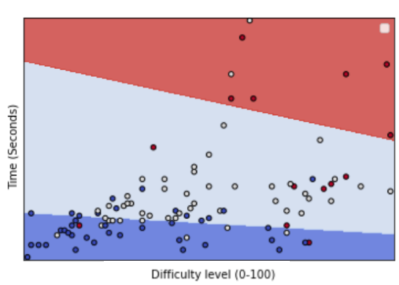


Figure 13: 2D Classification Visualization: Time and Difficulty

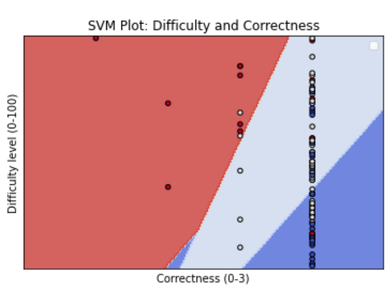


Figure 14: 2D Classification visualization: Difficulty and Correctness

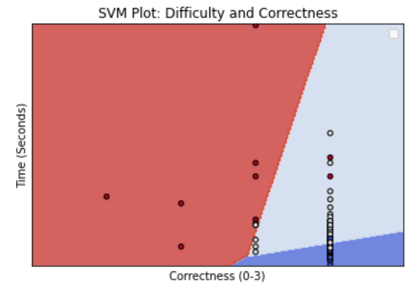


Figure 15: 2D Classification visualization: Time and Correctness

This trained SVM algorithm is able to predict the classification of new instances by calculating on what side of the hyperplanes the position of the new value lies.

The algorithm runs in a Python script that is in turn called by a different script that manages all communication with the Unity game engine. When the player completes a set of exercises, the current difficulty level, time, and correctness are sent to the algorithm. From this data the algorithm predicts the current flow state of the player which is used to return the new difficulty level from Python to Unity.

3.3 Testing and Results

Figure 16: Play Testing

After the game was finished, data to be used to evaluate the personalized learning algorithm was gathered during play-testing (example in figure 16). Play-testing was also used to gain insights into the experience of the game and identify potential directions for future work. The data was collected on a sample of 3 students from the Eindhoven University of Technology, taking turns to individually play the game. The participants were asked to express themselves as they played (e.g. talk-aloud method), and were subsequently asked questions to reflect on their experience of the game. The interpretations of the play tests are mentioned in the discussion chapter 4.

For collecting testing data, the exact *difficulty, time, and correctness* of each exercise set that the participants played was logged by the game. The researchers observed the flow state of the participant and generated labels for the logged data in real time.

3.3.1 Evaluation of Personalisation AI

Unfortunately the data collected during play testing was not sufficient to be used for evaluating the personalisation AI. The quantity of data was not enough to adhere to the predetermined goal of having a training-data to test-data ratio of 2:1, as well as the quality of the data was not reliable because technical difficulties prevented an accurate labeling of the player's flow state.

As a more reliable alternative, the researchers resulted in simulating the testing data the same way that the training data was collected, as described in chapter 3.2. In the course of one hour of simulating the game, an evaluation dataset of 50 labeled data points was collected.

With this dataset, the learning algorithm was evaluated by determining the accuracy and creating a confusion matrix (figure 17). Out of 50 instances, our previously trained model correctly classified 31 instances, resulting in an accuracy of 62%.

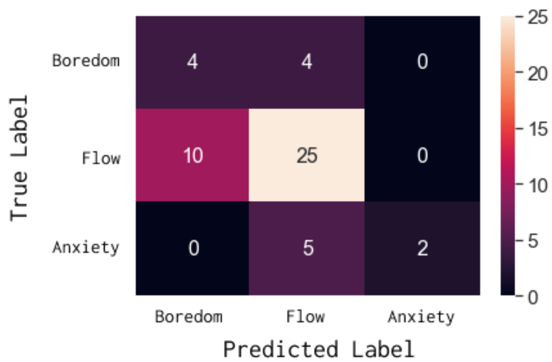


Figure 17: Confusion Matrix

3.3.2 Evaluation of Interaction

During the aforementioned playtest sessions, the tangible interaction with intelligent systems of ZooSmart was evaluated. Players experienced a very high enjoyment of the game due to its design; all participants expressed their enjoyment of the embodied interaction in combination with math equations. The participants adopted different techniques of solving the equations using the wooden chips; sometimes searching for chips that match the numbers given in the equation, other times solving the equation first and gathering the chips together afterwards. Additionally, positive comments were made on the in-game graphics and design of the tangible elements. Despite the game being educational, because of the tangible interaction, they saw it as a fun experience and did not immediately associate it with traditional learning.

However, the interaction with the intelligent systems integrated within ZooSmart was unsatisfactory for the participants. They experienced problems with getting the system to recognize the intended amount of wooden chips on the playing board, resulting in frustration with the interaction itself, as well as improper data collection for the personalisation AI to tailor the difficulty to the player. They were continuously engaged when solving math equations, but expressed distrust in the system whenever it was time to submit the answer.

This makes it challenging to evaluate the interaction as a whole; the embodied interaction with the intelligent system showed promise, as it received positive feedback across all participants. However, the compounded effect of maintaining the flow state through both personalisation AI and a tangible interaction has not been properly evaluated, due to the technological limitations.

4. Discussion

ZooSmart was designed to help primary school students (aged 8-11) with learning math. Through a personalized learning algorithm, the game aimed to provide the students with exercises that keep them in the flow of learning. The game featured both a physical component, making the learning experience hybrid and tangible. The goal of the concept is to help students in their learning by adopting a learning algorithm and XAI.

4.1 Interpretation of the Results

The evaluation of the interaction with ZooSmart showed that the participants were positive about the tangible design of ZooSmart. The playful context of the game combined with the tangible interaction made for a learning experience that was described as engaging. ZooSmart positions these math problems in a realistic environment, supporting situated cognition. The affordance of the tangible interaction allows for the player to apply several strategies to solve the math problems, which makes the exercises more dynamic than conventional math exercises.

Evaluation of the learning algorithm shows an accuracy of 61%, which is not enough for a good game experience as explained in the evaluation of the interaction in chapter 3.3.2.

Looking at the training data it was already observed that there is an unequal distribution; Most exercise sets were labeled *flow* or *boredom*, with very little being labeled *anxiety*. This led the algorithm to be insufficiently trained to correctly classify instances where the player is in anxiety. This is reflected in the confusion matrix, where 6 out of 50 instances were labeled *anxiety*, of which only 2 were correctly labeled by the trained algorithm. Designers should be careful of misclassification that confuses or harms a child that is feeling anxious.

In the confusion matrix it can also be observed that there is a relatively high amount of 10 instances where the true label *flow* was given the predicted label *boredom*. This can be explained by the difference in the training dataset and the testing dataset, as the training dataset has 38 out of 100 (38%) instances labeled *boredom*, the testing dataset only has 8 out of 50 (16%) instances labeled *boredom*. The difference in the two datasets is expected to be due to the unequal contexts in which the two data collection sessions were done. When playing the game to collect the testing data, the researchers were more experienced with doing math exercises.

4.2 Limitations

The first limitations of this study were experienced during the play-testing sessions. Before the sessions, the game was thoroughly tested and the Computer Vision (CV) seemed to have a high accuracy. However, during the play-testing sessions, this seemed to be more inaccurate than expected. Several times the right amount of food was laid on the board, but the computer vision did not recognize this correctly. We believe that this is due to changing lighting conditions; during the play-testing there was little sunlight, causing the colors captured by the webcam to be more dimmed than before, resulting in less distinct color ranges that we were able to define. For example, the green color of the cucumber was less saturated during the play-test, but when adapting the color recognition range to the change in color, it resulted in the system also recognizing black borders as green. The computer vision had a smaller window of suitable lighting conditions than initially expected to recognize the distinct features of the objects. Because of this, we were unable to determine a meaningful accuracy of the AI, beyond the observation that the current accuracy is not suitable for further deployment of the prototype.

This unfortunately did cause for the players to get a bit irritated with the game, which influenced their enjoyment, but also the effectiveness of the personalisation AI. Because of the inaccuracy with the object detection, perceptions of the player's flow state could not be recorded.

During play-testing, observations were also made about the set-up of the game. ZooSmart has a lot of different components with the intent to make the game more fun and interactive. However, during the set-up of the play-testing it was noticed by the researchers that this also could cause lots of problems. Because of the amount of different components, there is a bigger chance that things go wrong with setting them up correctly. This caused the actual play-testing to take less time than intended, as more time was needed to correctly set it up.

Because the data for both the training- and evaluation datasets were generated ourselves. The variability of players who imputed the data was not very high. Because there was a low variability, the personalisation AI would be likely to perform worse for players with a different level of math skill. More work needs to be done in order to assess the result of this and see if and how the flow model was influenced by a lower variability of player data.

4.3 Future Work

From the aforementioned points, several suggestions for future work can be made in order to improve the overall game and the experience thereof.

More work needs to be done for the game to be deployed in real social settings (e.g. classrooms, meeting rooms, etc). Real educational settings were outside the scope of this project but by exploring this further and improving the game upon it, better results might be achieved in-game. Next to this, the set-up needs to be simplified in order to facilitate a better experience in these different environments. Right now there are too many components and not setting up one of them correctly can have a big impact on the enjoyment and accuracy of the game.

The scope of this project also does not include testing with children, so future work should focus on evaluating the game and its setup with them as well. Although each component of the game was created keeping the child in mind, several rounds of player testing need to happen in order to confirm these assumptions. Exercises were also created based on a traditional assessment of the level of math for every grade but real-life testing should focus on validating this as well. It is important to create a good interaction between the child and the game as it will not only influence their enjoyment but also their progress and results in this course. Therefore it is crucial that the next steps will involve the target audience (e.g. primary school students aged 8-11) and improvements will be made upon their experiences.

Towards the end of the project, a teachers dashboard was also designed in order to implement the explainable AI in a suitable way. This was only conceptualized during this project, so future work needs to be more done on the experience, effect and efficiency of this add-on platform. Assumptions were made on what the teachers would want to be informed upon based on the students progress in the game, but player testing will need to show whether this is correct or needs to be improved upon. Next to this, future work can focus on the right implementation of the explainable AI. Right now, we believe that this dashboard is the right way of letting the AI communicate their actions and behaviors to the teachers, but several other explorations can be done to find potential other and better ways to do this. Within this exploration, also the type of useful information can be looked into as not all information from the game is useful to the teachers. By providing them with the right information, they can focus their efforts even better on helping the students in class.

As mentioned before, there are limitations to the learning algorithm of the personalisation AI. To improve the learning model, cross-validation could be used to train and test the model on different parts of the total body of collected data.

For the personalisation AI, more variability in datasets need to be added for a more accurate estimation of the state of flow. Right now, a dataset of 100 data points was used to train the personalisation AI which included data of four different players. Future work should focus on gathering more diverse data from various players, all with different levels of math and/or different backgrounds. This would improve the personalisation AI to become better at predicting the flow state of edge cases, which is especially important to prevent misclassifying player's that are in *anxiety*.

On top of this, future work should also look into the different features used to determine the state of flow. Right now three variables are used to determine the state of flow; time, correctness and difficulty. However, we suggest there might be more things that influence the state of flow but this exploration fell outside of the scope of this project. We suggest looking into how this larger set of features influences flow. This could make the game experience better as well and also potentially improve the results as the game is better in accurately predicting the state of flow.

The other implementation of AI within this project is the computer vision which is used to recognize the correct food type and amount of food. For future work, we suggest looking into improving accuracy, because right it is very dependable on a small window of suitable lighting conditions. The reliability of the system needs to be improved upon in order to facilitate a better game experience, because otherwise the computer vision recognizes the wrong amount on the board, whilst the player actually was correct. Right now the system is fully dependent on detecting contours in the defined color ranges. By implementing more ways of recognizing the correct amount of food on the playing board, the system can cross-reference several measurements against each other and determine the correct amount more accurately. This adds redundancy and removes the dependence on correct lighting conditions, which would likely increase the accuracy of the system.

5. Conclusions

The aim of this project was to apply AI/ML in an appropriate way to a specific social context. Early on, we defined this social context within the domain of education, focussing specifically on helping primary school students aged 8-11 with math. Our goal was to help alleviate some of the pressure experienced by the teacher associated with personal support and development of the students. Teachers in primary school often have a lot of students to take care of at the same time, making it very time-consuming and tiresome. For this we designed ZooSmart, a hybrid interactive math game designed to support personalized learning by using AI.

During play-testing we noticed a high enjoyment in the game by participants, commenting on the immersiveness of the game and how that helped make learning more engaging. Unfortunately, the accuracy of the computer vision was not very high during play-testing as mentioned before. This did lead to several instances of frustration where players got the right answers but the game did not recognize it. We are confident that with future development, the game can be optimized and be readied for the intended target group.

The evaluation of the personalization AI resulted in an accuracy of 62%. We noticed that the edge cases were very hard to classify by the system, especially the *anxiety* state was labeled wrong a few times. We do note that especially with students of that age, it is very important to be careful with classifying them as anxious as this could have more implications than intended. The right classification of flow state can however be improved by increasing the variability of the dataset in which the AI is trained. Moreover, classifying the flow state is not as straightforward in practice. In the scope of this project, three variables were used to determine the flow state but in practice there could be even more variables to be used for this. Further testing needs to be executed to determine a more suitable and accurate mix of variables to be used to determine the flow state. However, within the scope of this project, the three chosen variables (e.g. time, difficulty and correctness) were deemed suitable for the cause.

The explainable AI that was implemented in ZooSmart had a goal of sharing the information about students' progress to the students themselves and the teacher in an appropriate and useful way. We conclude that this is indeed achieved by the creation of Froggie and the teachers dashboard. The needs of the two stakeholders were evaluated and used to design these XAI interactions. The play-testing partly validated this, but future development needs to be done to optimize it.

Overall, the results of our play-testing and the evaluation of the personalization AI makes us confident that ZooSmart indeed achieves the project goals set as intended. Even though there is definitely room for improvements, the foundation is there. ZooSmart now still has very basic features that could be elaborated upon in the future but concerning the goal of making studying math more fun, engaging and personalized, it is definitely succeeding. Hopefully a product like this could indeed alleviate some of the pressure experienced by primary school teachers. We did not aim to take away something from the educational system, like replacing the role of the teacher, but merely providing a solution that could make learning more fun. We believe that ZooSmart can be the right tool to turn this into a reality.

6. REFERENCES

Bradski, G. (2000). *The OpenCV Library. Dr. Dobb's Journal of Software Tools.*

Caponetto, I., Earp, J., & Ott, M. (2014). *Gamification and education: A literature review.* In the European Conference on Games Based Learning (Vol. 1, p. 50). Academic Conferences International Limited.

Chen, L., Chen, P. & Lin, Z. (2019). *Artificial Intelligence in Education: A Review.* IEEE Access, 8, 75264–75278. https://doi.org/10.1109/ACCESS.2020.2988510.

Connolly, T. M., Boyle, E. A., MacArthur, E. & Boyle, J. M. (2012). *A systematic literature review of empirical evidence on computer games and serious games.* Computer & Education, *59*(2), 661–686. https://doi.org/10.1016/j.compedu.2012.03.004.

Csikszentmihalyi, M. (1991). *Flow: The Psychology of Optimal Experience*. Harper Perennial.

Csikszentmihalyi, M. (2002). *Flow: The Psychology of Optimal Experience* (2de editie). Harper & Row.

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). *Gamification in education: A systematic mapping study. Journal of educational technology & society,* 18(3), 75-88.

Ekelschot, J. (2021). *De invloed van het lerarentekort op de relatie tussen de SES compositie van de schoolklas en de leesprestaties van leerlingen.* https://studenttheses.uu.nl/handle/20.500.12932/39346

Garcia-Sanjuan, F., Jurdi, S., Jaen, J., & Nacher, V. (2018). *Evaluating a tactile and a tangible multi-tablet gamified quiz system for collaborative learning in primary education.* Computers & Education, 123, 65-84.

de Gloria, A., Bellotti, F., & Berta, R. (2014). Serious Games for education and training. *International Journal of Serious Games*, *1*(1). https://doi.org/10.17083/ijsg.v1i1.11

Greitzer, Frank & Kuchar, Olga & Huston, Kristy. (2007). Cognitive science implications for enhancing training effectiveness in a serious gaming context. *ACM Journal of Educational Resources in Computing.* 7. https://doi.org/10.1145/1281320.1281322.

Gros, B. (2007). Digital games in education: The design of games-based learning environments. *Journal of research on technology in education*, *40*(1), 23-38. https://doi.org/10.1080/15391523.2007.10782494

Inspectie van het Onderwijs (2021). *Binnen zonder kloppen. Digitale weerbaarheid in het hoger onderwijs*. Utrecht: Inspectie van het Onderwijs

Inspectie van het Onderwijs (2022). *Rapport De Staat van het Onderwijs 2022.* Utrecht: Inspectie van het Onderwijs

Kiili, K. (2005). Content creation challenges and flow experience in educational games: The IT-Emperor case. *The Internet and Higher Education*, *8*(3), 183–198. https://doi.org/10.1016/j.iheduc.2005.06.001

Knauder, H., & Koschmieder, C. (2019). *Individualized student support in primary school teaching: A review of influencing factors using the Theory of Planned Behavior (TPB).* Teaching and Teacher Education, 77, 66-76. https://doi.org/10.1016/j.tate.2018.09.012

Kucak, D., Juricic, V. & Dambic, G. (2018). Machine Learning in Education - a Survey of Current Research Trends. *Proceedings of the 29th International DAAAM Symposium 2018*, 0406–0410. https://doi.org/10.2507/29th.daaam.proceedings.059

Leicht, A., Heiss, J., & Byun, W. J. (2018). *Issues and trends in education for sustainable development* (Vol. 5). UNESCO publishing.

McFarlane, A., Sparrowhawk, A., & Heald, Y: (2002). *Report on the educational use of games*. Retrieved from http://www.teem.org.uk

Morin, D. (2017). *The effects of inclusion and positive reinforcement within the classroom.*

Nah, F. F. H., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., & Eschenbrenner, B. (2014). *Gamification of education: a review of literature.* In International conference on HCI in business (pp. 401-409). Springer, Cham.

Pedder, D. (2006). *Are small classes better? Understanding relationships between class size, classroom processes and pupils' learning.* Oxford Review of Education, 32(2), 213–234. https://doi.org/10.1080/03054980600645396

Procci, K., Singer, A. R., Levy, K. R. & Bowers, C. (2012). Measuring the flow experience of gamers: An evaluation of the DFS-2. *Computers in Human Behavior*, *28*(6), 2306–2312. https://doi.org/10.1016/j.chb.2012.06.039

Van Roy, R., & Zaman, B. (2018). *Need-supporting gamification in education: An assessment of motivational effects over time.* Computers & Education, 127, 283-297.

Scikit-learn: *machine learning in Python — scikit-learn 1.1.3 documentation. (n.d.).* Retrieved November 4, 2022, from https://scikit-learn.org/stable/

Suzuki, S. & Abe, K. (1985). *Topological structural analysis of digitized binary images by border following*. Computer Vision, Graphics, and Image Processing, 30(1), 32–46. https://doi.org/10.1016/0734-189x(85)90016-7

Tisza, G. (2021). The role of fun in learning. *Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play*. https://doi.org/10.1145/3450337.3483513

7. Appendix

Code of your program, links, link to the video and other technical details if necessary to enable the reader to check and eventually reproduce your prototype or behavior.

7.1 Link to video

ZooSmart Main Video: https://youtu.be/aBva59y7aaU

ZooSmart in Action: https://youtu.be/VHI8x-of3mI

7.2 Personalization AI - code

```python
"""
Created on Thu Oct  6 11:48:56 2022

@author: 20191808
"""

from sklearn import svm
import csv
import numpy

labels = []
ml_data = []

#import the list of time, correctness, and difficulty
with open('Playdata.csv', newline='') as csvfile:
    ml_data_temp = csv.reader(csvfile, delimiter=',', quotechar='|')
    for row in ml_data_temp:

        count = 0
        for i in row:
            row[count] = int(i)
            count = count + 1

        labels.append(row[3])
        row.pop(3)
        ml_data.append(row)

#predict difficulty
def run_svm(time: int, correctness: int, difficulty: int) -> int:

    intervention_threshold = 10
    flow_increase = 3

    clf = svm.SVC()
    X = ml_data
    y = labels
    clf.fit(X, y)

    predictionArray = clf.predict([ [correctness, difficulty, time] ])
    prediction = predictionArray[0]

    if prediction == 0:
        return_difficulty = difficulty + intervention_threshold
        print("boredom: increasing the difficulty with ", intervention_threshold)

    elif prediction == 1:
        return_difficulty = difficulty + flow_increase
        print("flow: increasing the difficulty with ", flow_increase)

    elif prediction == 2:
        return_difficulty = difficulty - intervention_threshold
        print("anxiety: decreasing the difficulty with ", intervention_threshold)

    return return_difficulty
```

7.3 Computer Vision - code

```python
# Standard imports
import cv2;
import numpy as np;

#QUICK CALIBRATION -- kan ik gemakkelijk de kleurenrange aanpassen afhankelijk van verlichtingsomstandigheden
greenH = 60
blueH = 190 / 2

#storing webcam footage
video = cv2.VideoCapture(1)

while True:
    succes, img = video.read()
    #cv2.imshow("board", img)

    ellipse = 0

    #BOARD DETECTION
    #Masking out playing board based on color (white)
    hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    boardLower = np.array([0,0,160])
    boardUpper = np.array([179,18,255])
    boardMask = cv2.inRange(hsv_img, boardLower, boardUpper)

    #Checking for contours of white objects with minimum area of 50000 to detect board
    boardCountours, hierarchy = cv2.findContours(boardMask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    if len(boardCountours) != 0:
        for contour in boardCountours:
            if cv2.contourArea(contour) > 10000:
                x, y, w, h, = cv2.boundingRect(contour)
                ellipse = cv2.fitEllipse(contour)

    #Creating circular mask
    if ellipse == 0:
        print("board not found")
    else:
        mask = np.zeros((img.shape[:2]), dtype="uint8")                    # creating mask
        cv2.ellipse(mask,ellipse,(255,255,255),-10)                       # drawing ellipse
        masked_img = cv2.bitwise_and(img, img, mask=mask)                 # applying mask on img

        #FOOD DETECTION
        masked_img_hsv = cv2.cvtColor(masked_img, cv2.COLOR_BGR2HSV)      # Converting masked img i

        #Masking colors of food types into binary images
        #Green
        greenLower = np.array([greenH - 30, 30,  20 ])
        greenUpper = np.array([greenH + 20, 255, 220])
        greenMask = cv2.inRange(masked_img_hsv, greenLower, greenUpper)

        #Red
        redLower = np.array([160 , 65, 20 ])
        redUpper = np.array([180 , 255, 255])
        redMask1 = cv2.inRange(masked_img_hsv, redLower, redUpper)
        redLower = np.array([0 , 125, 20 ])
        redUpper = np.array([5 , 255, 255])
        redMask2 = cv2.inRange(masked_img_hsv, redLower, redUpper)
        redMask = redMask1 + redMask2

        #Blue
        blueLower = np.array([blueH-20, 30,  20 ])
        blueUpper = np.array([blueH+30, 230, 255])
        blueMask = cv2.inRange(masked_img_hsv, blueLower, blueUpper)

        #Resetting Food counter
        p = 0; c = 0; t = 0;

        #Finding Red Berries
        redContours, hierarchy = cv2.findContours(redMask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        if len(redContours) != 0:
            for contour in redContours:
                if cv2.contourArea(contour) > 80:                          #For each contour large
                    x, y, w, h, = cv2.boundingRect(contour)
                    cv2.rectangle(masked_img, (x,y), (x + w, y + h), (0, 0, 255), 3) #And draw it on the webcam
                    p = (p+1)

                #Finding Blue Fishies
        blueContours, hierarchy = cv2.findContours(blueMask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        if len(blueContours) != 0:
            for contour in blueContours:
                if cv2.contourArea(contour) > 80:
                    x, y, w, h, = cv2.boundingRect(contour)
                    cv2.rectangle(masked_img, (x,y), (x + w, y + h), (255, 0, 0), 3)
                    c = (c+1)

        #Finding Green Cucumbers
        greenContours, hierarchy = cv2.findContours(greenMask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        if len(greenContours) != 0:
            for contour in greenContours:
                if cv2.contourArea(contour) > 80:
                    x, y, w, h, = cv2.boundingRect(contour)
                    cv2.rectangle(masked_img, (x,y), (x + w, y + h), (0, 255, 0), 3)
                    t = (t+1)

        #final string
        totalCV = f"{p}p{c}c{t}t"
        print(totalCV)

        #Close windows with keypress Q
        key = cv2.waitKey(1)
        if key == ord('q'):
            break

video.release()
cv2.destroyAllWindows()
```