COGNITIVE REHABILITATION BASED PERSONALIZED SOLUTION FOR DEMENTIA PATIENTS USING REINFORCEMENT LEARNING

Rathnayaka M.H.K.R.

IT16067370

BSc (Hons) in Information Technology
Specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

COGNITIVE REHABILITATION BASED PERSONALIZED SOLUTION FOR DEMENTIA PATIENTS USING REINFORCEMENT LEARNING

Rathnayaka M.H.K.R.

IT16067370

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

August 2020

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ……………………………. Date: ……………………...

(Rathnayaka M.H.K.R.)

The above candidate has carried out research for the B.Sc. Special (Hons) degree in SE Dissertation under my supervision.

Signature of the Supervisor: …………………………. Date: …………………...

(Dr. Dharshana Kasthurirathna)

Signature of Co-supervisor: ……………………… Date: …………………….

(Ms. Thilini Jayalath)

DEDICATION

This thesis is humbly dedicated to all the people who contributed so much and helped me to accomplish this work.

To my beloved parents, brother, sister and friends for their immeasurable support and for serving as strong force behind all my endeavors and give me inspiration to make this research possible.

And Specially to my supervisor, co-supervisor and external supervisor, who guided me to complete this research.

And finally, for all the Dementia patients who needed helping hand for evening of their lives.

ABSTRACT

Dementia is one of the most challenging health problems faced globally with the increase in the ageing population. The estimated current prevalence of dementia is 47.5 million worldwide. This number will nearly double in every 20 years globally [1]. Dementia is basically, a syndrome which cannot be cured by medicine [2], but non-pharmacological therapy can be used to treat Dementia patients, this is known as Cognitive Rehabilitation Therapy. According to the recommendations of the doctors, the use of a brain training application could be better than traditional approaches. There are number of Brain training mobile applications in the world that could be useful in improving human concentration, attention and all sorts of brain activities but there isn't any customized software solution that has games or activities. Patients can be in different stages of Dementia. So, for better cognitive rehabilitation in Attention and Concentration they need the personalized therapies with the games. Accordingly, developing this application is an actual global requirement for dementia patients who needs to improve their Attention and Concentration skills.

The world is evolving with new technologies and this application includes the games based on such technologies as Reinforcement Learning which predict the next level for patients based on user behavior. Patients, caregivers and doctors can view the score and the progress reports. All the levels of the game have designed along with the supervision and recommendation from a Consultant Psychiatrist in Sri Lanka. The main objective is to help the Dementia patients in cognitive rehabilitation to improve the quality of life with best suited personalized game for Attention and Concentration.

Key Words: Dementia, Reinforcement Learning, Cognitive Rehabilitation, Personalized, Attention, Concentration

ACKNOWLEDGEMENT

On the very outset of this report, I would like to express my sincere and heartfelt obligation towards all the personages who helped me in this endeavor. Without their active guidance, help, cooperation and encouragement, I would not have made headway in the project.

Foremost, I would like to express my sincere gratitude to our project supervisor Dr. Dharshana Kasthurirathna for providing his invaluable guidance, feedback and his valuable time throughout the course of our research project for complete success. Moreover, I would like to thank our lecture in-charge, Dr. Janaka Wijekoon and CDAP team for providing the guidance to reach the successful ending of the project. Also, I would specially thank Dr. Chathurie Suraweera(Senior Consultant at National Hospital, Colombo/ Senior Lecturer in Department of Psychiatry, Faculty of Medicine, University of Colombo) for spending her valuable time in sharing the medical knowledge and all the relevant information throughout each and every phase of the project. And I extend my gratitude to Mr.J.A.D.T. Jayawickrama (Lecturer, Department of Industrial Management, University of Kelaniya) for helping hand to the success of this project. Finally, I would like to express my special thanks to my group members, friends who supported me throughout this project and my parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
RL	Reinforcement Learning
MMSE	Mini Mental State Examination
SLDC	Software Development Life Cycle

1.INTRODUCTION

Dementia is a disease which describes a collection of symptoms such as decrease the ability to think and remember, emotional problems, problems with language, behavior and the ability to perform everyday tasks. This caused due to damage in the brain cells. This disease is affected enough to interfere with a person's normal social or working life. And this is a long-term disease that can be diagnosed from the Cognitive testing or Mini-Mental State examination. Since there is no special medicine and the Treatment that can be given is supportive care. So, this has a significant effect on the caregivers of the patients. Attention and concentration take prominent place from the mentioned symptoms, which results in being distracted easily, difficulties in focusing, thinking or the processing takes longer than usual.

Attention is compound mental process that cannot be shorten to one simple definition or one concrete anatomical structure. And it cannot be evaluating by one simple test since it encompasses various processes. To simplify, attention is the brain function or the cognitive function that we use to choose between stimuli that reach our brain at the same time, both internal and external, that are useful for carry out either a motor or mental activity. It is a complete set of processes that differ in intricacy and allow us to carry out the other remain cognitive functions as well.

With the current technologies, the new proposed system includes the game which can help the patients to increase the cognitive function required to maintain some level of Attention and Concentration. Carrying out simple tasks will be increased in complexity, will be introduced to the patients with the game. The game will be very simple at the initial stage and level by the level game will be started to change. And finally, up to a considerable level with continuous monitoring with the caregiver and the doctor.

This game will be implemented using reinforcement learning, which helps to make the game more specific to each patient. The most important of the application is to follow the advancement during the time that they are spending at home. While the patient is away from the doctor, the patient can increase the attention and concentration from their own by using this application. This will be an incredible assistant for the patients since there is not a suitable cure for Dementia.

1.1 Background

Dementia describes a collection of symptoms that caused due to the disorders affecting the brain. It influences thinking, conduct and the capacity to perform regular undertakings. And, this influence can meddle with the individual's ordinary social or working life. The vast majority with dementia is more seasoned, however, it is essential to recollect that not every single more established individual get dementia. Dementia can transpire; however, it is progressively basis after the age of 65 years. Individuals in their 40s and 50s also can have dementia. The early indications of dementia are extremely unpretentious and obscure and may not be quickly self-evident. Some basic manifestations may include:

- Progressive and frequent memory loss
- Confusion
- Personality change
- Apathy and withdrawal
- Reduce the ability to perform everyday tasks.

When we are looking back the history of Dementia, many decades ago, first diagnosed Dementia patient was found in Germany. During the late 1890s, the common symptoms in Dementia such as loss of memory, no sense of time and place, delusions were shown. After many years later she became completely demented. She died in 1906 April. After passing nearly century her case was examined again with the modern medical technologies and found the reason for the disease.[3] Since then the different types of researches were done to find a solution for the Dementia problem, but still, there is no specific medication for this syndrome.[2] In any case, much can be offered to help and improve the lives of individuals with dementia and their cares and families.

1.2 Literature Review

1.2.1 Dementia

Dementia is the greatest worldwide challenge for health and social care in the 21st century. It happens to the people older than 65 years, so increments in numbers and expenses are driven, around the world, by expanded life span coming about because of the welcome decrease in individuals passing on rashly. The Lancet Commission on Dementia Prevention, Intervention, and Care met to unite the tremendous steps that have been made and the developing information concerning what we ought to do to forestall and oversee dementia. Comprehensively, around 47 million individuals were living with dementia in 2015, and this number is anticipated to significantly increase by 2050. Dementia influences the people with the condition, who continuously lose their capacities, just as their family members and different supporters, who need to adapt to seeing a relative or companion become sick and decrease, while reacting to their requirements, for example, expanding reliance and changes in conduct. Furthermore, it influences the more extensive society since individuals with dementia additionally require wellbeing and social consideration. The 2015 worldwide expense of dementia was evaluated to be US$818 billion, and this figure will keep on expanding as the quantity of individuals with dementia rises. About 85% of expenses are identified with family and social, as opposed to clinical, care. It may be that new clinical consideration later, including general wellbeing measures, could supplant and conceivably lessen a portion of this expense.[4]

1.2.2 Mobile Applications for Dementia patients

In 21^{st} Century, everybody is running on exchange plant of life. Right now, rivalry, Dementia/Alzheimer can be tousled result of such upsetting life. Alzheimer is most basic sort of dementia. For the most part it occurs at an age of 65 or more noteworthy.[5] In our nation, dealing with patients who have been experiencing this infection is troublesome. Larger part of them have been admitted to the psychological clinic by their relative or they must be detained at home. With the assistance of this application, we have attempted to fill the hole among patient and guardians. The thought behind the execution of it on "Android" is, these days android is generally acknowledged and open source working framework. Lion's share of helpful devices is being bolstered by it like Tablet, Wristwatch, Cell Phone and so forth. This application has a few essential usefulness like "GPS Navigator", "Fall Detection System", "Brain Games", "Specialist Finder" and "Crisis". With the help of clinical experts like researcher and scientist's future improvement can be conceivable.

1.2.4. Reinforcement Learning for Dementia

Right now, informatics, it has gotten foremost to give customized proposals to moderate the impacts of data over-burden. This area of biomedical and medicinal services informatics is yet undiscovered most definitely. The vast majority of the current recommender frameworks have, somewhat, not had the option to address sparsity of information and non-linearity of client thing connections among different issues. Profound fortification learning frameworks can upset the proposal structures as a result of its capacity to utilize non-straight changes, portrayal learning, grouping displaying and adaptability for the usage of Wellbeing industry, for the most part, manages long haul issues. Customary recommender frameworks neglect to consider the long-haul impacts, henceforth neglecting to catch dynamic opinions of individuals. This methodology regards the procedure of proposal as a successive choice procedure, which addresses the previously mentioned issues. It is evaluated that more than 700 million individuals will have wearable gadgets that will screen each progression they take.[6] Information gathered with these savvy gadgets joined with different sources like Electronic Health Records, Nutrition Data and information gathered from studies can be prepared to utilize Big Data Analysis apparatuses and took care of two suggestion frameworks to create alluring proposals. The activity, state pair is taken care of to the pundit arrange, which produces a prize related to the activity, state pair. This prize is utilized to refresh the arrangement of the Actor organizes. The pundit arrange picks up utilizing a pre-characterized Expected Reward.

1.2.5 Different techniques of Reinforcement Learning

Reinforcement Learning is one of the subcategories from the Machine Learning, where the agent learns from the feedbacks of the try and error to get the next step through a prediction. Reinforcement can be used for different purposes and games take prominent place among them. In order to win a game, the player needs to have a good strategy. So, to produce a good strategy, the player needs to play the game multiple times which consuming time, energy and money. As a solution with the help of reinforcement learning agent in the game and after running multiple simulations of the game can come up with new good improving results after each iteration. Humans can improve the winning chances by imitating the reinforcement learning agent’s performance. Implementation of the reinforcement learning can be done in various ways [7]. Among them Q-Learning and State-Action-Reward-State-Action (SARSA) are two methods have almost similar aspects other than Q-learning is Off-policy while SARSA is on-policy algorithm.

1.2.6 Q-learning

Q-Learning is one of the model free reinforcement learning algorithms. And this algorithm is based on Bellman Equation. In this algorithm the learning is based on the action taken from another policy while including in the off-policy category. The following displaying the Bellman equation.

$$v(s) = E[R_{t+1} + \lambda v(S_{t+1})|S_t = s]$$

The letter ‘E’ referring to an exception and discount factor is referred by ‘ג’. The mail goal of Q learning is maximizing the Q value. In order to do that, first need to understand the two value updates that is necessary to recognize, first policy iteration and the value iteration. Policy iteration includes the policy evaluation and policy improvement that runs in a loop.

This is the best way to find the ideal procedure which considers all the things of the estimations and simplifying the findings [7]. This information has high cost because of the measure of data that should be removed although it has benefits. Q-learning is an investigation insensitive and any activity can be completed at any time. And the data is picked up from this experience.

1.2.7 Reinforcement learning for games

Reinforcement Learning and games have a long-term good mutual favorable ordinary history. Games are very challenging and rich realm for reinforcement learning when testing algorithms. Other than that, in many games the best players are the computer players which use the reinforcement learning. The reinforcement learning algorithms are infrequently adequate for high level game plays without doing any modifications. So, it is necessary to talk about the additional ideas and suggestions, the new methods of inserting the realm knowledge and implementation decisions are needed for the scaling up process of the algorithms. These are reviewed with insufficient details in order to identify their prospective and their limitations [8]. There are lists of challengers also for the reinforcement learning in games. This listing contains the game centric viewpoint and there are some items that are unique to games. And the reinforcement learning can also applicable in game development in a way commercializing the games.

1.2.8 Rehabilitation with games

People play games while learning as it provides enjoyment, and they continue playing. In addition to that intrinsic presence of competitiveness and confrontation in whichever game might be censorious in social behavior. And it is powerful mechanism for individual development and growth. So, games might fulfill various objectives as strengthening the educational programs and facilitating the recovery with the time from various pathologies. Game-based rehabilitation processes are more successful because of its more engaging nature [9]. The theories behind the artificial intelligence systems and machine learning, that indicating the repetitive task-oriented implementation in un-supervised scenario, which imitate the learning by doing. In motor re-training present a valid and viable method for rehabilitation. The other way of conveying the rehabilitation is obtained by virtual teacher or real teacher, because the triggering of an instance is learning by imitation scheme. If the rehabilitation exercises are supervised within a game, giving the meaning the player to select a solution in the matter of movement execution out with possible number discreate replacements. This can be hypothesizing as example for reinforcement learning scheme. Among the machine learning approaches the reinforcement learning is one of the best where a strategy is decided by the entity which can maximize the reward function as the output of the actions. The agent is the controller in the machine learning definitions. It’s not clear which actions needed to be taken since it’s not told. But it learns to transform the policy based on the rewards obtained. The algorithms have the ability to provide this way of trainings are at your fingertips. The mechanism also self-learning the about the behavioral analyses of the games. Simple reinforcement learning schemes demonstrate as the well-organized in considering two factors. First is law of the effect that the alternatives carrying the positive effects that tend to be repeated in the future. Second is the power law of practice where performing the natural system while facing the unseen conditions trying to increase proportionally with the decreasing rate with the time.

1.3 Research Gap

According to the diagnosis contexts Dementia is decreasing and losing of cognitive functions, that go along with many extensive and complex range of neurodegenerative diseases. The cognitive reserve which is the knowledge rederived through one's continues learning play a significant role in the appearance of dementia. This helps brain to alter its neural activities to cooperate with the challenge cause by the new play and learn situation. Cognitive brain training or training the untapped cognitive functions gives new the ability to the brain that needs in the rethink and rewire its neutral circuits into new ones.

There existing mobile applications that contains games and activities that can help to rethink and rewire the neutral circuits into new ones and help the dementia patients. But these applications are not customized ones. Dementia has three main levels as mild, moderate and severe. For patients in different levels has the same games and activities in the existing applications to improve the attention and concentration. The novelty of the implemented system is the ability of customizing the game according to the users by using Reinforcement Learning technologies. And most of the application are only in English language. It creates language barrier for the users. Specially for the local patients. So, the new system supports the Sinhala language also in the necessary interfaces.

In fact, moving on with mobile games which helps Dementia patients to improve Attention and Concentration, take through about research gap is most vital for forthcoming researches. While in a conversation about mobile applications that were introduced for Dementia patients and the proposed system, the newly implemented system supports many features which can be mainly suited for elderly population of Dementia rather existing applications.

Table 1 : Comparison between existing applications and proposed system

Features	Lumosity	Brain Games	Elevate	Proposed System
For both Attention and Concentration	✓		✓	✓
Suitable speed for game	✓			✓
Separate progress indicators for Attention and Concentration		✓		✓
Suitable colors and fonts for interfaces	✓			✓
Sent reports to doctor			✓	✓

1.4 Research Problem

Dementia is one of the most significant problems facing with the increase in the ageing population. The estimated current prevalence of dementia is 47.5 million worldwide [1]. This number will nearly double in every 20 years globally [1]. As well as the same problem has occurred in Sri Lanka. Dementia is basically a syndrome which can't be cured by medicines. In any case there are medications that help to slow down the movement of the sickness. There are medications just as mental and behavioral treatments that help. But a patient could be performed that treatments during their treatment sessions only. And, doctors can't get the history of treatments of each patient regularly. According to the requirements of the doctors, the use of a brain training application could be better than medicine. There are number of brain training mobile applications in the world that could be useful to develop human concentration, attention and all sorts of brain activities but there isn't suitable mobile application for Sri Lankan culture because most of the dementia patients in Sri Lanka belong to the old generation and mostly they are not proficiency with the English language and also the available current applications are not free even at the first stages and the users must pay a fee for using those applications. And, the activities in the existing application are with similar colors which difficult to identify separately, small fonts which difficult to read and the speed also very high in a way as healthier person can't even play. In some application instructions are not clear enough to follow. According to the resources, developing this application is an actual global requirement for dementia patients.

1.5. Research Objectives

1.5.1 Main Objectives

Due to the demographic trends, the growth of the ageing population rising dramatically in Sri Lanka as well as in the world. As a result, the number of Dementia patients will be increased gradually. But there isn't any medication for curing Dementia. So cognitive rehabilitation or non-pharmacological therapy is a good way to help Dementia patients. Considering about that, the main objective is to give rehabilitation to the patients in a cognitive way from a personalized application using reinforcement theories of the ML.
The proposed game has an ability to overcome those issues. It has game with several levels to improve the cognitive functionalities for the attention and concentration for Frontal lobe of the Cerebral cortex. This game consists of different levels. Reinforcement Learning is used in this game in order to predict the best level for the users. This game is designed

with user friendly interfaces specially for the elder people and in a way suitable for our culture. And the patients can view their progress with the score level as well as the doctor of the relevant patient via a report.

1.5.2 Specific Objectives

- Implement a game to help to improve the Attention and Concentration level of patients.
- Make the customized the best suited levels for each patient in the game using RL.
- Give the patient daily and weekly progress of the Attention and Concentration levels.
- Update the caregiver as well as the doctor with weekly reports via emails.
- Make both patients and caregivers life easier.
- Slow down the moving from one stage to another stage of Dementia.

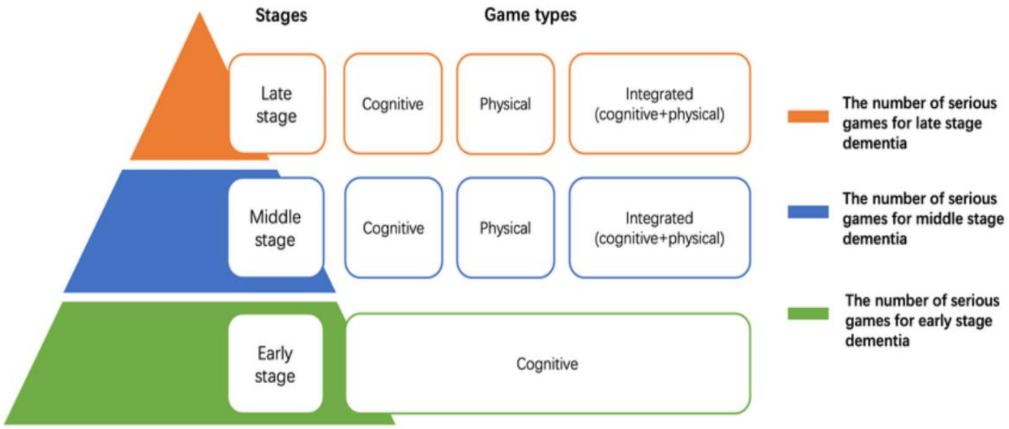

Stages
Game types
Late stage
Cognitive
Physical
Integrated (cognitive+physical)
Middle stage
Cognitive
Physical
Integrated (cognitive+physical)
Early stage
Cognitive
The number of serious games for late stage dementia
The number of serious games for middle stage dementia
The number of serious games for early stage dementia

Figure 1: Stages of Dementia

2. METHODOLOGY

2.1 Methodology

2.1.1 System overview diagram

The doctor let the patient to do the MMSE and according to the results of it the doctor identify Dementia stage of the patient. If the patient in Mild or Moderate stage the doctor will recommend the implemented application to the patient. Referring to the Figure 2, the caregiver can set up the application for patient by creating account, providing all the needed patient and doctor information into system. This information will display in the patient's profile and also some needed information will be included in the reports as well.

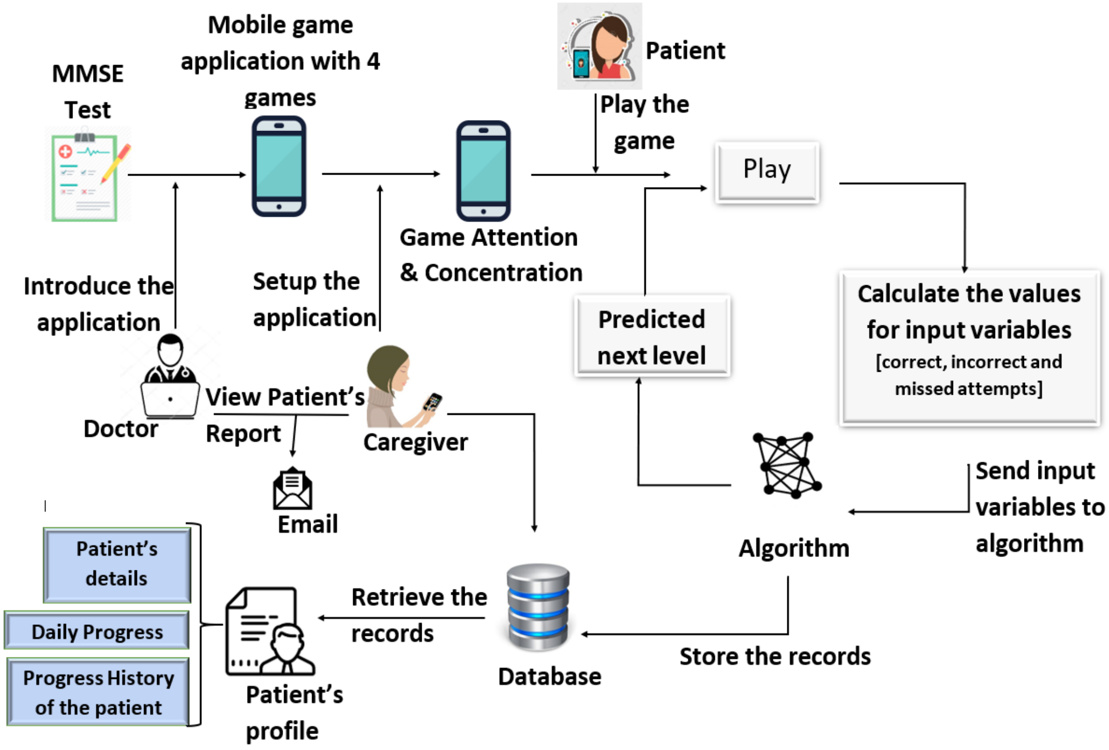

MMSE Test
Mobile game application with 4 games
Patient
Play the game
Play
Game Attention & Concentration
Introduce the application
Setup the application
Predicted next level
Calculate the values for input variables [correct, incorrect and missed attempts]
Doctor
View Patient's Report
Caregiver
Email
Algorithm
Send input variables to algorithm
Patient's details
Daily Progress
Progress History of the patient
Retrieve the records
Patient's profile
Database
Store the records

Figure 2: System Overview Diagram

After setup the application the patient can start the game. The instructions will be displayed in both Sinhala and English languages. Before playing the game, the user can watch the help videos. Game has 3 levels and theses levels planned for one-month period as first level one week, second level one week and the third level two weeks. The user can play the first step and then the daily progress will be displayed. Then the data of the first step will be send to algorithm and the algorithm will predict the necessary variables to set up the second step in a very personalized way for the patient. Like this after every step the algorithm will gives the predictions that needed for the next step. After completing one week the report will be sent to the doctor including personal data of the patient and the progress data of the week. This mail will be Cc to the caregiver as well. After completing the three steps the patient can get the advice from the doctor to whether to continue playing or not. So, after completing the last step of level 3 it will display message saying successfully completed the cognitive therapy and give the user option to reset the game to the first level first step. This way the patient can get the help from this application to improve the Attention and Concentration functions in a cognitive way by playing the game.

2.1.2 Software Development Life Cycle

The Software Development Life Cycle explains regarding the different stages that is interacted in the development of software. The SLDC gives a well-planned structed flow of six different phrases that helps to produce a high-quality software that is tested well enough and ready product. Planning, Analysis, Design, Implementation, Testing & Integration and finally the Maintenance are the mentioned six phrases in the SLDC. And it is necessary to stick to the phrases to give the product in systematic and disciplined manner. Some steps can merge with other steps of the phrase. As example the testing and the implementation goes parallel way during bug fixation during the testing period.

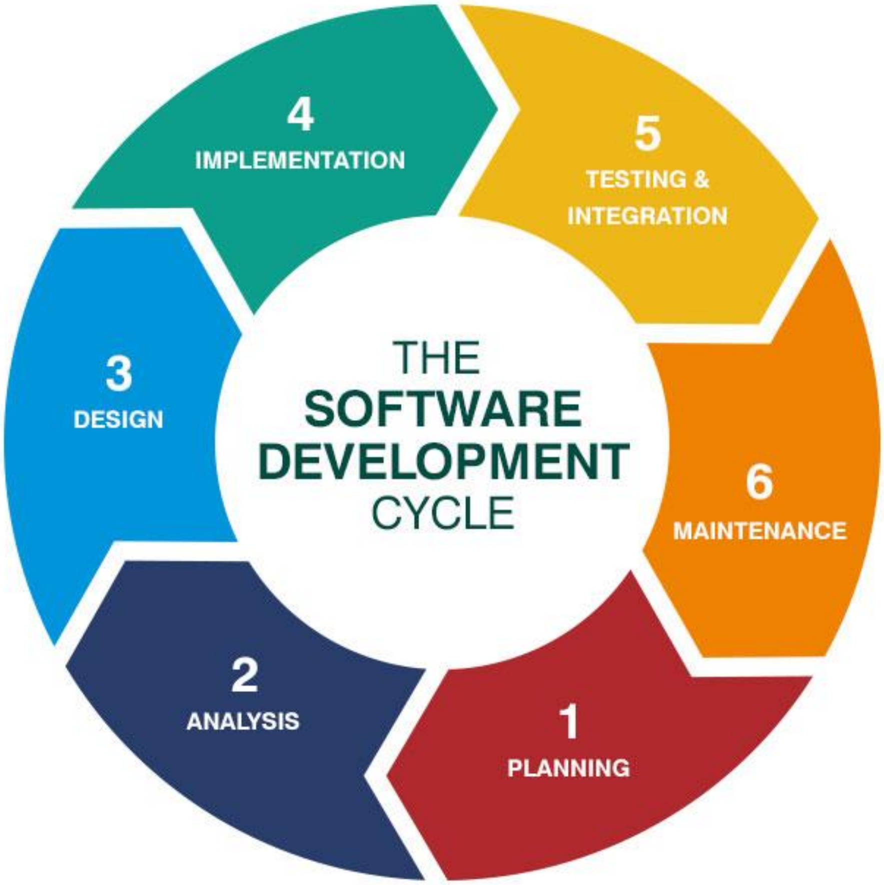


Figure 3: Software Development Life Cycle

SLDC can achieve the different goals such as good quality software, short development period etc. by following a plan. So, the implementation of the D-care application was also followed the SLDC.

- Planning

First identified the problem as need of personalized application for the dementia patients in improve the Attention and Concentration functions in cognitive way. Gathered the needed requirements which is important fundamental in SLDC.

- Analysis

After the requirement gathering, completed the feasibility study (Feasibility study for Proposed solution, Economic Feasibility study and Operational Feasibility Study). After that analyzed all the gathered information. Then defined the different technical approaches that will suited for the successful project implementation.

- Design

Designed the solution with the help and guidance of consultant psychiatrist, after few rounds of sessions with feedback and suggestions. The game design for the Attention and Concentration was completed with finalized sketch of game, interfaces and database etc.

- Implementation

Implementation was started with the core layer of the game. Then completed the first level and followed the same way to finish second level and the third level. Connect the server and managed the payer data. After that implemented the reinforcement learning algorithm. As the next step finished the progress reports and email functionalities. Finally added the other supportive pages and the final touch for the game.

- Testing

The application was tested with unit testing, integration testing etc. Some bugs were found and fixed them and retested.

- Maintenance

During this phrase with the customer feedback or with suggested enhancements the product deployment may happen in stages as new releases.

2.1.3 Agile Scrum Process

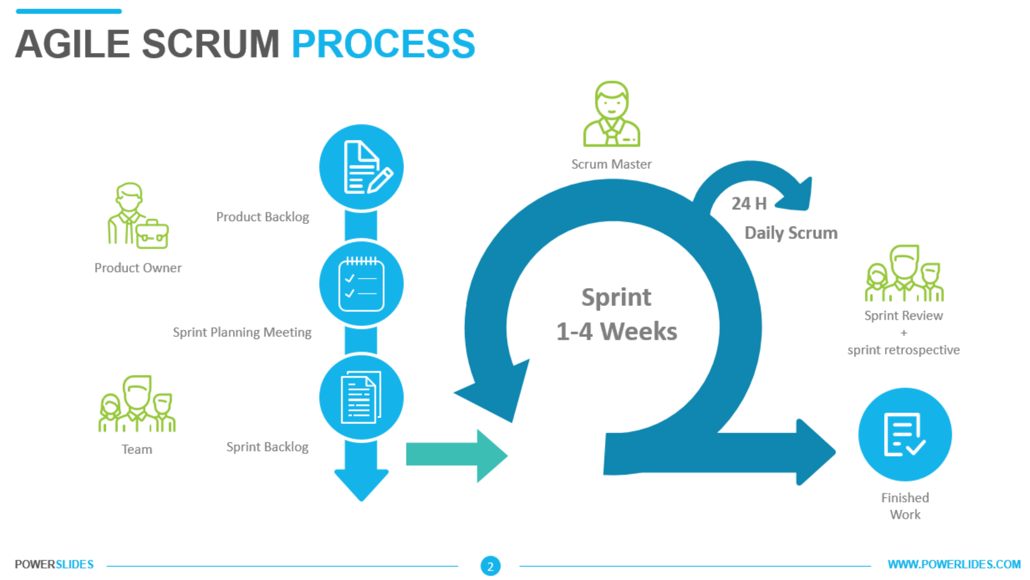
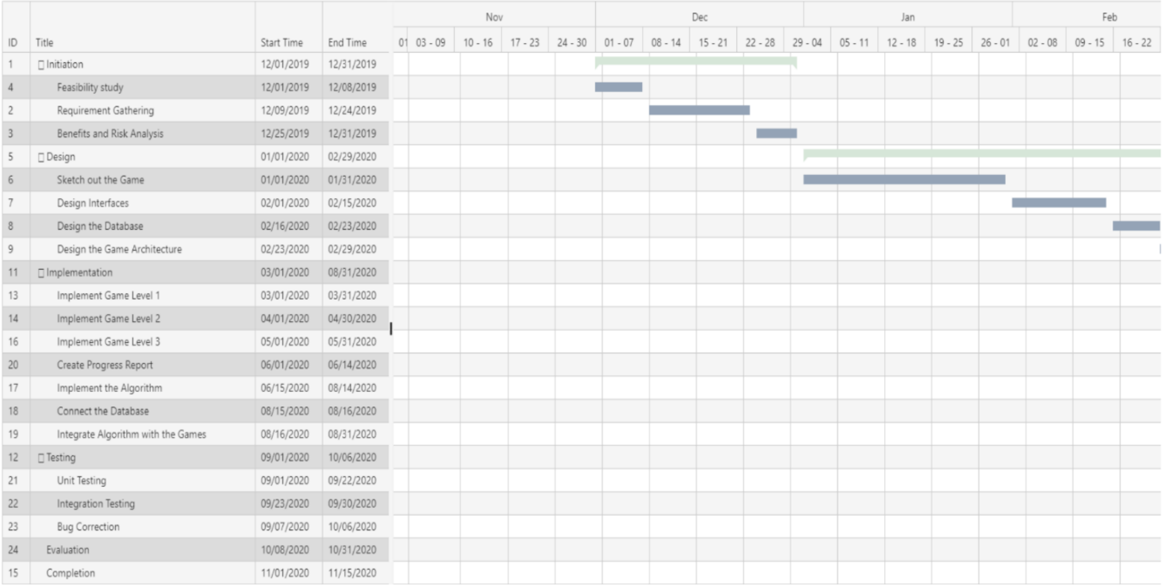


Figure 4: Agile Scrum Process

The software development methodology which will be utilized the Agile Scrum Process. The developers will be provided lots of features to build up a system in an adaptive manner. As well as it is easy for system developers to fulfill the new requirements of the customer. When considering about this research project, there are lots of changes to be done in the future while implementing the system. It is suitable when comparing to processes like Waterfall model etc. Scrum process is mostly focused on task management within a time-based environment. This process is supported for team performance and solved the problems individually since this project is going to be done among four members. Having daily scrum meetings will help to understand the updates of all members of their research area.

2.1.4 Gantt Chart



ID	Title	Start Time	End Time
1	□ Initiation	12/01/2019	12/31/2019
4	Feasibility study	12/01/2019	12/08/2019
2	Requirement Gathering	12/09/2019	12/24/2019
3	Benefits and Risk Analysis	12/25/2019	12/31/2019
5	□ Design	01/01/2020	02/29/2020
6	Sketch out the Game	01/01/2020	01/31/2020
7	Design Interfaces	02/01/2020	02/15/2020
8	Design the Database	02/16/2020	02/23/2020
9	Design the Game Architecture	02/23/2020	02/29/2020
11	□ Implementation	03/01/2020	08/31/2020
13	Implement Game Level 1	03/01/2020	03/31/2020
14	Implement Game Level 2	04/01/2020	04/30/2020
16	Implement Game Level 3	05/01/2020	05/31/2020
20	Create Progress Report	06/01/2020	06/14/2020
17	Implement the Algorithm	06/15/2020	08/14/2020
18	Connect the Database	08/15/2020	08/16/2020
19	Integrate Algorithm with the Games	08/16/2020	08/31/2020
12	□ Testing	09/01/2020	10/06/2020
21	Unit Testing	09/01/2020	09/22/2020
22	Integration Testing	09/23/2020	09/30/2020
23	Bug Correction	09/07/2020	10/06/2020
24	Evaluation	10/08/2020	10/31/2020
15	Completion	11/01/2020	11/15/2020

Figure 5:Gantt Chart

Gantt Chart

(Link for the Gantt chat, since this image is not clear.)

2.1.5 Work Breakdown Structure

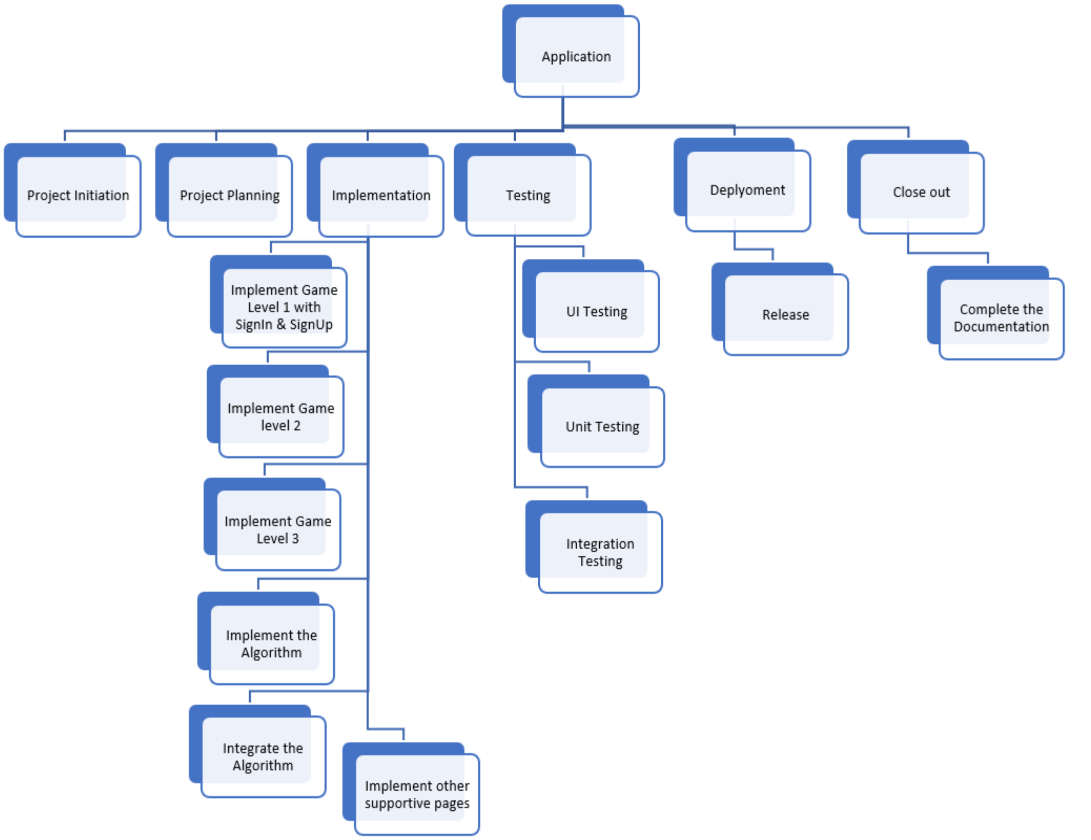


Figure 6: Work Breakdown Structure

The development of the application was done using work breakdown structure. The whole project was broken into smaller tasks, which is popular productivity technique use to manage the work. The first level (root level) and the second level define the set of outcomes that planned collectively and exclusively present the total scope of the project. And this is one of the important project management tools.

First the work was breakdown in to six main tasks. Among them the implementation task was again divide in to six sub parts to make the task less complex and approachable. And the testing task also divided into subtasks as shown in the Figure6.

2.2 Commercialization

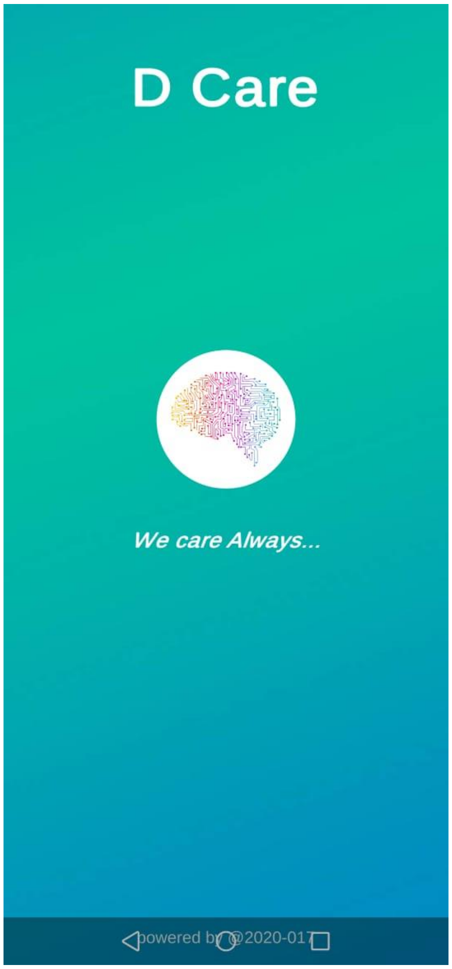


Figure 7: D-care Application

D-care is the best cognitive rehabilitation mobile application for Dementia patients. The cognitive therapy needed to improve the Attention and Concentration is providing in a fun and very interactive way via a game. The game was designed under the supervision of the consultant psychiatrist in the best suited way for the patients by considering almost all the necessary medical requirements. Each patient is provided with personalized game using reinforcement learning theories. The pages are in both Sinhala and English languages in necessary places, so the users who are not fluent in English also can use the application without having any language issues.   Before the playing the user can read the instructions in both languages. Here there are three separate sections to explain the instructions for the three main levels separately. To give the user more help in undressing the game included a help page which has small demonstration of playing the game. The game has three main levels, and first two levels consist of seven steps and the level three have fourteen steps altogether. These step count and levels are designed in the best way to improve the cognitive skills. Normally the patient should at least engage in these games around three weeks to get better results. So, this game was planned for one moth time duration. If after playing the game for one month and then the doctor recommends the patient to play the game again, then a small amount of payment is taken, which is fair and bearable for the patients. The game is on the daily basis and the next level will be available for the user when the next day starts. This will stop complete the all the steps at once which is not the best practice in cognitive rehabilitation. While playing the game if user needs to pause the game, it is possible to pause the game. But for the better results it is recommend completing the game without pause in middle. More than adding text to show the side of the vehicle category an image is added which will be easy for the user when playing more than reading the category. After playing the game the user can see the daily progress of the game and the next level that needs to achieve. This daily progress will be added to the weekly progress chart so that the user can see the improvement of the Attention and Concentration separately throughout the week. And in the progress history all the results from the first step up to the latest finished step will be displayed. After playing the game for one week an email will send automatically to the doctor and the caregiver which contains the progress of the patient, which will be helpful in keeping the doctor updated with the current status of the patient. This is very user-friendly application. that has lot of new features.

2.3 Testing and Implementation

2.3.1 Implementation

The personalized application for dementia patients to improve the Attention and Concentration was developed using the Unity (2019.4.4f1) as the game engine and used C# as the language. For the data management used the internal data managing tool PlayerPrefs and the PlayFab which is complete backend platform service.

- Unity

  Unity is a cross-stage game engine created by Unity Technologies, first declared and delivered in June 2005 at Apple Inc's. Worldwide Developers Conference as a Mac OS X-selective game motor. Starting at 2018, the motor had been reached out to help in excess of 25 stages. The engine can be utilized to make three-dimensional, two-dimensional, computer generated experience, and expanded reality games, just as reproductions and other experiences. The engine has been embraced by ventures outside video gaming, for example, film, car, design, building and development.

Figure 8: Unity

Figure 9: PlayFab

- PlayFab

PlayFab is a complete backend stage for live games with oversaw game administrations, constant investigation, and LiveOps. Lift your income and increment player commitment while reducing expenses. This theme gives an elevated level review of the PlayFab highlights.

PlayFab's backend administrations lessen the hindrances to dispatch for game engineers, offering both huge and little studios savvy improvement arrangements that scale with their games and help them draw in, hold and adapt players. PlayFab empowers designers to utilize the astute cloud to construct and work games, break down gaming information and improve by and large gaming encounters. The PlayFab stage is a characteristic supplement to Azure for gaming (Visit azure.com/gaming for more data). Sky blue, with areas in 42 districts around the world, gives elite worker framework, permitting makers to zero in on building incredible games with best-accessible worldwide reach. For gamers, this prompts a higher, quicker level of advancement and better encounters.

- PlayerPrefs

PlayerPrefs stores and accesses player preferences between game sessions. On android devices the data is saved in the SharedPreferences. C# is using to access all the PlayerPrefs data. The data is physically stored in xml file. (/data/data/pkg-name/shared_prefs/pkg-name.xml)

- C#

Figure 10: C#

C# is modern, simple and general-purpose programming language which object oriented. Microsoft developed this language that runs on the .NET Framework.

- SMTP

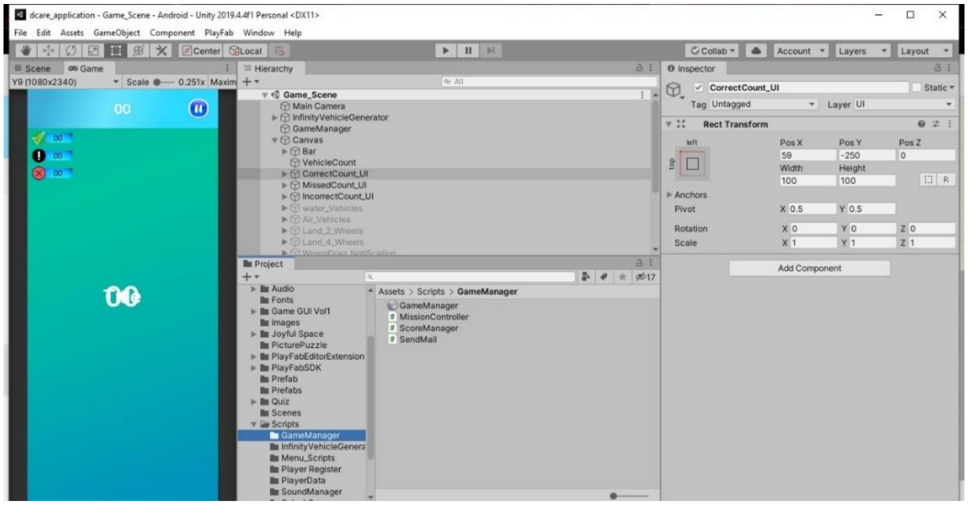
The Simple Mail Transfer Protocol (SMTP) is a correspondence convention for electronic mail transmission. As an Internet standard, SMTP was first characterized in 1982 by RFC 821, and refreshed in 2008 by RFC 5321 to Extended SMTP increments, which is the convention assortment in far and wide use today. Mail workers and other message move operators use SMTP to send and get mail messages. SMTP workers usually utilize the Transmission Control Protocol on port number 25.

Used Microsoft planner as project management tool and GitLab as the version controller.

- **Game Implementation**

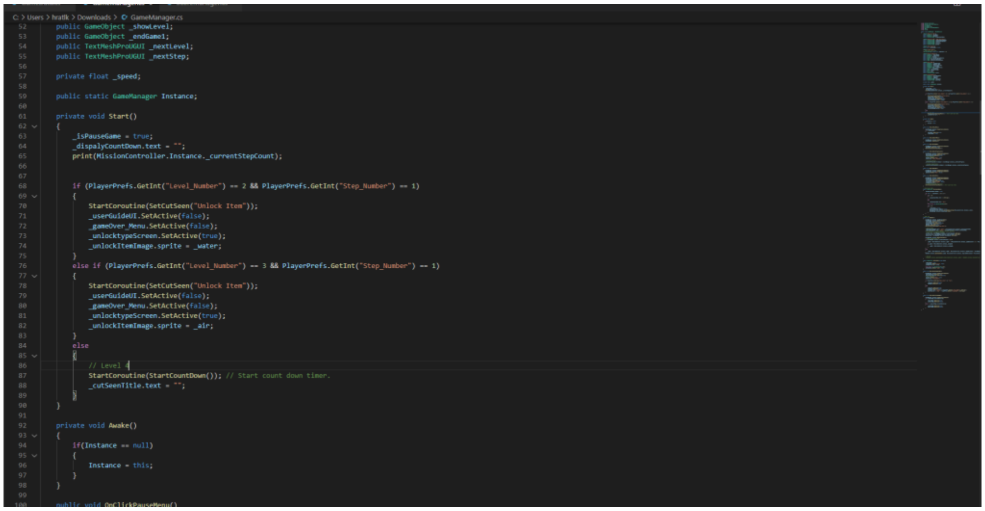
The game attention and concentration have three main levels. The first level and the second level have seven steps needs to play throughout the week. And when the user is more familiar with the game the difficult level the third level will starts after two weeks and this level has fourteen steps which can count as two week of time period.

First implemented the first level of the game, the first scene was implemented and then the vehicle generator was created. This will generate the vehicles coming from top of the screen as randomly selected from a set of images. To get the control of the swipes the Swipe controller was implemented. The score calculations and displayed them to in the user interfaces. Mission controller and the Game manager is handling the flow of the games managing all the levels and step. The second level and the third level of the game was implemented in the same way adding one more vehicle category.



CorrectCount_UI
Static
Tag Untagged
Layer UI
Rect Transform
Pos X 59
Pos Y -250
Pos Z 0
Width 100
Height 100
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

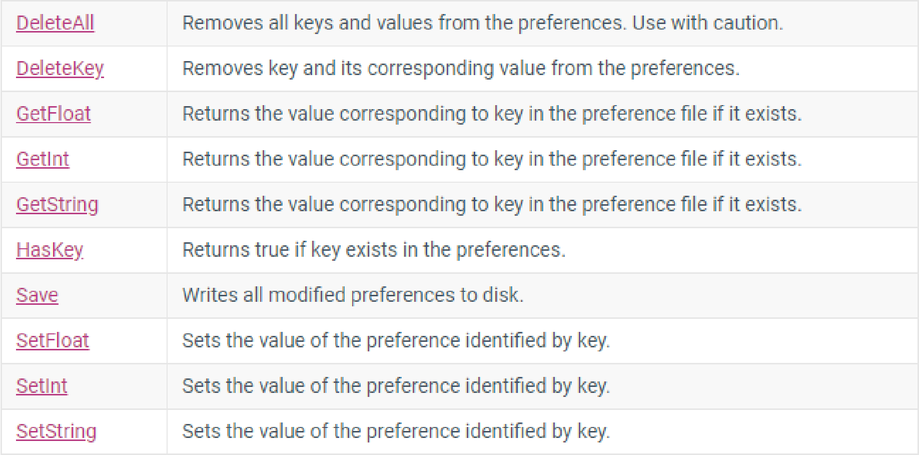
Figure 11: Unity scene



C: > Users > hratk > Downloads > GameManager.cs

```
    public GameObject _showLevel;
    public GameObject _endGame1;
    public TextMeshProUGUI _nextLevel;
    public TextMeshProUGUI _nextStep;

    private float _speed;

    public static GameManager Instance;

    private void Start()
    {
        _isPauseGame = true;
        _dispalyCountDown.text = "";
        print(MissionController.Instance._currentStepCount);


        if (PlayerPrefs.GetInt("Level_Number") == 2 && PlayerPrefs.GetInt("Step_Number") == 1)
        {
            StartCoroutine(SetCutSeen("Unlock Item"));
            _userGuideUI.SetActive(false);
            _gameOver_Menu.SetActive(false);
            _unlocktypeScreen.SetActive(true);
            _unlockItemImage.sprite = _water;
        }
        else if (PlayerPrefs.GetInt("Level_Number") == 3 && PlayerPrefs.GetInt("Step_Number") == 1)
        {
            StartCoroutine(SetCutSeen("Unlock Item"));
            _userGuideUI.SetActive(false);
            _gameOver_Menu.SetActive(false);
            _unlocktypeScreen.SetActive(true);
            _unlockItemImage.sprite = _air;
        }
        else
        {
            // Level 4
            StartCoroutine(StartCountDown()); // Start count down timer.
            _cutSeenTitle.text = "";
        }
    }

    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
    }

    public void OnClickPauseMenu()
```

Figure 12: Game Manager



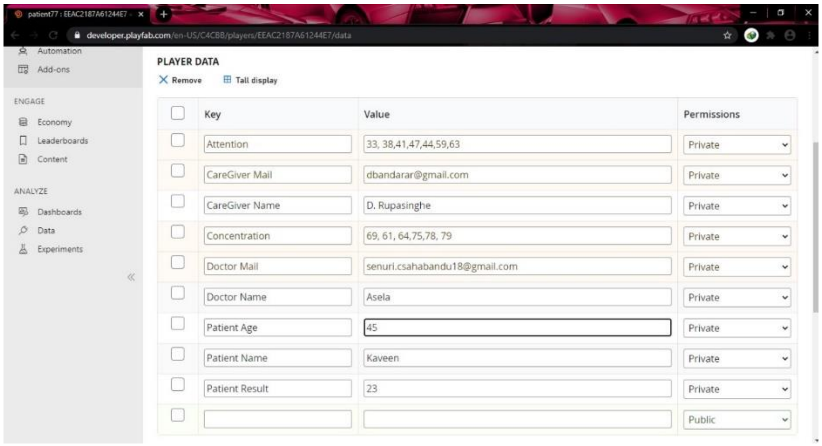
DeleteAll	Removes all keys and values from the preferences. Use with caution.
DeleteKey	Removes key and its corresponding value from the preferences.
GetFloat	Returns the value corresponding to key in the preference file if it exists.
GetInt	Returns the value corresponding to key in the preference file if it exists.
GetString	Returns the value corresponding to key in the preference file if it exists.
HasKey	Returns true if key exists in the preferences.
Save	Writes all modified preferences to disk.
SetFloat	Sets the value of the preference identified by key.
SetInt	Sets the value of the preference identified by key.
SetString	Sets the value of the preference identified by key.

Figure 13: Static methods in PlayerPrefs

Then implemented the other supportive pages as Help, Instructions, Game menus etc.

The PlayerPrefs was used to manage data among the scenes. Figure 13 shows some commonly used methods in data management.

The Signup, Sign In functionalities along with the user profile was implemented. To keep the user data stored safely not only locally used PlayFab. So, the user can Sign In from any device and can continue from where he stopped previously. Otherwise the when the cache is cleared, or user change the phone the data will be lost. After signup and completing every step the data will be send to the PlayFab (Figure: 14). And it is faster, best-available, high reliability and better experience in managing the game data specially.



PLAYER DATA

□	Key	Value	Permissions
□	Attention	33, 38,41,47,44,59,63	Private
□	CareGiver Mail	dbandarar@gmail.com	Private
□	CareGiver Name	D. Rupasinghe	Private
□	Concentration	69, 61, 64,75,78, 79	Private
□	Doctor Mail	senuri.csahabandu18@gmail.com	Private
□	Doctor Name	Asela	Private
□	Patient Age	45	Private
□	Patient Name	Kaveen	Private
□	Patient Result	23	Private
□			Public

Figure 14: PlayFab

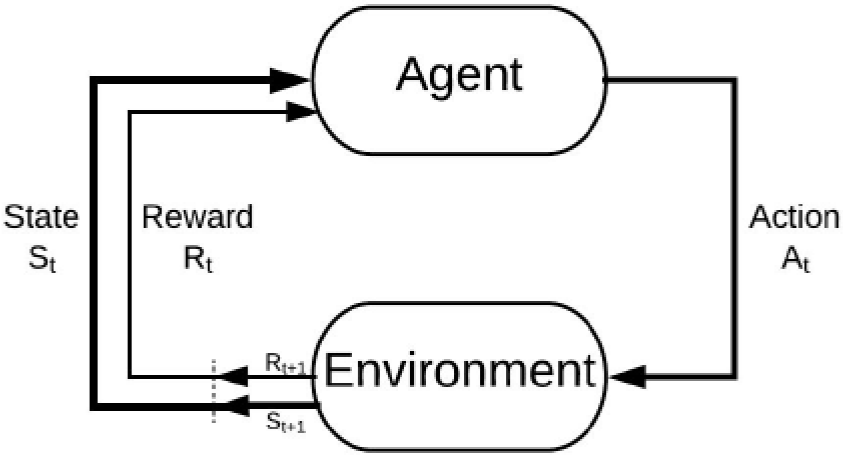


Figure 15: Basic components in RL

- **Algorithm Implementation**

The reinforcement learning which is one of the machine learning technology that used in the games to get the prediction for the next level. RL is scientific art to make the optimal decisions using the experiences. The figure 15 shows the main components in the RL.

There are two types of RL algorithms as model free and model based. According to the game requirements the model free algorithm was used in order to estimate the optimal policy without using reward function and transition (dynamics) of the environment.

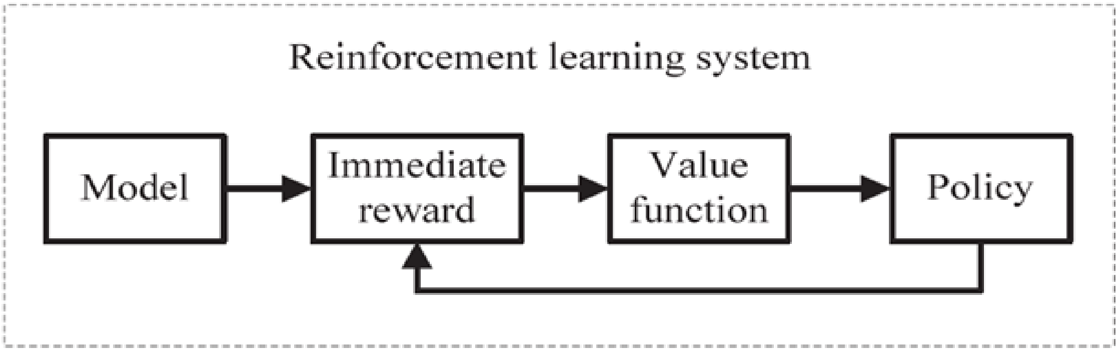
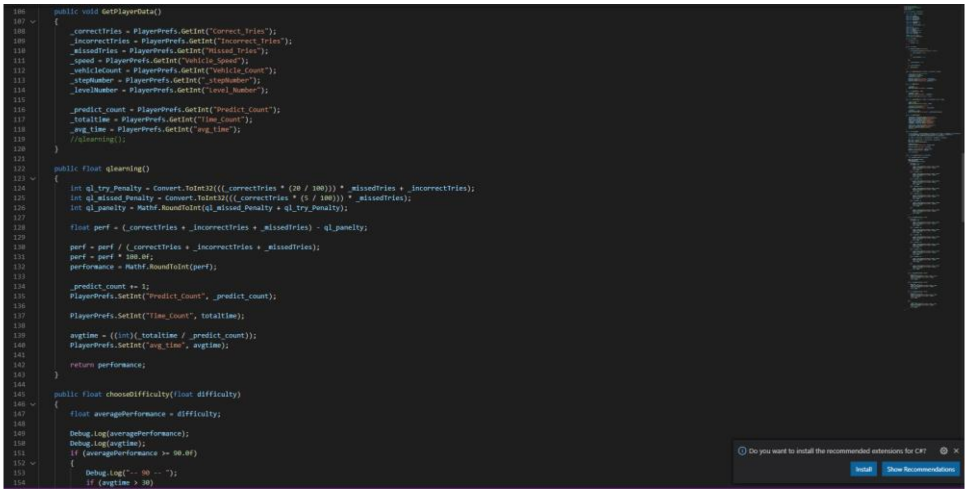


Figure 16: Reinforcement Learning System



```
public void GetPlayerData()
{
    _correctTries = PlayerPrefs.GetInt("Correct_Tries");
    _incorrectTries = PlayerPrefs.GetInt("Incorrect_Tries");
    _missedTries = PlayerPrefs.GetInt("Missed_Tries");
    _speed = PlayerPrefs.GetInt("Vehicle_Speed");
    _vehicleCount = PlayerPrefs.GetInt("Vehicle_Count");
    _stepNumber = PlayerPrefs.GetInt("_stepNumber");
    _levelNumber = PlayerPrefs.GetInt("Level_Number");

    _predict_count = PlayerPrefs.GetInt("Predict_Count");
    _totaltime = PlayerPrefs.GetInt("Time_Count");
    _avg_time = PlayerPrefs.GetInt("avg_time");
    //qlearning();
}

public float qlearning()
{
    int ql_try_Penalty = Convert.ToInt32(((_correctTries * (20 / 100))) * _missedTries + _incorrectTries);
    int ql_missed_Penalty = Convert.ToInt32(((_correctTries * (5 / 100))) * _missedTries);
    int ql_panelty = Mathf.RoundToInt(ql_missed_Penalty + ql_try_Penalty);

    float perf = (_correctTries + _incorrectTries + _missedTries) - ql_panelty;

    perf = perf / (_correctTries + _incorrectTries + _missedTries);
    perf = perf * 100.0f;
    performance = Mathf.RoundToInt(perf);

    _predict_count += 1;
    PlayerPrefs.SetInt("Predict_Count", _predict_count);

    PlayerPrefs.SetInt("Time_Count", totaltime);

    avgtime = ((int)(_totaltime / _predict_count));
    PlayerPrefs.SetInt("avg_time", avgtime);

    return performance;
}

public float chooseDifficulty(float difficulty)
{
    float averagePerformance = difficulty;

    Debug.Log(averagePerformance);
    Debug.Log(avgtime);
    if (averagePerformance >= 90.0f)
    {
        Debug.Log("-- 90 -- ");
        if (avgtime > 30)
```

Do you want to install the recommended extensions for C#?

Figure 17: The reinforcement learning algorithm

This algorithm was implemented referring the Q-Learning and Deep Q-Learning Algorithms in the Reinforcement Learning. Due to the nature of both algorithms and the game it was not possible to move with one algorithm. So, taking the necessary features from the both algorithms the new algorithm was implemented. The cycle of Q-Learning makes a definite lattice for the working specialist which it can "refer to" to maximize its reward over the long haul. Although this methodology is right, this is just viable for little conditions and rapidly loses its attainability when the quantity of states and activities in the environment.

The answer for the above issue originates from the acknowledgment that the qualities in the matrix just have relative significance is the qualities just have significance regarding different qualities. Along these lines, this reasoning leads us to Deep Q-Learning which utilizes a profound neural organization to rough the qualities. This estimate of qualities doesn't hurt if the overall significance is preserved.

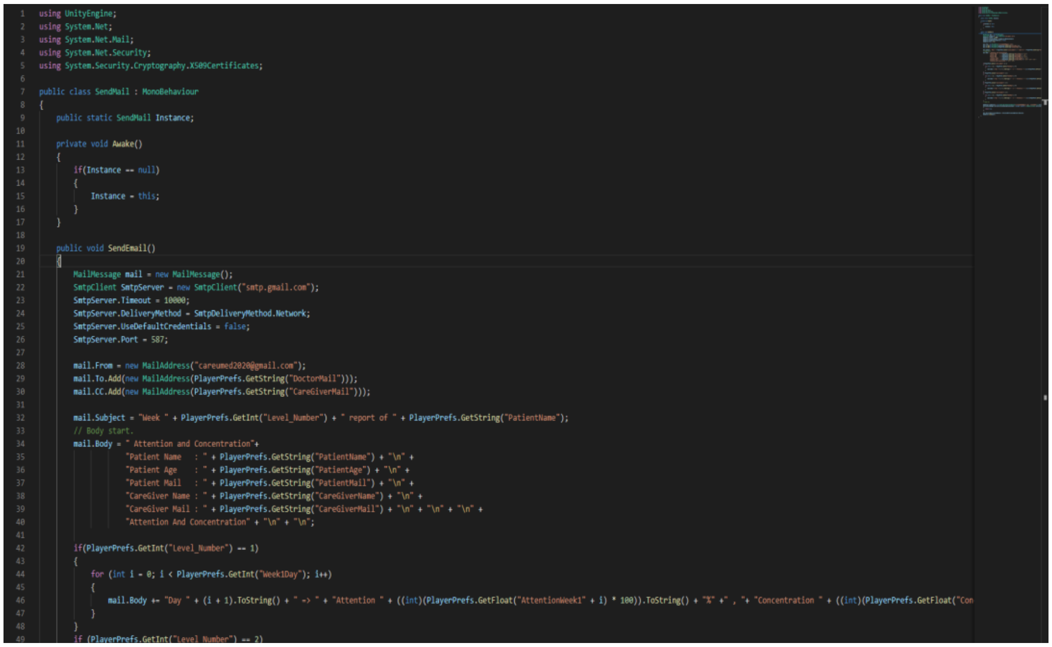
As the parameters for the algorithm the correct count, missed count and wrong count was taken. We cannot have a better accuracy in predicting the variables needed for the next level if we haven’t considered about the penalty situations that could happen. Human cannot play with 100% performance not a healthy young person even. And also, there could be technical issues as well in the application and the using device. Considering all these factors the penalty calculation was used for the input variables. In this algorithm the penalty percentage is taken as 20 and the missed penalty is taken as 5. This will improve the accuracy of the output values from the algorithm.

And referred the Dynamic Difficulty Adjustment (DDA). DDA is a strategy for consequently altering a game's highlights, practices, and situations continuously, contingent upon the player's ability, with the goal that the player, when the game is exceptionally straightforward, doesn't feel exhausted or baffled when it is extremely troublesome. The expectation of the DDA is to keep the player fascinated till the end and to give him/her with a difficult encounter. In conventional games, trouble levels increment straightly or stepwise throughout the game. The highlights, for example, recurrence, beginning levels, or rates can be set distinctly toward the start of the game by picking a degree of trouble. This can, in any case, bring about a negative encounter for players as they attempt to plan a pre decided expectation to absorb information. DDA endeavors to tackle this issue by introducing an altered answer for gamers.

The fundamental working advance for Deep Q-Learning is that the underlying state is taken care of into the neural organization and it restores the Q-estimation of all potential activities as on yield.

- Completion of the Implementation

After integrating the algorithm with the game, implemented the report generation functionalities (Daily report, Weekly report and progress history). Then implemented the email functionality to send the report to the doctor and caregiver after every week via email (Figure: 18). SMTP server using implementing this functionality. Added the sounds for the game, game icon and animation to some interfaces.



```
using UnityEngine;
using System.Net;
using System.Net.Mail;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;

public class SendMail : MonoBehaviour
{
    public static SendMail Instance;

    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
    }

    public void SendEmail()
    {
        MailMessage mail = new MailMessage();
        SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
        SmtpServer.Timeout = 10000;
        SmtpServer.DeliveryMethod = SmtpDeliveryMethod.Network;
        SmtpServer.UseDefaultCredentials = false;
        SmtpServer.Port = 587;

        mail.From = new MailAddress("careumed2020@gmail.com");
        mail.To.Add(new MailAddress(PlayerPrefs.GetString("DoctorMail")));
        mail.CC.Add(new MailAddress(PlayerPrefs.GetString("CareGiverMail")));

        mail.Subject = "Week " + PlayerPrefs.GetInt("Level_Number") + " report of " + PlayerPrefs.GetString("PatientName");
        // Body start.
        mail.Body = " Attention and Concentration"+
                    "Patient Name   : " + PlayerPrefs.GetString("PatientName") + "\n" +
                    "Patient Age    : " + PlayerPrefs.GetString("PatientAge") + "\n" +
                    "Patient Mail   : " + PlayerPrefs.GetString("PatientMail") + "\n" +
                    "CareGiver Name : " + PlayerPrefs.GetString("CareGiverName") + "\n" +
                    "CareGiver Mail : " + PlayerPrefs.GetString("CareGiverMail") + "\n" + "\n" + "\n" +
                    "Attention And Concentration" + "\n" + "\n";

        if(PlayerPrefs.GetInt("Level_Number") == 1)
        {
            for (int i = 0; i < PlayerPrefs.GetInt("Week1Day"); i++)
            {
                mail.Body += "Day " + (i + 1).ToString() + " -> " + "Attention " + ((int)(PlayerPrefs.GetFloat("AttentionWeek1" + i) * 100)).ToString() + "%" +" , "+ "Concentration " + ((int)(PlayerPrefs.GetFloat("Con
            }
        }
        if (PlayerPrefs.GetInt("Level_Number") == 2)
```

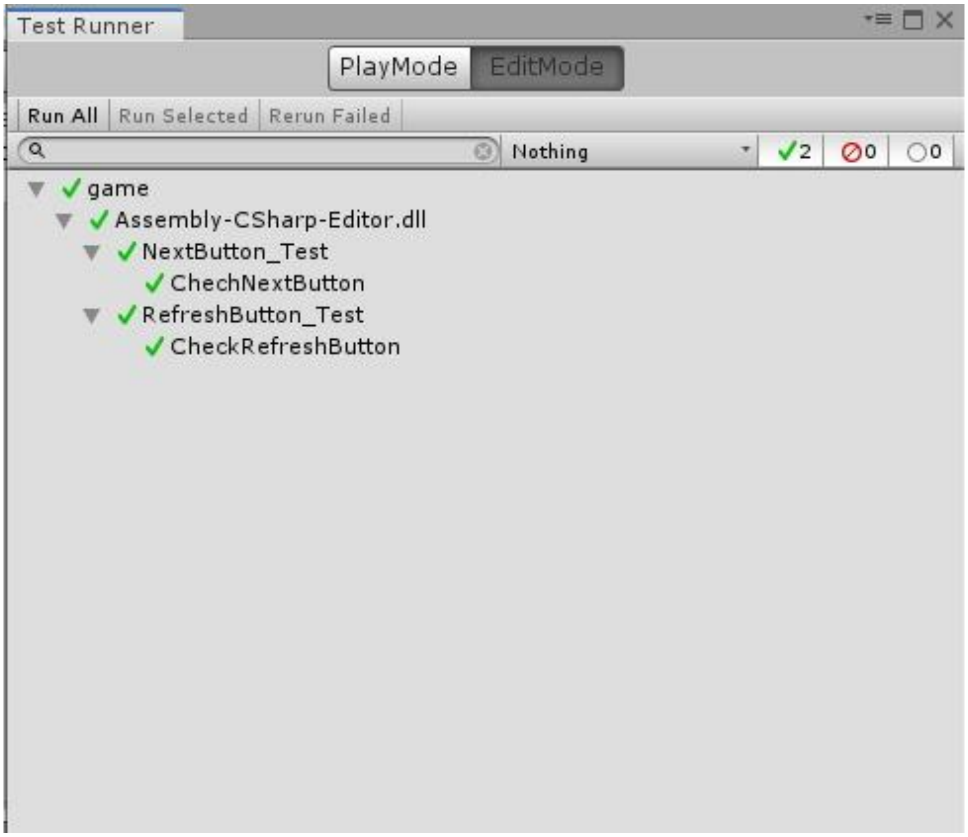
Figure 18:SendEmail.cs file

2.3.2 Testing

Unit Testing is a sort of programming testing where singular units or segments of a product are tried. The reason for existing is to approve that every unit of the product code proceeds true to form. Unit Testing is finished during the turn of events (coding period) of an application by the designers. Unit Tests confine a part of code and check its rightness. A unit might be an individual capacity, strategy, system, module, or article.

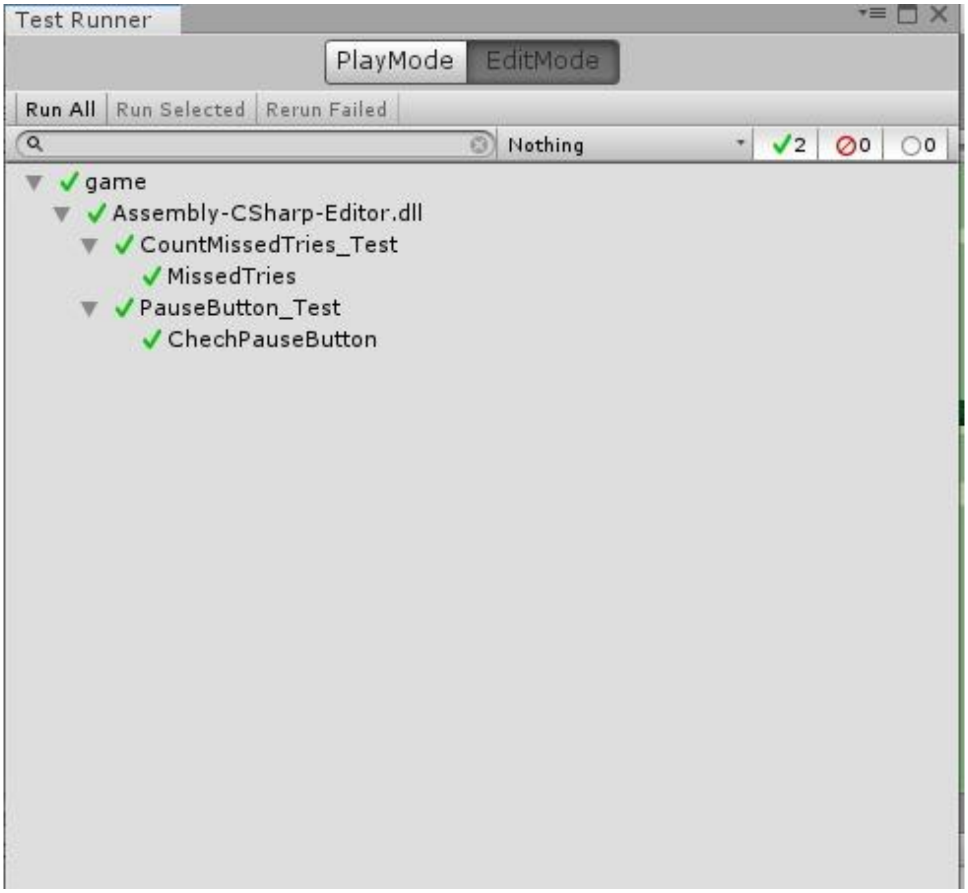
Unit Testing is significant in light of the fact that product engineers now and then have a go at sparing time doing insignificant unit testing and this is legend in light of the fact that wrong unit testing prompts significant expense Defect fixing during System Testing, Integration Testing and even Beta Testing after application is constructed. On the off chance that legitimate unit testing is done in early turn of events, at that point it sets aside time and cash at long last.

The following figure are displaying the few test cases performed for the Attention and Concentration application.



```
▼ ✓ game
    ▼ ✓ Assembly-CSharp-Editor.dll
        ▼ ✓ NextButton_Test
              ✓ ChechNextButton
        ▼ ✓ RefreshButton_Test
              ✓ CheckRefreshButton
```

Figure 19: Test for Next button



```
▼ ✓ game
  ▼ ✓ Assembly-CSharp-Editor.dll
    ▼ ✓ CountMissedTries_Test
        ✓ MissedTries
    ▼ ✓ PauseButton_Test
        ✓ ChechPauseButton
```

Figure 20: Test for the missed tries

Test Cases

Table 2: Test case 1 for display correct attempts

Test ID	**01**
Description	Display the count of correct attempts
Pre-condition	The patient should login to the game
Steps	1. Go to home page. 2. Navigate to Attention and Concentration game. 3. Click the play button.

	4. Swipe the vehicle images towards the correct sides.
Extensions	If the patient clicks the pause button, the game will pause.
Expected output	Display the correct no of attempts
Actual output	Display the correct no of attempts
Pass/Fail	Pass

Table 3: Test case 2 for Display the daily progress

Test ID	**02**
Description	Display the daily progress for both attention and concentration
Pre-condition	The patient should login to the game
Steps	5. Go to home page. 6. Navigate to Attention and Concentration game. 7. Click the play button. 8. Swipe the vehicle images towards the correct sides. 9. Finish the step
Extensions	If the patient clicks the pause button, the game will pause.
Expected output	Display the Daily progress for both Attention and Concentration in Sinhala and English languages.

Actual output	Display the Daily progress for both Attention and Concentration in Sinhala and English languages.
Pass/Fail	Pass

Table 4: Test case 3 for New vehicle category

Test ID	**03**
Description	Level 2 new selection category
Pre-condition	The patient should login to the game
Steps	10. Go to home page. 11. Navigate to Attention and Concentration game. 12. Complete the first level of the game. 13. Click the play button for the second level. 14. Swipe the vehicle images towards the correct sides and play the game.
Extensions	If the patient clicks the pause button, the game will pause.
Expected output	Display the unlock category and new transportation vehicle category added, can swipe the water transportation vehicles to the newly added category.
Actual output	Display the unlock category and new transportation vehicle category added, can

	swipe the water transportation vehicles to the newly added category.
Pass/Fail	Pass

Table 5: Test case 4 for sending email

Test ID	**04**
Description	Send email to Doctor and Caregiver
Pre-condition	The patient should login to the game
Steps	15. Go to home page. 16. Navigate to Attention and Concentration game. 17. Complete the first level of the game.
Extensions	If the patient clicks the pause button, the game will pause.
Expected output	Sends email to the doctor while caregiver receiving the carbon copy with the completed week’s progress and including patient’s and caregiver’s details.
Actual output	Sends email to the doctor while caregiver receiving the carbon copy with the completed week’s progress and including patient’s and caregiver’s details.
Pass/Fail	Pass

3. RESULTS AND DISCUSSION

3.1 Results

Implemented interfaces of the game.

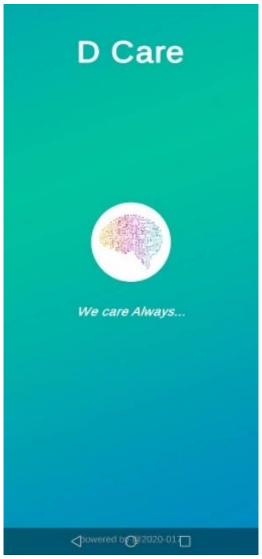


Figure 21: Initial page

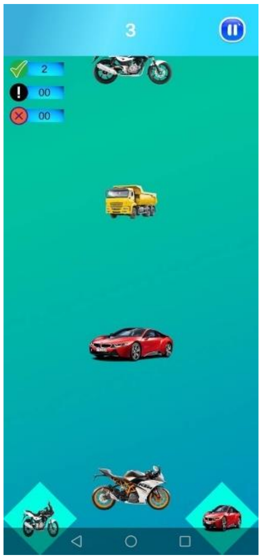


Figure 22: First level

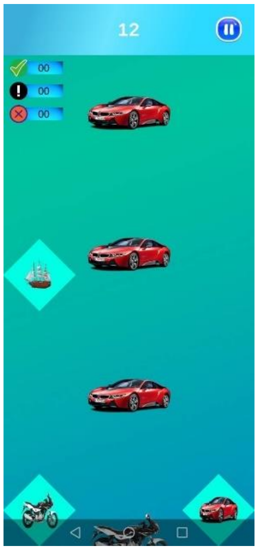


Figure 23: Second Level

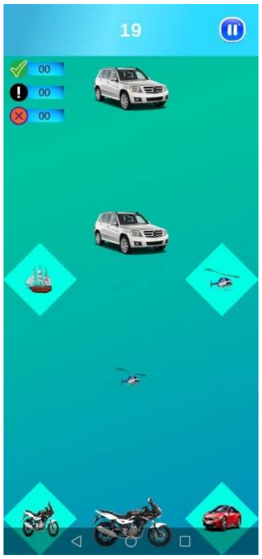


Figure 24: Third level

Progress reports of the patient

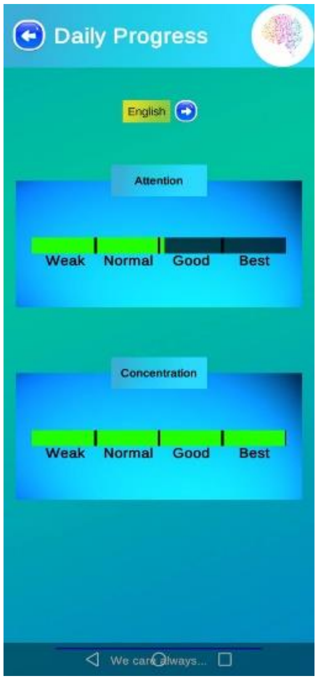


Figure 25: Daily progress -English

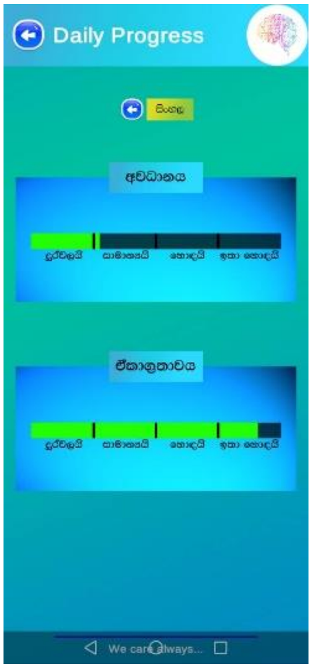


Figure 26: Daily progress - Sinhala

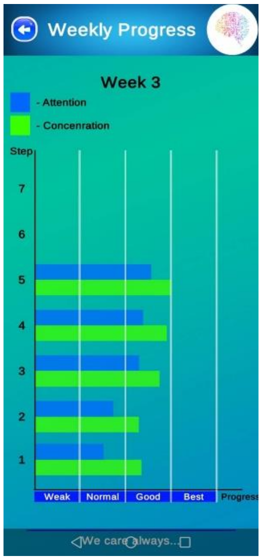
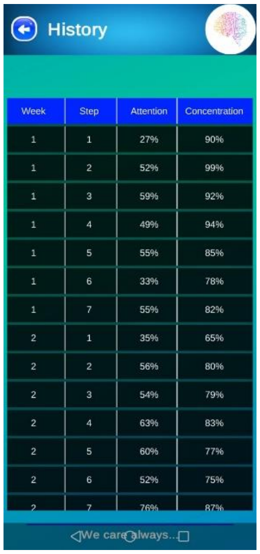
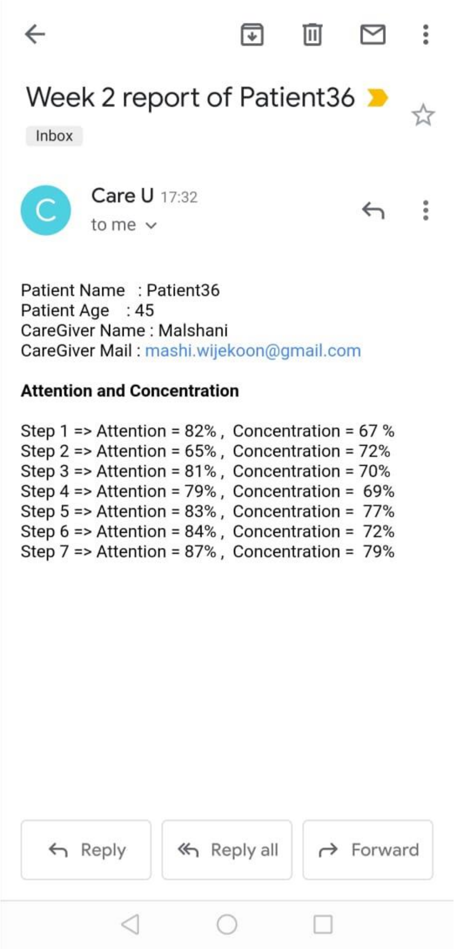


Figure 27: Weekly progress



Week	Step	Attention	Concentration
1	1	27%	90%
1	2	52%	99%
1	3	59%	92%
1	4	49%	94%
1	5	55%	85%
1	6	33%	78%
1	7	55%	82%
2	1	35%	65%
2	2	56%	80%
2	3	54%	79%
2	4	63%	83%
2	5	60%	77%
2	6	52%	75%
2	7	76%	87%

Figure 28: Progress history



Patient Name : Patient36
Patient Age : 45
CareGiver Name : Malshani
CareGiver Mail : mashi.wijekoon@gmail.com

Attention and Concentration

Step 1 => Attention = 82% , Concentration = 67 %
Step 2 => Attention = 65% , Concentration = 72%
Step 3 => Attention = 81% , Concentration = 70%
Step 4 => Attention = 79% , Concentration = 69%
Step 5 => Attention = 83% , Concentration = 77%
Step 6 => Attention = 84% , Concentration = 72%
Step 7 => Attention = 87% , Concentration = 79%

Figure 29: Doctor's and Caregiver's report

Dementia patients can use the D-Care application to improve their cognitive skills with best suited personalized games and activities.

After the implementation and testing the D-care application was tested with the users.

Application was divided in to two categories as with Reinforcement learning and without reinforcement learning. Then the application was given to users.

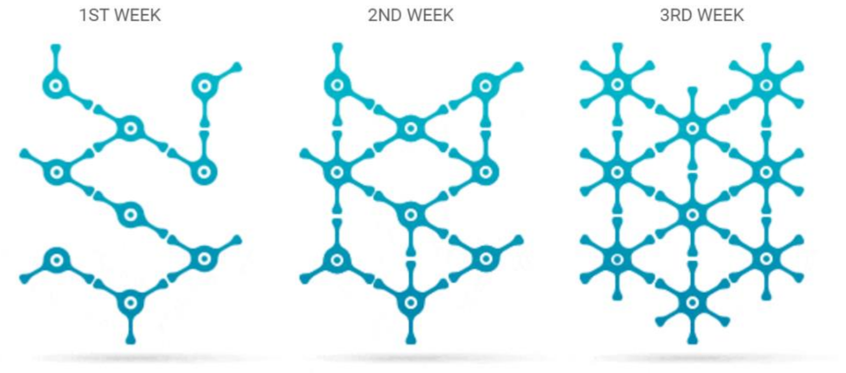
We gathered and analysed the progress results of the users using same application with and with using machine learning technologies as mentioned above. With the results we can say the personalized application shows good results than the other.

So, to make the game personalize the reinforcement learning was good solution. When we increase the penalty values the algorithm doesn’t predict the suitable variables for the next level. As example the speed become inappropriate for the user. And when we reduce the penalty level it increases the difficulty level of the patients. As standard the highest difficulty level using for the games is 75%. If the difficulty level increases continuously the user won’t continue the playing game. With the penalty of 20% and 5% of missed penalty the algorithm predicts the most suited nest level variables.

Because of the limited no of actions in the Q-learning algorithm it cannot use completely for this game. And in the deep q learning algorithm the it stores all the experience in the past and based on them the algorithm predicts the next level. So, with these reasons the new algorithm implemented was success combining the needed features from the q-learning and Deep Q-Learning algorithms.

3.2 Research Findings and Discussion

D-cares an intervention application that uses for Attention and Concentration. This clinical game help reinforce and strengthen the neural activation patterns used in attention and concentration. The repeated activation of these patterns may produce the creation of new synapses and the myelination of the neural circuits capable of recovering or improving attention. The rehashed actuation of these examples may deliver the making of new neurotransmitters and the myelination of the neural circuits fit for recuperating or improving consideration. D-care utilizes its clinical game for Attention regarding help the sensory system advance the recuperation of the mind after experiencing auxiliary shortages, issues, or injury, where consideration and attention are influenced. This personalized game is useful for cognitive recovery as well as ideal for any individual who needs to challenge and improve their cognitive abilities. Playing around with no set preparing program won't assist you with improving your psychological aptitudes. For mind games to be compelling, it must meet the accompanying qualities: A proper intellectual preparing requires a helpful objective, logical approval, and controlled activities, like the games that D-care offers. Following these prerequisites, the mind will get the suitable brain training that it needs.



Graphic projection of neural networks after **3 weeks.**

Technically during the implementation of the game use reinforcement learning for the next level prediction. The RL is the technology of deciding the optimal decisions using experiences. So, to get the needed predicted variables needed for the next step was implemented using this theory. First start the investigating the Q-Learning algorithm which is model free reinforcement learning algorithm

4. CONCLUSION

Dementia patients need activities for Cognitive rehabilitation. To provide the patients personalized therapies, a user-friendly application was developed in a very interactive way including games and activities.

These games and activities are under four main functional areas from Frontal and Temporal lobes of Cerebral cortex such as Attention & Concentration, Executive functions, Language and Memory. The main objective of each game or activity is to improve the relevant cogitative functionalities of the patients in personalized way. The games are executed in touch screen and all the necessary places both Sinhala and English languages are used eliminating the gap between Sinhala and English and the technology gap for the elderly people. In three games QL and DQL algorithms of Reinforcement Learning is used to learn from the user and predict the user specific next level for each user. This will help the patients to receive customized and most suited rehabilitation for the Dementia patients. And two of the games and activities have used Voice to text communication get the necessary user inputs. For the implementation CNN is used. Progress reports are generated for each functional area separately and both the caregiver and doctor can view the progress of the patient continuously. The new games introduced by D-care application improves the cognitive functionalities of the Dementia patients with continuous monitoring.

5. REFERENCES

[1] "Some facts about Dementia and Alzheimer’s disease", *Allianz Sri Lanka | Motor Insurance | Life Insurance | General Insurance*, 2020. [Online]. Available: https://www.allianz.lk/articles/facts-dementia-alzheimers-disease/. [Accessed: 20- Jan-2020].

[2] M. Nezerwa et al., "Alive Inside: Developing Mobile Apps for the Cognitively Impaired", in *2014 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2014.

[3] "Auguste Deter," 2019 [Online]. Available: https://en.wikipedia.org/wiki/Auguste_Deter#cite_note-2/.

[4] A. S. Gill Livingston, "Dementia prevention, intervention, and care," [Online]. Available: https://www.thelancet.com/action/showPdf?pii=S0140-6736%2817%2931363-6.

[5] M. H. Acharya, T. B. Gokani, K. N. Chauhan and B. P. Pandya, "Android application for Dementia patient," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, 2016, pp. 1-4.

[6] "Deep Reinforcement Learning Based Personalized Health Recommendations," [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-33966-1_12.

[7] M. A. Samsuden, N. M. Diah and N. A. Rahman, "A Review Paper on Implementing Reinforcement Learning Technique in Optimising Games Performance," *2019 IEEE 9th International Conference on System Engineering and Technology*

[8] Szita I. (2012) Reinforcement Learning in Games. In: Wiering M., van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27645-3_17

[9] Schmid M (2009). Reinforcing motor re-training and rehabilitation through games: a machine-learning perspective. *Front. Neuroeng*. (2009). **2**:3. doi: 10.3389/ neuro.16.003.2009

[10] A. Hammoudeh, "A Concise Introduction to Reinforcement Learning", 2018. https://www.researchgate.net/publication/323178749_A_Concise_Introduction_to_Reinforcement_Learning. [Accessed 15 July 2020]. [Accessed 15 July 2020]. 47

[11] C. Yu, J. Liu and S. Nemati, "Reinforcement Learning in Healthcare: A Survey", *arXiv.org*, 2019. [Online]. Available: https://arxiv.org/abs/1908.08796. [Accessed: 15-Jul- 2020].

6. APPENDICES

- Game Manger

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using TMPro;
using System;

public class GameManager : MonoBehaviour
{
    [Header("Game UI Items")]
    public GameObject _pauseMenu;
    public GameObject _pauseBtn;
    public TextMeshProUGUI _dispalyCountDown;

    [Header("Game UI Items - Game Over Menu")]
    public TextMeshProUGUI _completedLevelNumber;
    public TextMeshProUGUI _completedStepNumber;
    public TextMeshProUGUI _correct;
    public TextMeshProUGUI _incorrect;
    public TextMeshProUGUI _missed;

    [Header("Game States")]
    public bool _isPauseGame = false;

    [Header("Count Down")]
    // Count down start value.
    [SerializeField] private int _timerStart = 3;

    [Header("Daily Progress")]
    public GameObject _englishPage;
    public GameObject _sinhalaPage;
    public Animator _menuAnim;
    public GameObject _dailyProgressDisplay;
    public Image _attentionProgressUI;
    public Image _concentrationProgressUI;

    [Header("Cut seens")]
    public GameObject _cutSeenScreen;
    public GameObject _gameOver_Menu;
    public GameObject _unlocktypeScreen;
    public GameObject _userGuideUI;
    public TextMeshProUGUI _cutSeenTitle;
    public Animator _cutSeenAnim;
    public Image _unlockItemImage;
    public Sprite _water;
    public Sprite _air;
    public TextMeshProUGUI _title;

    [Header("Achievements")]
    public GameObject _AchieveDisplay;
    public GameObject _showLevel;
    public GameObject _endGame1;
    public TextMeshProUGUI _nextLevel;
    public TextMeshProUGUI _nextStep;

    private float _speed;

    public static GameManager Instance;

    private void Start()
    {
        _isPauseGame = true;
        _dispalyCountDown.text = "";
        print(MissionController.Instance._currentStepCount);
        if (PlayerPrefs.GetInt("Level_Number") == 2 &&
PlayerPrefs.GetInt("Step_Number") == 1)
        {
            StartCoroutine(SetCutSeen("Unlock Item"));
            _userGuideUI.SetActive(false);
            _gameOver_Menu.SetActive(false);
            _unlocktypeScreen.SetActive(true);
            _unlockItemImage.sprite = _water;
        }
        else if (PlayerPrefs.GetInt("Level_Number") == 3 &&
PlayerPrefs.GetInt("Step_Number") == 1)
        {
            StartCoroutine(SetCutSeen("Unlock Item"));
            _userGuideUI.SetActive(false);
            _gameOver_Menu.SetActive(false);
            _unlocktypeScreen.SetActive(true);
            _unlockItemImage.sprite = _air;
        }
        else
        {
            // Level 4
            // No cut seen
            StartCoroutine(StartCountDown()); // Start count down
timer.
            _cutSeenTitle.text = "";
        }
    }

    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
    }

    public void OnClickPauseMenu()
    {
        SoundManager.Instance.PlayButtonClickSound();
        if (!_isPauseGame)
        {
            _pauseMenu.SetActive(true);
            _isPauseGame = true;
        }
    }

    public void OnClickBackToMenu()
    {
        SoundManager.Instance.PlayButtonClickSound();
        SceneManager.LoadScene(3);
    }

    public void OnClickPlayGame()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _pauseMenu.SetActive(false);
        StartCoroutine(StartCountDown());
    }

    public void OnClickDailyProgress_Btn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _dailyProgressDisplay.SetActive(true);
        _AchieveDisplay.SetActive(false);
        // Play Animation.
        _menuAnim.SetBool("Show", true);
        _title.text = "Daily Progress";

        // Display Attention.
        _attentionProgressUI.fillAmount =
ScoreManager.Instance._attentionProgress;

        // Display Concentration.
        _concentrationProgressUI.fillAmount =
ScoreManager.Instance._concentrationProgress;
    }

    public void OnClickBackBtn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        SoundManager.Instance.PlayButtonClickSound();
        _menuAnim.SetBool("Show", false);
    }

    public void OnClickGotItBtn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        SoundManager.Instance.PlayButtonClickSound();
        _userGuideUI.SetActive(false);
        _unlocktypeScreen.SetActive(false);
        _cutSeenAnim.SetBool("PopUp", false);
        _cutSeenTitle.text = "";
        //_cutSeenScreen.SetActive(false);
        StartCoroutine(StartCountDown()); // Start count down timer.
    }

    // Count down timer.
    public IEnumerator StartCountDown()
    {
        _dispalyCountDown.enabled = true;

        for (int i = _timerStart; i >= 0; i--)
        {
            if(i != 0)
            {
                _dispalyCountDown.text = i.ToString();
            }
            else
            {
                _dispalyCountDown.text = "GO!";
            }
            yield return new WaitForSeconds(1.0f);

            if(i == 0)
            {
                _isPauseGame = false;
                _dispalyCountDown.enabled = false;

MissionController.Instance.SetLevelVehicleTypes(MissionController.Insta
nce._level);
                SoundManager.Instance._backGround.Play();
            }
        }
    }

    // Game over.
    public void GameOver()
    {
        SoundManager.Instance._backGround.Stop();
        SoundManager.Instance.GameOverSound();
        _userGuideUI.SetActive(false);
        _unlocktypeScreen.SetActive(false);
        _gameOver_Menu.SetActive(true);
        _pauseBtn.SetActive(false);
        StartCoroutine(SetCutSeen("Game Over"));

        // Set UI items.
        _completedLevelNumber.text = "Level " +
MissionController.Instance._previousLevelIndex;
        _completedStepNumber.text = "Step " +
MissionController.Instance._previousStep;
        //int totalScore = (int)ScoreManager.Instance._gameTime;
        //_score.text = totalScore.ToString();
        _correct.text =
ScoreManager.Instance._correctTriesCount.ToString();
        _incorrect.text =
ScoreManager.Instance._incorrectCount.ToString();
        _missed.text = ScoreManager.Instance._missedCount.ToString();

        ScoreManager.Instance.UpdatePlayerData();
        // Update speed Without ML
        if (ScoreManager.Instance._attentionProgress < 0.5f)
        {
            _speed = VehicleGenerator.Instance._speed -
((MissionController.Instance._speedAccelator / 2) -
MissionController.Instance._speedAccelator *
ScoreManager.Instance._attentionProgress);

            if(_speed < VehicleGenerator.Instance._minSpeed)
            {
                _speed = VehicleGenerator.Instance._minSpeed;
            }
        }
        else
        {
            _speed = VehicleGenerator.Instance._speed +
(MissionController.Instance._speedAccelator *
ScoreManager.Instance._attentionProgress);
        }
        GameData.Instance.updateGameData(_speed,
(MissionController.Instance._requiredVehicleCount +
MissionController.Instance._vehicleCountIncrese), DateTime.Now);


        // With ML

//GameData.Instance.updateGameData((VehicleGenerator.Instance._speed +
(GameData.Instance.chooseDifficulty(GameData.Instance.qlearning()))),
(MissionController.Instance._requiredVehicleCount +
MissionController.Instance._vehicleCountIncrese), DateTime.Now);
    }

    public IEnumerator SetCutSeen(string title)
    {
        _isPauseGame = true;
        _cutSeenAnim.SetBool("Title", true);
        _cutSeenTitle.text = title;

        yield return new WaitForSeconds(2.0f);
        _cutSeenAnim.SetBool("PopUp", true);
    }

    public void OnClickAchievementsBtn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _AchieveDisplay.SetActive(true);
        _dailyProgressDisplay.SetActive(false);
        // Play Animation.
        _menuAnim.SetBool("Show", true);
        _title.text = "Achievement";

        if (PlayerPrefs.GetString("Game_1_Over") == "True")
        {
            _showLevel.SetActive(false);
            _endGame1.SetActive(true);
        }
        else
        {
            _showLevel.SetActive(true);
            _endGame1.SetActive(false);
            _nextLevel.text = "Level " +
PlayerPrefs.GetInt("Level_Number").ToString();
            _nextStep.text = "Step " +
PlayerPrefs.GetInt("Step_Number").ToString();
        }
    }

    public void OnClickSwitchLanguage()
    {
        SoundManager.Instance.PlayButtonClickSound();
        if (_sinhalaPage.activeInHierarchy)
        {
            _englishPage.SetActive(true);
            _sinhalaPage.SetActive(false);
        }
        else if (_englishPage.activeInHierarchy)
        {
            _sinhalaPage.SetActive(true);
            _englishPage.SetActive(false);
        }
    }
}
```

- Player Authentication

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using PlayFab;
using PlayFab.ClientModels;
using TMPro;
using UnityEngine.UI;

public class PlayerAuthentication : MonoBehaviour
{
    [Header("Input Fields")]
    public string _userName;
    public string _password;
    public string _patientName;
    public string _patientAge;
    public string _patientResult;
    public string _careGiverName;
    public string _careGiverMail;
    public string _doctorName;
    public string _doctorMail;

    [Header("Sign In Input Fields")]
    public TMP_InputField _userNameSignIn;
    public TMP_InputField _passwordSignIn;

    [Header("Sign Up Input Fields")]
    public List<TMP_InputField> _playerDataInputs;
    public TMP_InputField _userNameSignUp;
    public TMP_InputField _passwordSignUp;

    public GameObject _loadingImage;
    public Animator _anim;

    public TextMeshProUGUI _notificationBar;
    public string _playFabID;
    public string _attention;
    public string _concentration;

    public static PlayerAuthentication Instance;
    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
    }

    private void Start()
    {
        //PlayerPrefs.DeleteAll();
        _notificationBar.text = "";

        //Note: Setting title Id here can be skipped if you have set the value in Editor Extensions already.
        if (string.IsNullOrEmpty(PlayFabSettings.TitleId))
        {
            PlayFabSettings.TitleId = "789AA"; // Please change this value to your own titleId from PlayFab Game Manager
        }
        //var request = new LoginWithCustomIDRequest { CustomId = "GettingStartedGuide", CreateAccount = true };
        //PlayFabClientAPI.LoginWithCustomID(request, OnLoginSuccess, OnLoginFailure);

        //if (PlayerPrefs.HasKey("UserName"))
        //{
        //    _userName = PlayerPrefs.GetString("UserName");
        //    _password = PlayerPrefs.GetString("Password");

        //    var req = new LoginWithPlayFabRequest { Username = _userName, Password = _password };
        //    PlayFabClientAPI.LoginWithPlayFab(req, OnLoginSuccess, OnLoginFailure);
        //}
    }

    public void OnClickCreateAccount()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _anim.SetBool("Show", true);
        _notificationBar.text = "";
    }
    public void OnClickBackBtn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        // Back Btn in signUp page
        _anim.SetBool("Show", false);
        _notificationBar.text = "";
    }

    #region Player Registration

    public void OnClickSignUp()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _notificationBar.text = "";
        _loadingImage.SetActive(true);
        bool empty = false;
        for(int i = 0; i < _playerDataInputs.Count ; i++)
        {
            if(_playerDataInputs[i].text == "")
            {
                empty = true;
            }
        }

        if(!empty)
        {
            if(_userName.Length <= 3)
            {
                _loadingImage.SetActive(false);
                _notificationBar.text = "Username should have atleast 4 
characters";
            }
            else
            {
                var req = new RegisterPlayFabUserRequest { Username = 
_userName, Password = _password , RequireBothUsernameAndEmail = false};
                PlayFabClientAPI.RegisterPlayFabUser(req, 
OnRegisterSuccess, OnRegisterFailure);
            }
        }
        else
        {
            _loadingImage.SetActive(false);
            _notificationBar.text = "Please fill the required fields.";
        }
    }

    public void OnClickSignIn()
    {
        SoundManager.Instance.PlayButtonClickSound();
        _notificationBar.text = "";
        _loadingImage.SetActive(true);

        if (_userName != "" && _password != "")
        {
            var req = new LoginWithPlayFabRequest { Username =
_userName, Password = _password};
            PlayFabClientAPI.LoginWithPlayFab(req, OnLoginSuccess,
OnLoginFailure);
            PlayerPrefs.SetString("UserName", _userName);
            PlayerPrefs.SetString("Password", _password);
        }
        else
        {
            _loadingImage.SetActive(false);
            _notificationBar.text = "Please fill the required fields.";
        }
    }

    private void OnLoginSuccess(LoginResult result)
    {
        print("login success");
        SceneManager.LoadScene(2);
        _playFabID = result.PlayFabId;
        GetUserData(_playFabID);
    }

    private void OnLoginFailure(PlayFabError error)
    {
        if (PlayerPrefs.HasKey("UserName"))
        {
            // invalid
            _loadingImage.SetActive(false);
            _notificationBar.text = "Invalid username or password.";
        }
        else
        {
            // create account
            _loadingImage.SetActive(false);
            _notificationBar.text = "Please sign up before sign in.";
        }

        ClearSignInPage();
    }

    private void OnRegisterSuccess(RegisterPlayFabUserResult result)
    {
        Debug.Log("Registered!");

        // Update palayer data.
        PlayerPrefs.SetString("UserName", _userName);
        PlayerPrefs.SetString("Password", _password);
        PlayerPrefs.SetString("PatientName", _patientName);
        PlayerPrefs.SetString("PatientAge", _patientAge);
        PlayerPrefs.SetString("PatientResult", _patientResult);
        PlayerPrefs.SetString("CareGiverName", _careGiverName);
        PlayerPrefs.SetString("CareGiverMail", _careGiverMail);
        PlayerPrefs.SetString("DoctorName", _doctorName);
        PlayerPrefs.SetString("DoctorMail", _doctorMail);

        SetUserData();

        SceneManager.LoadScene(2);
    }

    private void OnRegisterFailure(PlayFabError error)
    {

        // already exists
        //Debug.LogError(error.GenerateErrorReport());
        _loadingImage.SetActive(false);
        _notificationBar.text = "Username or password you entered is
already exsist.";

        ClearSignUpPage();
    }

    #endregion

    #region Get Input Fields

    public void GetUserNameSignUp(string userName)
    {
        _userNameSignUp.text = _userNameSignUp.text.Replace(" ", "");
        _userName = _userNameSignUp.text;
        print(_userName);
    }

    public void GetPasswordSignUp(string password)
    {
        _passwordSignUp.text = _passwordSignUp.text.Replace(" ", "");
        _password = _passwordSignUp.text;
    }

    public void GetUserNameSignIn(string userName)
    {
        _userNameSignIn.text = _userNameSignIn.text.Replace(" ", "");
        _userName = _userNameSignIn.text;
    }

    public void GetPasswordSignIn(string password)
    {
        _passwordSignIn.text = _passwordSignIn.text.Replace(" ", "");
        _password = _passwordSignIn.text;
    }

    public void GetPatientName(string patientName)
    {
        _patientName = patientName;
    }

    public void GetPatientAge(string patientAge)
    {
        _patientAge = patientAge;
    }

    public void GetPatientResult(string result)
    {
        _patientResult = result;
    }

    public void GetCareGiverName(string name)
    {
        _careGiverName = name;
    }

    public void GetCareGiverMail(string mail)
    {
        _careGiverMail = mail;
    }
```

```
public void GetDoctorName(string name)
{
    _doctorName = name;
}

public void GetDoctorMail(string mail)
{
    _doctorMail = mail;
}

#endregion

#region Clear Input Data

public void ClearSignUpPage()
{
    _userNameSignUp.text = "";
    _passwordSignUp.text = "";
}

public void ClearSignInPage()
{
    _userNameSignIn.text = "";
    _passwordSignIn.text = "";
}

#endregion

public void SetUserData()
{
    PlayFabClientAPI.UpdateUserData(new UpdateUserDataRequest()
    {
        Data = new Dictionary<string, string>() {
        {"Patient Name", _patientName},
        {"Patient Age", _patientAge},
        {"Patient Result", _patientResult},
        {"CareGiver Name", _careGiverName},
        {"CareGiver Mail", _careGiverMail},
        {"Doctor Name", _doctorName},
        {"Doctor Mail", _doctorMail},
        {"Attention", ""},
        {"Concentration", ""}
    }
    },
    result => Debug.Log("Successfully updated user data"),
    error => {
        Debug.Log("Failed");
        Debug.Log(error.GenerateErrorReport());
    });
    }

    public void GetUserData(string myPlayFabeId)
    {
        PlayFabClientAPI.GetUserData(new GetUserDataRequest()
        {
            PlayFabId = myPlayFabeId,
            Keys = null
        }, result => {
            Debug.Log("Got user data:");
            if (result.Data == null) {
                Debug.Log("No data");
            }
            else
            {
                //Debug.Log("Ancestor: " +
result.Data["Ancestor"].Value);
                // Update palayer data.
                PlayerPrefs.SetString("PatientName",
result.Data["Patient Name"].Value);
                PlayerPrefs.SetString("PatientAge",
result.Data["Patient Age"].Value);
                PlayerPrefs.SetString("PatientResult",
result.Data["Patient Result"].Value);
                PlayerPrefs.SetString("CareGiverName",
result.Data["CareGiver Name"].Value);
                PlayerPrefs.SetString("CareGiverMail",
result.Data["CareGiver Mail"].Value);
                PlayerPrefs.SetString("DoctorName", result.Data["Doctor
Name"].Value);
                PlayerPrefs.SetString("DoctorMail", result.Data["Doctor
Mail"].Value);
                _attention = result.Data["Attention"].Value;
                _concentration = result.Data["Concentration"].Value;
            }
        }, (error) => {
            Debug.Log("Got error retrieving user data:");
            Debug.Log(error.GenerateErrorReport());
        });
    }
    public void UpdateUserProgressData(int atention,int concentration)
    {
        _attention = _attention  + atention.ToString() + ", ";
        _concentration = _concentration  + concentration.ToString() +
", ";

        PlayFabClientAPI.UpdateUserData(new UpdateUserDataRequest()
        {
            Data = new Dictionary<string, string>() {
            {"Attention", _attention},
            {"Concentration", _concentration}
        }
        },
    result => Debug.Log("Successfully updated user data"),
    error => {
        Debug.Log("Failed");
        Debug.Log(error.GenerateErrorReport());
    });
    }
}
```

- Send Email

```
using UnityEngine;
using System.Net;
using System.Net.Mail;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;

public class SendMail : MonoBehaviour
{
    public static SendMail Instance;

    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
    }
    public void SendEmail()
    {
        MailMessage mail = new MailMessage();
        SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
        SmtpServer.Timeout = 10000;
        SmtpServer.DeliveryMethod = SmtpDeliveryMethod.Network;
        SmtpServer.UseDefaultCredentials = false;
        SmtpServer.Port = 587;

        mail.From = new MailAddress("careumed2020@gmail.com");
        mail.To.Add(new
MailAddress(PlayerPrefs.GetString("DoctorMail")));
        mail.CC.Add(new
MailAddress(PlayerPrefs.GetString("CareGiverMail")));

        mail.Subject = "Week " + PlayerPrefs.GetInt("Level_Number") + "
report of " + PlayerPrefs.GetString("PatientName");
        // Body start.
        mail.Body = "Patient Name   : " +
PlayerPrefs.GetString("PatientName") + "\n" +
                    "Patient Age    : " +
PlayerPrefs.GetString("PatientAge") + "\n" +
                    "Patient Mail   : " +
PlayerPrefs.GetString("PatientMail") + "\n" +
                    "CareGiver Name : " +
PlayerPrefs.GetString("CareGiverName") + "\n" +
                    "CareGiver Mail : " +
PlayerPrefs.GetString("CareGiverMail") + "\n" + "\n" + "\n" +
                    "Attention And Concentration" + "\n" + "\n";

        if(PlayerPrefs.GetInt("Level_Number") == 1)
        {
            for (int i = 0; i < PlayerPrefs.GetInt("Week1Day"); i++)
            {
                mail.Body += "Day " + (i + 1).ToString() + " => " +
"Attention " + ((int)(PlayerPrefs.GetFloat("AttentionWeek1" + i) *
100)).ToString() + "%" +" , "+ "Concentration " +
((int)(PlayerPrefs.GetFloat("ConcentrationWeek1" + i) *
100)).ToString() + "%" + "\n";
            }
        }
        if (PlayerPrefs.GetInt("Level_Number") == 2)
        {
            for (int i = 0; i < PlayerPrefs.GetInt("Week2Day"); i++)
            {
                mail.Body += "Day " + (i + 1).ToString() + " => " +
"Attention = " + ((int)(PlayerPrefs.GetFloat("AttentionWeek2" + i) *
100)).ToString() + "%" + " , " + "Concentration = " +
((int)(PlayerPrefs.GetFloat("ConcentrationWeek2" + i) *
100)).ToString() + "%" + "\n";
            }
        }
        if (PlayerPrefs.GetInt("Level_Number") == 3)
        {
            for (int i = 0; i < PlayerPrefs.GetInt("Week3Day"); i++)
            {
                mail.Body += "Day " + (i + 1).ToString() + " => " +
"Attention = " + ((int)(PlayerPrefs.GetFloat("AttentionWeek3" + i) *
100)).ToString() + "%" + " , " + "Concentration = " +
((int)(PlayerPrefs.GetFloat("ConcentrationWeek3" + i) *
100)).ToString() + "%" + "\n";
            }
        }
        if (PlayerPrefs.GetInt("Level_Number") == 4)
        {
            for (int i = 0; i < PlayerPrefs.GetInt("Week4Day"); i++)
            {
                mail.Body += "Day " + (i + 1).ToString() + " => " +
"Attention = " + ((int)(PlayerPrefs.GetFloat("AttentionWeek4" + i) *
100)).ToString() + "%" + " , " + "Concentration = " +
((int)(PlayerPrefs.GetFloat("ConcentrationWeek4" + i) *
100)).ToString() + "%" + "\n";
            }
        }
        // Body end.

        SmtpServer.Credentials = new
System.Net.NetworkCredential("careumed2020@gmail.com", "careu2020") as
ICredentialsByHost; SmtpServer.EnableSsl = true;
        ServicePointManager.ServerCertificateValidationCallback =
delegate (object s, X509Certificate certificate, X509Chain chain,
SslPolicyErrors sslPolicyErrors)
        {
            return true;
        };

        mail.DeliveryNotificationOptions =
DeliveryNotificationOptions.OnSuccess;
        SmtpServer.Send(mail);
```

```
    }
}
```