CODE2DRIVE: A CODE-BASED INTERACTIVE AND EDUCATIONAL DRIVING ENVIRONMENT FOR IMPROVING THE YOUTH DRIVING LEARNING AND TRAINING USING MACHINE LEARNING

Zihao Lin[1], Jonathan Sahagun[2]

[1]Lower Merion High School, 315 E Montgomery Ave, Ardmore, PA 19003

[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Serious games are video games designed for more than just pure entertainment purposes [1]. Serious games developers combine traditional game mechanics and the ideas to educate, inform and facilitate social change [2]. These games can be used in many occasions, such as education, healthcare and more. Serious games use simulations and scenarios to provide an immersive and interactive learning experience. They offer an environment to experiment with a variety of solutions to real-world problems, promoting critical thinking and decision making skills. These games can also improve knowledge retention, motivation, and engagement, as they provide instant feedback, rewards, and challenges. This application is like one of the many serious games, it provides a simulation of a highway, its primary purpose is to help to train juvenile's knowledge on driving and logical thinking, and relax during the playthrough [3].

KEYWORDS

Educational, Serious game, Coding, Driving

1. INTRODUCTION

Recent years have witnessed the dramatic popularity of software engineering careers, due to the rapid advancement of technology, which are expected to grow by 25% from 2021 to 2031, will cause a high demand on the software engineering employers [4]. Even though the increase in demand for software engineering is likely to have several positive effects on the industry such as more job opportunities, higher salaries, greater emphasis on education and training, more diversity in the workforce, and increased innovation. As the industry continues to evolve, due to the increasing demand for software engineering, it's most likely to cause more companies competing for the same skill level of software engineers, there will be a huge increase of demand for top talent. This will result in companies offering more attractive salaries, benefits, and perks to attract and retain the best talent. It might cause an increased stress level on old software engineers and the inexperienced young engineer as a consequence of the increase of competition in the field. It will be important for software engineers to stay up-to-date with the latest technologies and trends to remain competitive in the field, or for young potential engineers to start to get into more fields of software engineering [5].

David C. Wyld et al. (Eds): CMLA, GRAPH-HOC, CIoT, DMSE, ArIT, WiMoNe, CSEIT – 2023
pp. 159-168, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.1310014

There are some techniques or systems that have been designed to help the kids to have a kick start and a growth of interest in software engineering. In particular, Scratch is an excellent example [6]. Scratch is a visual programming language and website that employs high-level block-based coding to serve as an educational tool primarily targeted at children aged 8 to 16. Users can design their desired program using pre-coded blocks. Users on the site, called Scratchers, can create projects on the website using a block-like interface. Projects can be exported to standalone HTML5, Android apps, Bundle (macOS) and EXE files using external tools [7][9]. The service is developed by the MIT Media Lab, has been translated into 70+ languages, and is used in most parts of the world [8]. Scratch is taught and used in after-school centers, schools, and colleges, as well as other public knowledge institutions. However, the sufficiency of freedom is the most crucial problem of its design, new users will usually be overwhelmed by the numerous options, if the user couldn't find the result they are looking for, they might soon lose the interest of continuing learning as a programmer. Also, dependency on an internet connection is another major flaw of this design, since Scratch is an online tool, due to this factor, the users must have an internet connection to use it. This can be a barrier for users who live in areas with limited internet access or who prefer to work offline. Beside those problems, Scratch is not commonly used in professional programming settings, which can limit the real-world application and practical value of learning the language.

Code 2 Drive takes a similar yet different approach, although both designs are block based coding tools and design with the similar top priority of to teach the youth logical thinking , but unlike the overwhelming variety Scratch offers, Code 2 Drive takes a much simpler way, it’s just around a big idea of safe driving. The straightforward topic might immediately catch the user's attention and keep it interesting especially for the children that are into cars driving. Also, as an offline application, this means that it doesn't require the user to have an internet connection to use it, they can use it wherever and whenever they want, whether the users are on a plane, in a park, or simply without Wi-Fi. It also doesn’t require any sign up or account creation, making the whole thing easier to use. The most important part is that this application not just trains the youth’s logical thinking, it also connects with some of the real world problems. For example, the problem that is used in this application is safe driving, each time the user would like to take a turn, not just they have to remember to look around, also they need to remember to hit the turn light, using this application will create a good habit.

In two application scenarios, we demonstrate how the above combination of techniques increases the thinking behavior of users. First, we show the usefulness of our approach by a comprehensive case study on the evolution of logical thinking. Second, we analyze the evolution of knowledge on safe driving. We let 16 people with no experience at all during this one week experiment. According to the experiment, the results showed that the intervention group had a statistically significant decrease in the frequency of risky driving behaviors compared to the control group. The intervention group showed around a 42% decrease in risky driving behaviors compared to the previous behavior, while the control group showed little to almost no changes. The results also showed that there was no significant difference in demographics between the two groups. There are also significant improvements on logical thinking as well, the results showed that the intervention group had a statistically significant improvement in logical thinking skills compared to the control group. The intervention group showed a 20% increase in logical thinking skills, while the control group showed no change. Overall, the results of this study suggest that using the application can be effective in improving logical thinking skills and safe driving awareness. Regardless of the demographics of the participants. These results have implications for educators and individuals who are interested in improving their logical thinking skills.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Stay focus

During the process of learning, people may experience a variety of cognitive, motivational or affective challenges that affect their learning, they might not be able to focus on one certain thing for some time before slowly losing its interest. Or they might leave everything to wait to finish later. Stressful situations can also make it challenging to concentrate and focus on tasks. Anxiety can also make it challenging to focus by causing your mind to constantly wander. Large tasks can be overwhelming and lead to loss of focus, but by breaking them down into smaller and more manageable tasks, and trying to focus on completing one task at a time. This techniques in this design and it could help user to improve their focus levels and eliminate distractions.

2.2. Reading, writing or spelling skills

During the process of learning, reading, writing or spelling skills are playing a huge factor on it. For the most of the time, if someone has difficulty reading, they may struggle to understand basic instructions for assignments or tests, which can impact their ability to complete the task accurately or affect the efficiency. They may struggle to extract meaning from texts, making it challenging to learn new information. When someone is struggling with reading difficulty, it is important to provide them with appropriate support and intervention. This may include reading intervention programs, assistive technology, or individualized instruction. In this application, most instruction are given by most basic learning material - simple animation, reading, writing or spelling difficulty or just genuinely a very young age will have almost no effect on the this learning process.

2.3. Stress and anxiety

Stress and anxiety also play a big role on the path of learning, whether it’s caused by lack of rest or sufficient amount of work, they might all cause you to lose your patience and focus [10]. According to science study, research has demonstrated that the secretion of stress hormones can impact both short-term and long-term memory recall abilities. A study conducted by Sandra Ackermann at the University of Basel found that participants who had elevated levels of cortisol during a recall activity had greater difficulty recalling specific events. During high-pressure situations such as exams and presentations, students frequently experience what is commonly referred to as "mind blanks," wherein they have difficulty recalling specific information. Moreover, the presence of stress hormones can alter stored information, as evidenced by past research conducted by Elizabeth Loftus, who showed how misinformation and the creation of false memories can lead to confusion and the recall of inaccurate academic information for students.. In this application, the simple texture and the easy instruction and game play could significantly reduce the user’s stress and anxiety, to help them keep their focus longer.

3. SOLUTION

Code 2 Drive is a game-based technology to create interactive learning environments. It incorporates elements such as simulation, role-playing, decision-making, and problem-solving to engage users in learning activities. It personalized and adaptive learning experiences, and enhancing retention of information. The first step in creating a serious game is to design the game mechanics, and learning objectives. The game is designed to engage users and provide a fun and interactive learning experience, it has a top priority of upskilling the user's way of logical thinking and the knowledge on safe driving. With those in mind, after finishing designing the basic material, I started to develop the learning part. As the users use this application, their progress is tracked and assessed to ensure they are meeting the learning objectives. The application may also provide feedback and guidance to help users improve their skills and knowledge, this ensures that users are receiving a personalized learning experience that meets their individual needs. After users complete each level, whether they succeed or not, the program will provide immediate feedback to users, which helps them understand their strengths and weaknesses or mistakes and identify areas where they need to improve. In summary, while providing a fun and interactive learning experience that engages users in learning activities. Code 2 Drive is also designed to meet specific learning objectives and incorporate adaptive learning technology to provide a personalized experience. As users experience the game, their progress is tracked and appraised, and feedback is provided to help them improve their own skills and knowledge.

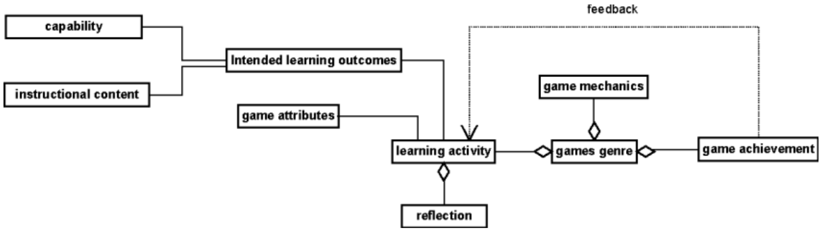
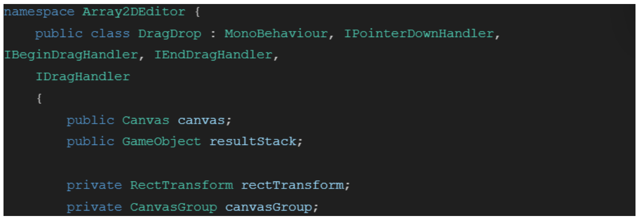
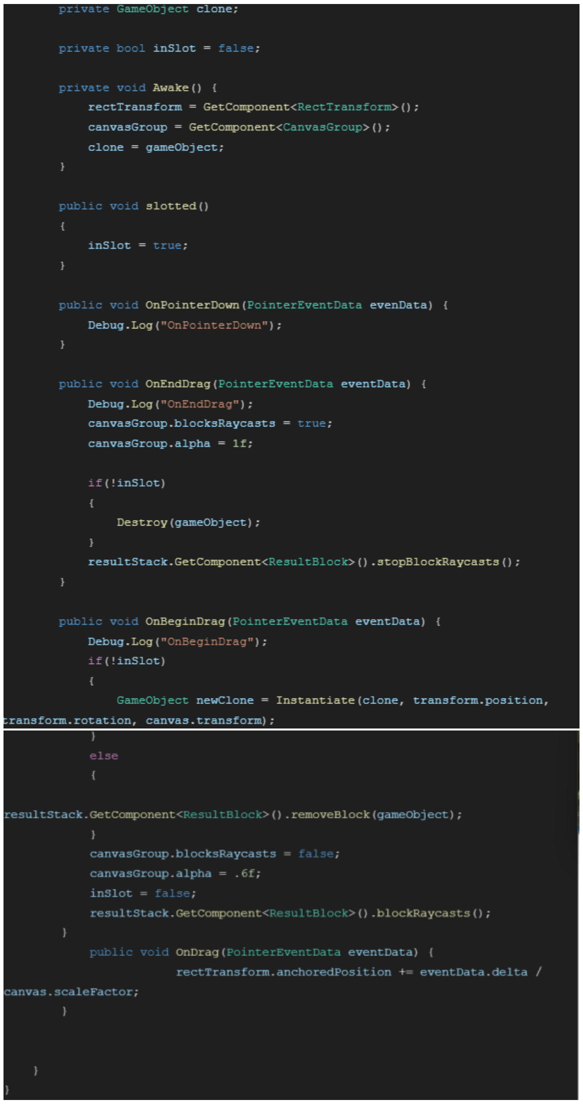


Figure 1. Overview of the solution



```
namespace Array2DEditor {
    public class DragDrop : MonoBehaviour, IPointerDownHandler,
IBeginDragHandler, IEndDragHandler,
    IDragHandler
    {
        public Canvas canvas;
        public GameObject resultStack;

        private RectTransform rectTransform;
        private CanvasGroup canvasGroup;
```



```
        private GameObject clone;

        private bool inSlot = false;

        private void Awake() {
            rectTransform = GetComponent<RectTransform>();
            canvasGroup = GetComponent<CanvasGroup>();
            clone = gameObject;
        }

        public void slotted()
        {
            inSlot = true;
        }

        public void OnPointerDown(PointerEventData evenData) {
            Debug.Log("OnPointerDown");
        }

        public void OnEndDrag(PointerEventData eventData) {
            Debug.Log("OnEndDrag");
            canvasGroup.blocksRaycasts = true;
            canvasGroup.alpha = 1f;

            if(!inSlot)
            {
                Destroy(gameObject);
            }
            resultStack.GetComponent<ResultBlock>().stopBlockRaycasts();
        }

        public void OnBeginDrag(PointerEventData eventData) {
            Debug.Log("OnBeginDrag");
            if(!inSlot)
            {
                GameObject newClone = Instantiate(clone, transform.position,
transform.rotation, canvas.transform);
            }
            else
            {

resultStack.GetComponent<ResultBlock>().removeBlock(gameObject);
            }
            canvasGroup.blocksRaycasts = false;
            canvasGroup.alpha = .6f;
            inSlot = false;
            resultStack.GetComponent<ResultBlock>().blockRaycasts();
        }
            public void OnDrag(PointerEventData eventData) {
                        rectTransform.anchoredPosition += eventData.delta /
canvas.scaleFactor;
        }


    }
}
```

Figure 2. Screenshot of the game 1

First I worked on the drag drop script, this is the key component of the whole program, without it, the users will not be able to have access to control the player car, this code here give all the action blocks the property to be able to drag and drop the pre coded blocks and control the car. With this finished, then I began to work on the other action blocks. Each block is assigned with a different property such as go forward and backward and more. After the blocks were done I started to work on a program to give me easy access to create a grid, which serves a purpose of helping to create the spawn position of each type of car and the goal for each level. After finishing those frameworks, it is time to work on a different type of car. In Code 2 Drive cars are pretty much only serving a purpose of roadblock or obstacle, but I want it to be more than that, therefore, In Code 2 Drive there are three different types of obstacles, the first one is the most basic one, just simply blocking one block of path. The second one is a truck, it will block two blocks of path, it will have no response to the player's turning light. The last one is the most interesting one, the model of it is a sports car, whenever there is no obstacle block in front of it, it will have a block forward as the user takes an action. With all those factors, and more than one path to the finishing goal, the users are most likely to find a different solution each time they play, with all the unpredictable moves Code 2 Drive will create an infinite amount of probability.

4. EXPERIMENT

4.1. Experiment 1

In this experiment I am testing how this program will affect the user's way of logical thinking. First I find around 12 participants with different ages and absolutely no experience with the topic, while trying to control the outside factor and keeping track of their progress and time spent. Then dividing the participants into 2 teams, after 1 week of experimenting, then I will compare the result with the other group and with different programs to find the best result.

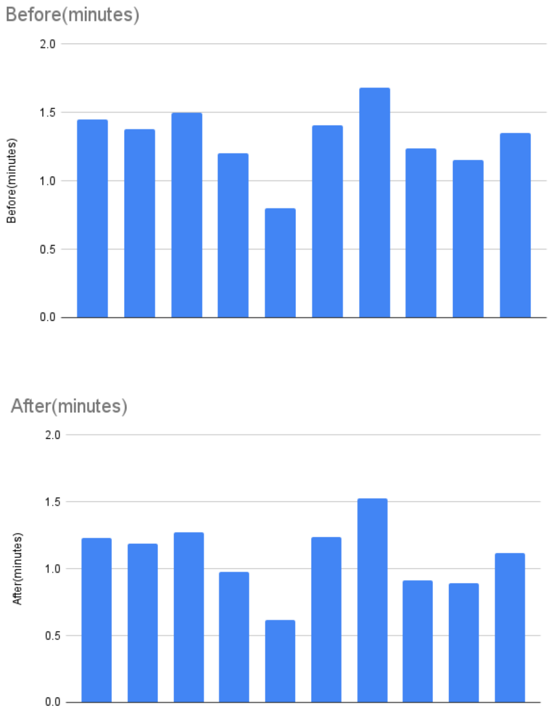


Figure 3. Figure of experiment 1

This is a comparison of the user’ s time spend on a similar level, according to the research, after the end of the experience, almost all the participants shows an significant decrease of time spend on the similar level design, each with an average of 20% improvement on the time spend, which shows the experiment result on the logical thinking. Also, we find out that participants with a younger age will have a more successful result on the improvement compared to the more mutual participants. Also, according to the experience result on other similar applications, which only shows an average increase of about 12% improvement on time spent. In conclusion, Code 2 Drive is a much better tool to use to help to train logical thinking, especially to young users, Code 2 Drive shows a significantly better result compared to some of the other tools.

4.2. Experiment 2

In this experiment I am testing how this program will affect the user's way of Safe driving. First I find around 12 participants with different ages and absolutely no experience with the topic, while trying to control the outside factor and keeping track of their progress and time spent. Then dividing the participants into 2 teams, after 1 week of experimenting, then I will compare the result with the other group and with different programs to find the best result.

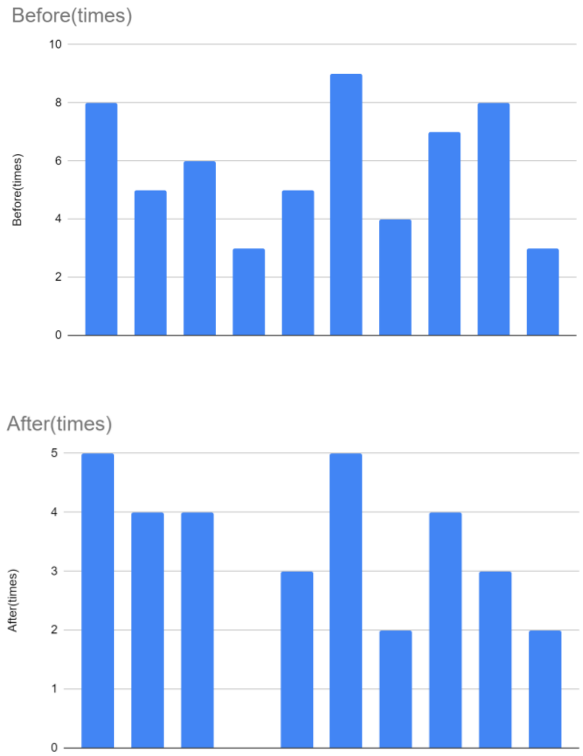


Figure 4. Figure of experiment 2

This is a comparison of the amount of the user’s mistake made on a similar level, according to the research, after the end of the experience, almost all the participants shows an significant decrease of mistake they made on the similar level design, each with an average of 23% improvement on the time spend, which shows the experiment result on the logical thinking. But different from the other experiment, this time participants with an older age showed much better performance compared to the younger participants. Also, according to the experience result on other similar applications, which only shows an average increase of about 19% improvement on time spent. In conclusion, Code 2 Drive is a slightly better tool to use to help to train logical thinking, especially to older users, Code 2 Drive shows a significantly better result compared to some of the other tools.

In conclusion, as the result of the experiments, both of them have a better result compared to other similar applications. With an 8% higher improvement on logical thinking and a 4% higher improvement on safe driving both prove that Code 2 Drive is a much better tool when it comes to studying than others. Code 2 Drive is not only good for a certain age too, according to the experiments, users with a young age will have a significantly better improvement on logical thinking, and the users with an older age will have an outstanding improvement on safe driving. Overall, the results of the experiments suggest that Code 2 Drive is a promising tool for improving both logical thinking and safe driving skills. Further research and user testing could help to refine and improve the application for even better results in the future.

5. Related Work

Provide the definition of a serious game and the purpose for it [11]. “A serious game or applied game is a game designed for a primary purpose other than pure entertainment. The "serious" adjective is generally prepended to refer to video games used by industries like defense, education, scientific exploration, health care, emergency management, city planning, engineering, politics and art.”

“Serious games are video games designed for more than just pure entertainment purposes. Serious games developers combine traditional game mechanics and the ideas to educate, inform and facilitate social change.” In my version of work, I shorten the unnecessary information to make the overall review to look more clean visually.

Provide the definition of Scratch and its design usage and user target [12]. “Scratch is a high-level block-based visual programming language and website aimed primarily at children as an educational tool, with a target audience of ages 8 to 16” “Scratch is a visual programming language and website that employs high-level block-based coding to serve as an educational tool primarily targeted at children aged 8 to 16. Users can design their desired program using pre-coded blocks.” In my version of work, I keep the straightforward explanation of the program while further explaining how it functions.

Use to learn further knowledge and research on how stress levels affect someone’s learning ability [13]. “For students, this relates to the common term ‘mind blanks’ that affects many students during high-pressure situations including examinations and presentations.” “During high-pressure situations such as exams and presentations, students frequently experience what is commonly referred to as "mind blanks," wherein they have difficulty recalling specific information.” In my version of work, I explain the definition of the term “mine blinks” while explaining the effect of this term.

6. Conclusions

In conclusion, with an initial idea of developing an application targeting logical thinking learning and safe driving practice for users at all ages, with those ideas in mind, I began to work on this project. To create a successful serious game, the initial step involves designing the game mechanics and learning objectives [15]. The game should engage users in a fun and interactive manner with a focus on improving logical thinking and knowledge of safe driving. After designing the basic material, the next step involves developing the learning component, which tracks user progress and provides personalized feedback and guidance to meet their individual learning needs. Upon completing each level, the program provides immediate feedback to users to help them identify their strengths and weaknesses, as well as areas for improvement. After the basic framework is finished, it begins to enter the experimenting stage, where I find 12 participants with different ages and no experience with the topic to test its logical thinking training and safe driving capability. After one week of experimenting, the results shows that the application have significantly better improvement on logical thinking and safe driving compare to competitive tools, with a better improvement on logical thinking on younger users, to a greater improvement on safe driving on older users, prove that Code 2 Drive is one useful and trustworthy tools to use to help improve those problems.

Even though the overall design is pretty much flawless; it serves a good purpose and decent result on the experiments. There are still some limitations in the application, for example, there isn’t a background music that fits with the theme, although this does not affect the overall learning performance from this application, however, it will not be as easy to keep focus and interest in this if the users are not attracted to the topic. It also has some small problems like limited levels, but issues like that are easy to fix.

In the future, I’m looking forward to collaborate with friends to make a background music that fit with the game environment, also, I will add more level into the applicant and perhaps using AI to generate the level or allow the users to customize their own unique level [14].

REFERENCES

[1] Granic, Isabela, Adam Lobel, and Rutger CME Engels. "The benefits of playing video games." American psychologist 69.1 (2014): 66.

[2] Readman, Jeff, and Andrew Grantham. "Shopping for Buyers of Product Development Expertise:: How Video Games Developers Stay Ahead." European Management Journal 24.4 (2006): 256-269.

[3] Driver, Christine, and Eve M. Brank. "Juveniles' knowledge of the court process: Results from instruction from an electronic source." Behavioral sciences & the law 27.4 (2009): 627-642.

[4] Karunasekera, Shanika, and Kunal Bedse. "Preparing software engineering graduates for an industry career." 20th Conference on Software Engineering Education & Training (CSEET'07). IEEE, 2007.

[5] Glass, Robert L., Iris Vessey, and Venkataraman Ramesh. "Research in software engineering: an analysis of the literature." Information and Software technology 44.8 (2002): 491-506.

[6] Maloney, John, et al. "The scratch programming language and environment." ACM Transactions on Computing Education (TOCE) 10.4 (2010): 1-15.

[7] Hoy, Matthew B. "HTML5: a new standard for the Web." Medical reference services quarterly 30.1 (2011): 50-55.

[8] La Spina, Martina, et al. "MiT/TFE family of transcription factors: an evolutionary perspective." Frontiers in cell and developmental biology 8 (2021): 609683.

[9] Fang, Hanming, and Peter Norman. "To bundle or not to bundle." The RAND Journal of Economics 37.4 (2006): 946-963.

[10] Bystritsky, Alexander, and David Kronemyer. "Stress and anxiety: counterpart elements of the stress/anxiety complex." Psychiatric Clinics 37.4 (2014): 489-518.

[11] Alvarez, Julian, and Damien Djaouti. "An introduction to Serious game Definitions and concepts." Serious games & simulation for risks management 11.1 (2011): 11-15.

[12] Ouahbi, Ibrahim, et al. "Learning basic programming concepts by creating games with scratch programming environment." Procedia-Social and Behavioral Sciences 191 (2015): 1479-1482.

[13] Amit, Z. A. L. M. A. N., and Z. HARRY Galina. "Stress-induced analgesia: adaptive pain suppression." Physiological reviews 66.4 (1986): 1091-1120.

[14] Holzinger, Andreas, et al. "Causability and explainability of artificial intelligence in medicine." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9.4 (2019): e1312.

[15] Whyte, Elisabeth M., Joshua M. Smyth, and K. Suzanne Scherf. "Designing serious game interventions for individuals with autism." Journal of autism and developmental disorders 45 (2015): 3820-3831.