Augmented Reality Game Targeting Children with Cerebral Palsy

Team Members:

Aaruni Arora

Shreya Das

Cassius Kua

Pranathi Poojary

Chun Tang

Supervisor:

Ian Radcliffe

Contents

Introduction

The aim of the project was to build an educational game in association with our client, the Pace Centre. The Pace Centre runs a school for children with neurodisabilities and they are looking for an educational game featuring augmented reality (AR) for students with cerebral palsy (CP) to promote healthy movement and fun learning. Throughout the project, our point of contact with the Pace Centre was Luke Thompson who provided us with valuable information regarding the user requirements and past stories.

User Requirements

The most important requirements for the game were that it had to be fun and engaging, followed by it being educational, and if possible, for it to be physically engaging.

Cerebral Palsy

Cerebral palsy is a neurodisability that is caused by a foetus' brain not developing normally before, during, or after their birth. It is a lifelong condition and impacts each individual differently.

There are 4 types of CP: spastic, dyskinetic, ataxic, and mixed which classifies which part of the body's motor function is impacted by the condition. The degree of motor function impairment is classified using the Gross Motor Function Classification System (GMFCS) ranging from Level 1 to 5 with Level 5 having the most motor function impairment.

We decided to target students on level 4 of the GMFCS as the Pace Centre mainly works with students at levels 3-5. Level 5 students are hard to accommodate due to their restricted motor function, but we aim to make the game scalable to try and make it as inclusive as possible.

Physical therapy

Many cerebral palsy students at The Pace Centre are wheelchair-bound, therefore we will be focusing on upper-body movements, such as the arms, neck, and torso. We also want to limit compensatory movements, which are unhealthy for the students in the long-term. These movements are caused by students moving in a way to compensate for their restricted motor function, and could become an unhealthy habit. These movements vary from student to student, so we will need to be mindful of what movements our game will incorporate.

Available technology

The most readily available device The Pace Centre has is the Surface Pro (which runs on Windows) and can be mounted on all students' wheelchairs. The HoloLens was considered before, but this isn't readily available at The Pace Centre. Moreover, headsets are heavy and can cause fatigue. In addition, due to the lack of access to specialised AR/VR devices at the Pace Centre, we have chosen the Surface Pro as our device of focus.

To build the game two game engines were considered: Unity and Unreal Engine. We decided to develop the game in Unity due to the large amount of resources available on learning Unity. Moreover, Unreal Engine is more suitable for complex 3D games. Unity is sufficient for our requirements.

How the students are grouped

The students are grouped according to their intellectual abilities, rather than their CP type/severity or age. Thus, we will place emphasis on the educational aspects of our games while referencing their primary and secondary education content. Ideally we would want to cover all possible content, but due to time constraints we narrowed down our target educational level to Key Stage 1 and 2 students.

Game Topic

The Pace Centre's curriculum is unique, they wish to integrate various subjects and occasionally include physical therapy as part of their lessons.

Since the Pace Centre works with students aged 3-16, the range of topics we can choose is very wide. Ideally, our game would have a difficulty setting so that it is useful for as many students as possible.

Our target is younger students, thus simpler educational topics would be the most appropriate. For example: colours, shapes, animals or counting. With a longer time frame, we could incorporate more concepts and integrate arts or sports into the lesson plans.

Individual or Group Game

During our meeting, group games were brought up as a potential interesting direction for the game. This would encourage the students to communicate, and potentially make the game more engaging. However, this isn't one of our core requirements.

Accommodating for the school environment

Upon receiving the floor plans of the 2 building of their campus, we have decided to accommodate for the school environment. To make the AR game more engaging and interactive, the corridors and outside areas can be utilised for lessons and our game. Hence, with the information provided, we can allocate specific areas to help teachers execute their lessons more effectively and efficiently.

Merit recognition and grading system

This system would help encourage students when they are doing well, and help them keep track of their own progress. Students can collect coins during the game and use these to buy new avatars.

Progress Tracking

The teachers are required to write regular reports about each student, since this is a specialised school. Introducing progress tracking to our game would be very beneficial to the teachers, as it can be incorporated into their reports.

User Stories

These user stories helped us tailor our final product to our users' needs.

Certain topics may not be suitable for CP students

One of the past projects that The Pace Centre collaborated on was a game that focused on teaching students about food and nutrition. Despite it being an interesting concept, some students require a feeding tube and hence felt excluded.

Whilst carefully curating the topic of our game and maintaining the aspect of fun, we decided to keep it simple with animals and educational topics in math.

Cerebral Visual Impairment (CVI)

CVI is a disorder some of the students have to deal with. Accessibility is a very important aspect of our game, as we want the game to be fun for as many students as possible. Including a high contrast mode would be the most practical way around this problem.

Lesson Time

We have determined that a lesson at The Pace Centre's school, which typically lasts for one hour, needs to consider factors such as students settling in, teacher introduction, stretching, and post-lesson discussion. Additionally, we've considered fatigue levels when using headsets, which limits playing time to 10-20 minutes. Based on these factors, we plan on designing the game to last 10 minutes.

Progress Tracking Ease of Use

Our goal is to give teachers progress data on student engagement and performance through our game-based approach. This will be done using a user-friendly design, with data that can be easily incorporated into regular reports, in simple formats such as graphs and tables. We also need to address potential challenges of unfamiliarity with technology to facilitate adoption.

Case Diagram

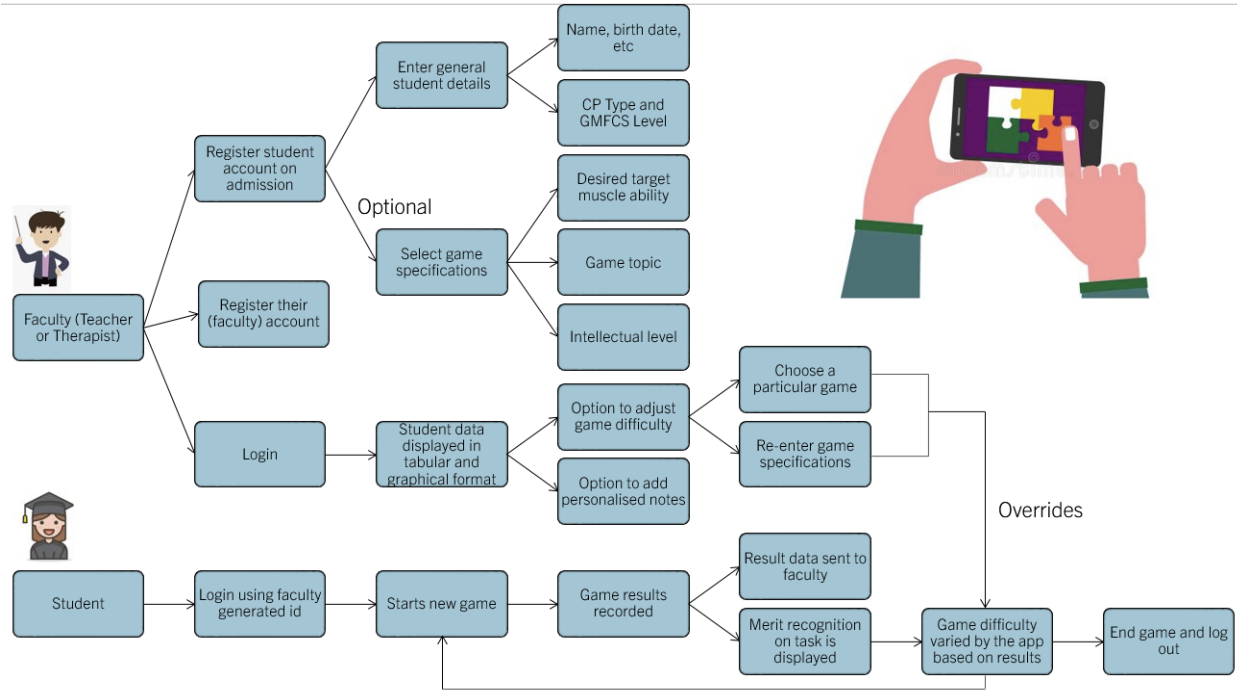


Figure 1: Case diagram showing how the product will be used

There are two possible types of accounts: Student and Faculty accounts. Both accounts must be made by faculty at the Pace Centre as it might be difficult for younger students to register themselves. However, logging into the game will be done by the student.

Faculty members will have access to all student information. Based on a child's specific GMFCS types, intellectual and physical abilities, the teachers or therapists should be able to change the game to target specific learning topics of choice, level of difficulty and muscle groups. Through the faculty account, they can access the student data to track the student's progress and override the game parameters to choose an appropriate level for each individual.

As described in the student pathway above, each student will have a faculty generated ID which will be their unique in-game ID. Upon successfully logging in and completing the game, their results will be recorded and shared with the faculty. At the end of each game merit recognition will be given based on their game score. The app should then ideally suggest a difficulty level for the student's next game and will loop through the diagram above from 'Starts new game'.

Ultimately our goal is to optimise students' learning and to interlink physical and intellectual progress and efficiently track it for the faculty.

The Final Product

Designing the game

Our game consists of 2 major parts, (1) an account and database system that uses the Firebase utility, and (2) the augmented reality game, DinoMaze, all created using Unity.

Firebase Account System

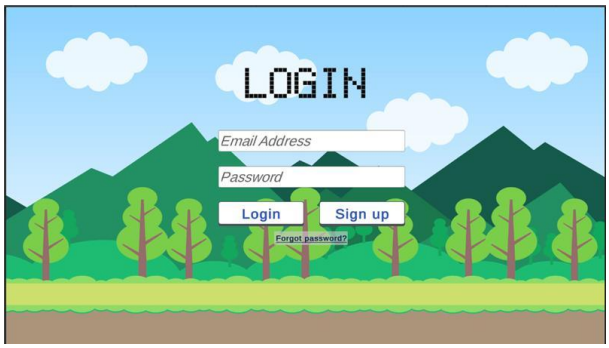
The account system is important for the game as a way to track each student's progress. The reason we chose to use Firebase is because they have easy to use SDKs designed specifically for Unity with authentication and database features i.e. Firebase Authentication and Firebase Realtime Database.

A user is registered on the Register UI. Both types of accounts, student and faculty, will require a username, email, and password. A student account will in additon allow optional fields to specify the student's GMFCS level, CP type, and age. The additional parameters for students will be helpful for tailoring the user experience to each student, and be presented as part of the students' progress tracking for teachers and therapists to see. Successful registration will create a user account on the Firebase Authentication system and upload user data (username, email, age, GMFCS level, CP type) to the Firebase Realtime Database system.

A registered user can login on the Login UI. The code checks if the user exists using Firebase Authentication. Successful login will display the Start Page, download the user data from the Firebase Realtime Database (e.g. their username, their score from previous attempts as a further improvement, etc.) and allow the player to begin the game.

A user can also reset their password however this feature is currently under development. This will email the user instructions on resetting their password.

Since this system will only be used internally at The Pace Centre, we decided an email verification system would not be necessary, especially as each student will be registered by a member of the staff.



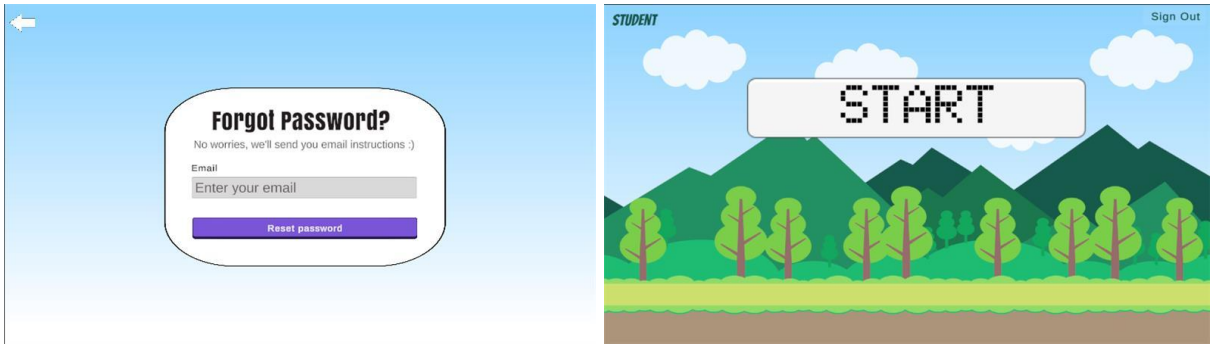


Figure 2: The 4 images above show the UI for different screens, Login, Register, Forgot Password and Start. The Register UI contains dropdown menus for the CP Type and GMFCS Level.

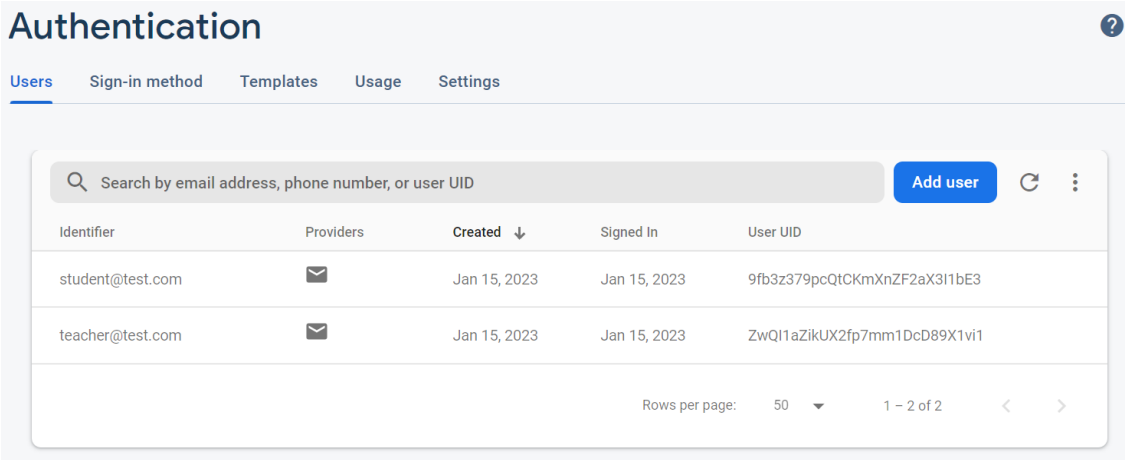


Figure 3: The authentication page on the firebase console showing the registered users.

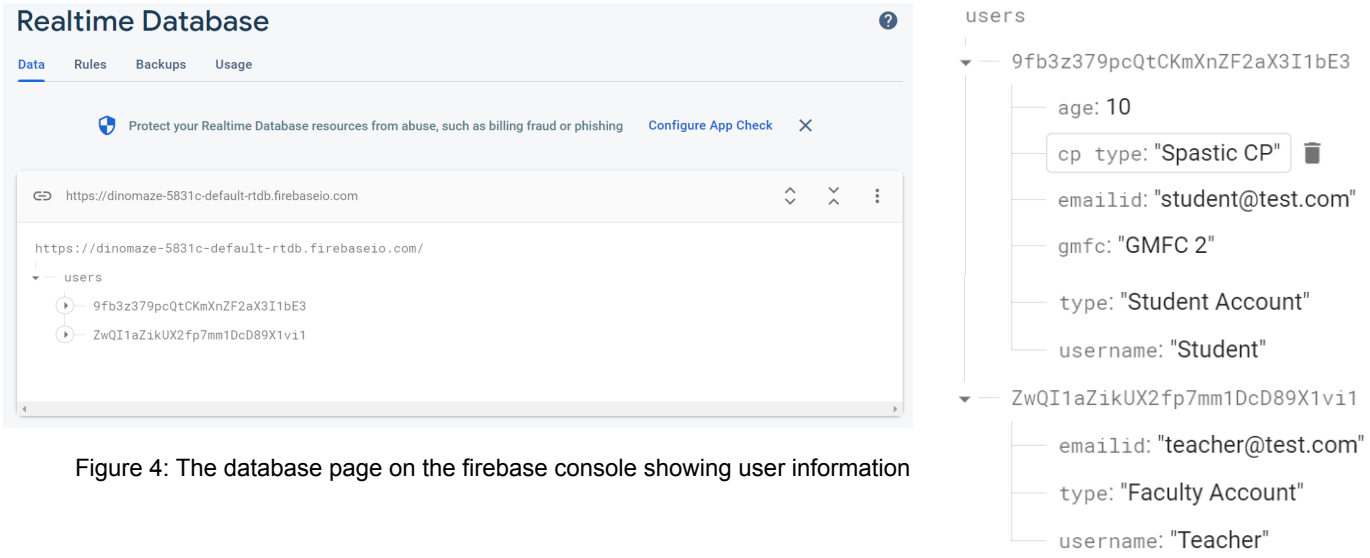


Figure 4: The database page on the firebase console showing user information

AR DinoMaze

Once the user is logged in, they are directed to the Main Menu Page. Here the user has an option to select levels, access the shop to buy avatars, or go to settings. Once they select a level within the minigame they want to play, the device's camera switches on to detect for the specific image. Since this is an AR based game, we have obtained a Wikitude licence to use their Camera. The user just has to hover the device's camera over an image to start playing the mini game. Different mini games would be associated with different images. Teachers can place these images around the room or campus at different heights to vary difficulty and encourage mobility. This could further lead to students communicating with each other, promoting group activity during the session to find out where the placards are.

Currently, we have one mini game - an obstacle course where the user controls a dinosaur avatar, Dino, to collect coins and try to reach the end of the course. You can control the avatar using an on-screen direction-pad (joystick) that responds to touch or with arrow keys on a laptop. This is important since some students would need to use touch or eye-tracking input devices. The front and back arrow keys allow translation in z-axis and the left and right arrows trigger rotation. This is because we wanted the controls to be restricted to minimise the potential for error, since we need to accommodate for the limited mobility of the students. This set of controls are easy to use, and forces the user to move carefully.

When the avatar hits an obstacle, the object will turn red. After hitting 3 obstacles, the user will be required to answer a multiple choice question. The game pulls from a database of pre-made questions, and displays a prompt over their game. After they answer the question correctly, the avatar will be sent to the nearest checkpoint, either the start of the level or where they picked up their most recent coin. A 6th trigger of this (hitting 18 obstacles) will result in 'game over', the mini game will reset and the user can choose to restart the mini game.

The coins collected during the mini game are the reward system. The user can spend the coins they collected during the game at the Shop to buy avatars. They can then change their avatar to the newly bought avatar, however this feature is currently being developed. The Shop is included to incentivise students to play the game.

Some extra features are the pause menu screen and the settings menu. The pause menu serves the purpose of allowing the students to take intermittent breaks from the game so as to not over-exert themselves physically and provide them with the time needed to recharge. The settings menu is still work in progress. Once developed, it should be able to control volume, vary the font size, etc.

Figure 5: Top - Menu screen displayed from the Start page. Bottom - Difficulty level of the game can be picked; currently only level 1 has been implemented. Right - Pausing the game

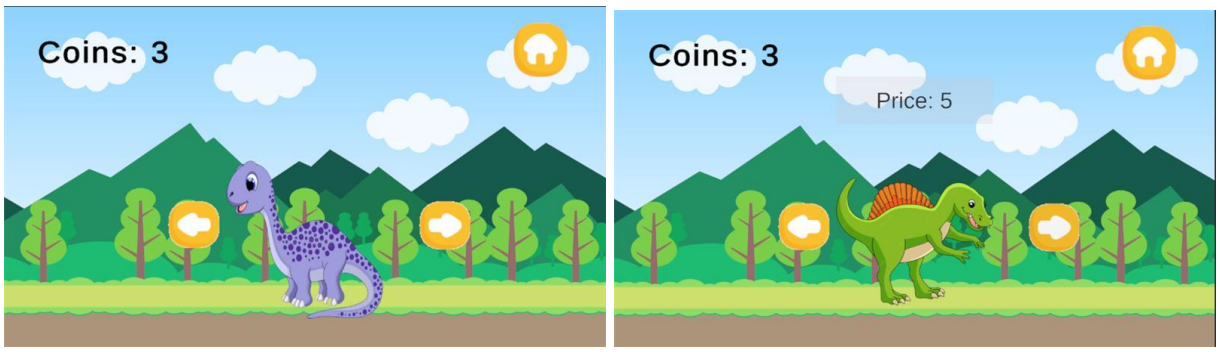


Figure 6: The shop menu - students can spend the coins they earn through playing the game; currently there is not an avatar changing function, so buying a new dinosaur would not change the gameplay.

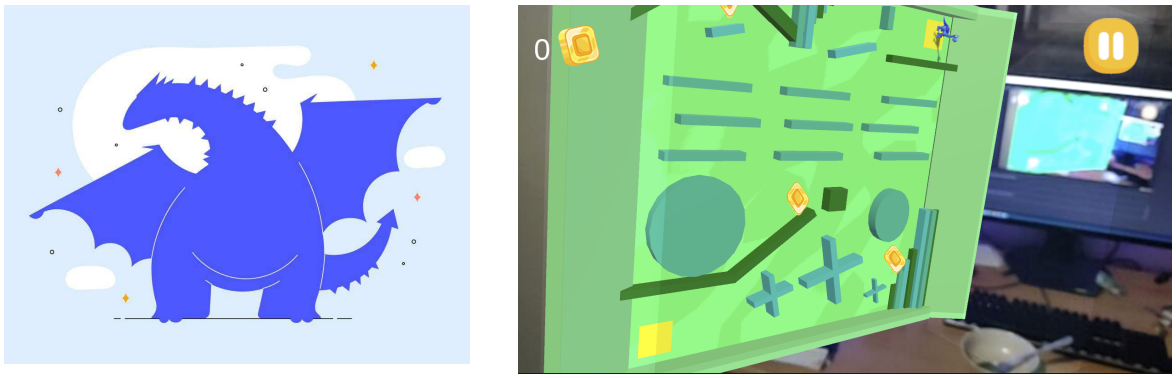


Figure 7: The dinosaur image is scanned by the user to load the first mini AR game.

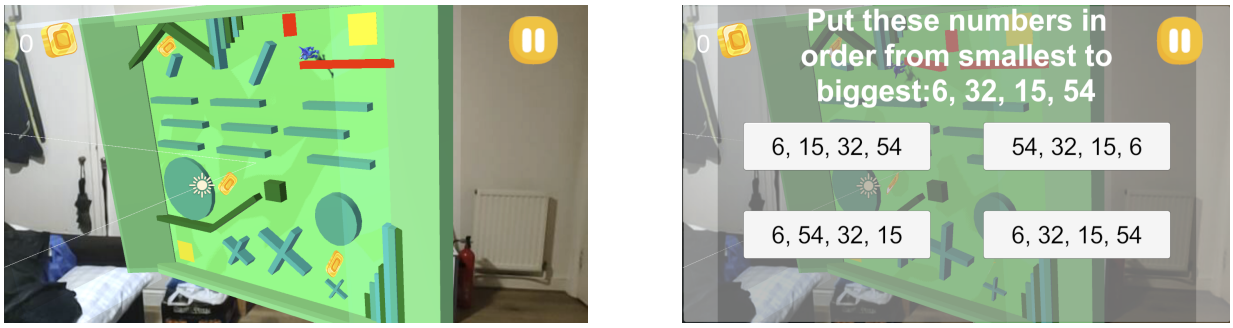
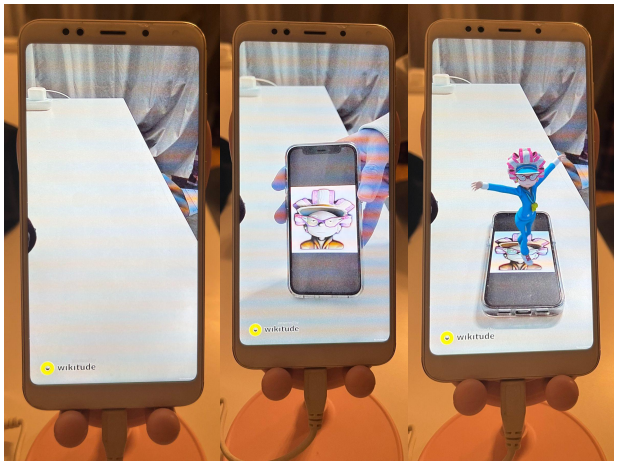


Figure 8: Objects that have been hit by the avatar turn red. Questions pop up over the maze after every 3 hits.

Android App

Unity has features for android app development. Settings in Unity were changed to download the app onto an Android device. The app was installed into the android device via USB debugging after which the app could work independently (without connection to the computer).

Figure 9: Left - The app, Dino Maze, shown on the mobile device. Top - Login page. Bottom - Difficulty level selection



The actual game is made for a larger screen (SurfacePro) and so is not well scaled to be seen on this mobile device.

Figure 10: An example image of ‘Granny’ using the Wikitude Camera on a mobile device.

Class Diagrams

Firebase class diagram

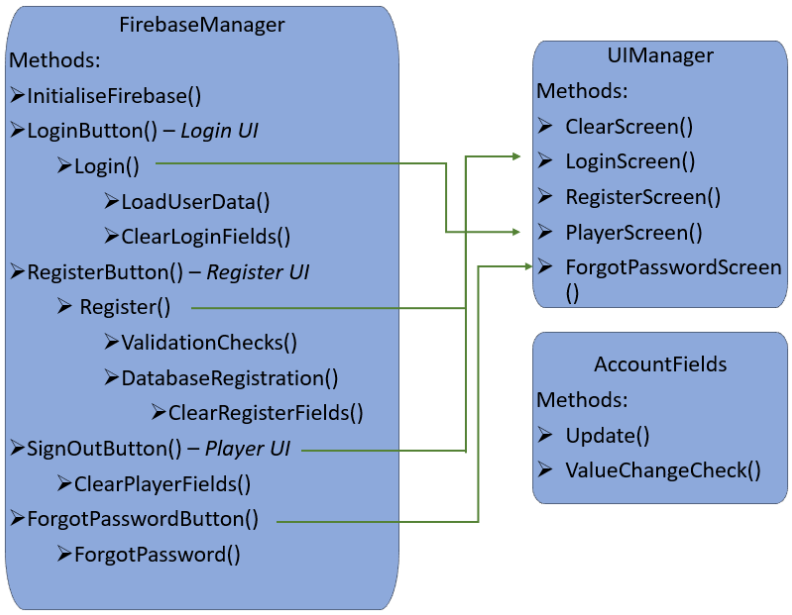


Figure 11: Class diagram for Firebase Dependencies. Three scripts are shown in purple. Green arrows are called upon success. Parts of the code are run through buttons on the UI and so are not displayed here.

AR DinoMaze class diagram

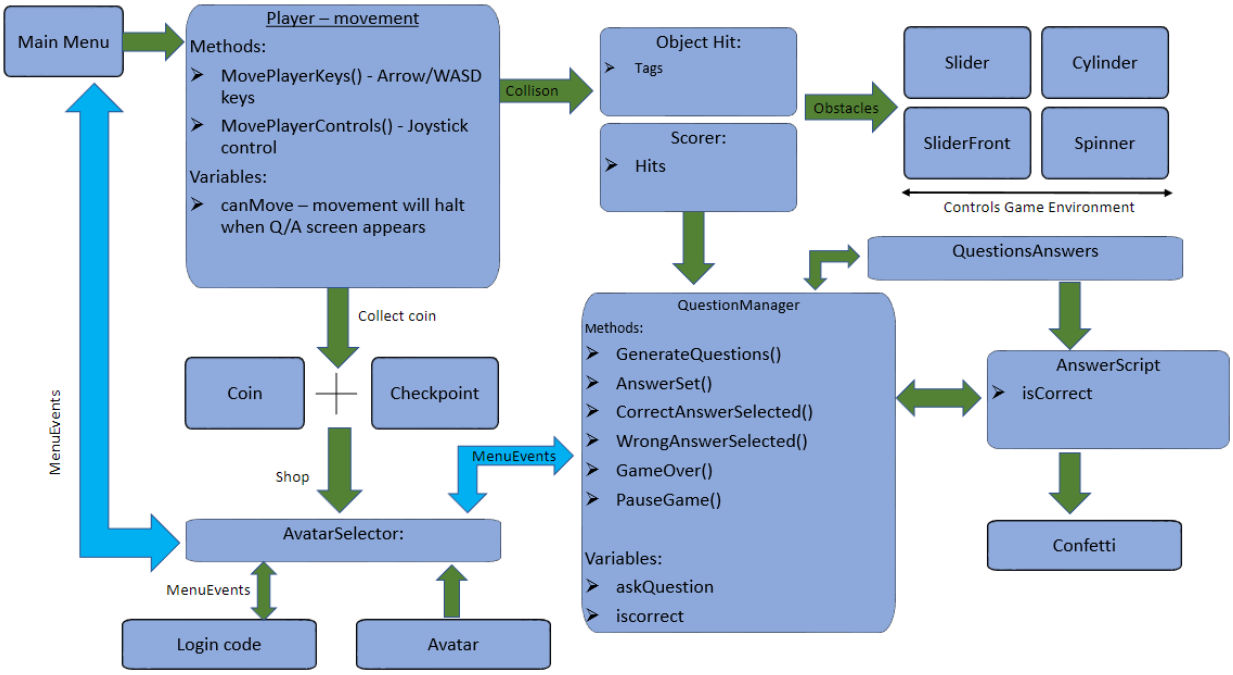


Figure 12: Class diagram for DinoMaze

Description of the Scripts:

- Coin: keeps count of number of coins collected by the user
- Checkpoint: stores the last position of the avatar. This is it is transported after a question is asked
- AvatarSelector: helps with buying selected avatar. You can only buy if your number of coins are sufficient (as per avatar price). Further improvements: play the game with avatars that can be bought
- ObjectHit: turns the objects that have been bumped red using tags so they aren't counted again
- Scorer: keeps score of the number of times the avatar bumps into an object. Accesses the QuestionManager screen after every 3 bumps.
- QuestionManager: is the main game manager. GenerateQuestions() randomises questions, Answerset() assigns answers to each button, CorrectAnswerSelected and WrongAnswerSelected() determines game progression and there is no penalty for wrong answers. GameOver() and PauseGame() are accessing panels in the UI.
- Avatar: creates a new type of object to make a list of avatars.
- QuestionsAnswers: creates a new type of object to create list of Q/A
- AnswerScript: determines whether or not the correct answer is selected
- MenuEvents: is a script that links scenes together, whereas green arrows with text are events

Others:

- Tags: categorise GameObjects and help access multiple objects

<u>Error handling and Validation checks</u>

1] Firebase checks: done using built in Firebase exceptions

These checks are done upon logging in and registering a user.

A corresponding warning message is displayed if the validation check is unsuccessful.

Checks are placed to reject login or registration if there is a:

- Missing email
- Invalid email
- Account existing with that email (for Registration only)
- Missing password
- Weak password (less than 6 letters)

As not all possible error codes have been accounted for, a general 'Register Failed' and 'Login Failed' messages can also be displayed respectively.

2] Custom checks: customised validation checks for appropriate username, password and age

These checks are done upon registering a user.

A corresponding warning message is displayed if registration is unsuccessful.

Checks are placed to reject registration if there is a

- Blank username
- Mismatch between password and confirm password fields
- Age outside the range of 2-100

The optional age field only takes on integer values however the default, placeholder text 'Age' can throw an error if the user decides to leave it as it is. Hence, a try-catch is in place to catch the exceptions.

3] Game:

Player movement error handling: Wrapping the code inside the "MovePlayerKeys" method in a try-catch block to catch any exceptions that may occur.

Unity exception: When this occurs, the UnityEngine throws an error itself. If the vertical and horizontal axes are not well defined in the input, a UnityException will occur when Input.GetAxis() is called.

NullReferenceException: This occurs when you attempt to access an object that is null. If the DinoAnimator method is not attached to a player object, a NullReferenceException will occur.

ArgumentOutOfRangeException: Will happen when an argument that is passed through the method exceeds the maximum or minimum limits. This will output an ArgumentOutOfRangeException when the argument is assigned to the translation or rotation variables, hence to target this a clamping of input values to an allowed range.

Unit Testing

For unit testing, we utilised the built-in Unity Test Framework (UTF) which uses a Unity integration of the NUnit library. NUnit is an open-source unit testing library for .Net languages

We were able to implement two PlayMode unit tests, one for testing the avatar's forward movement, and another for testing its turning. PlayMode tests are done while the whole game is being run.

We were also able to create two EditMode tests, which tests the code within the editor. One test checks whether the function we use to calculate how much the avatar moves when inputting the forward direction works, and the other is very similar, but checks the function for rotating the avatar.

Tests were used throughout development to ensure our core movement mechanics never change.

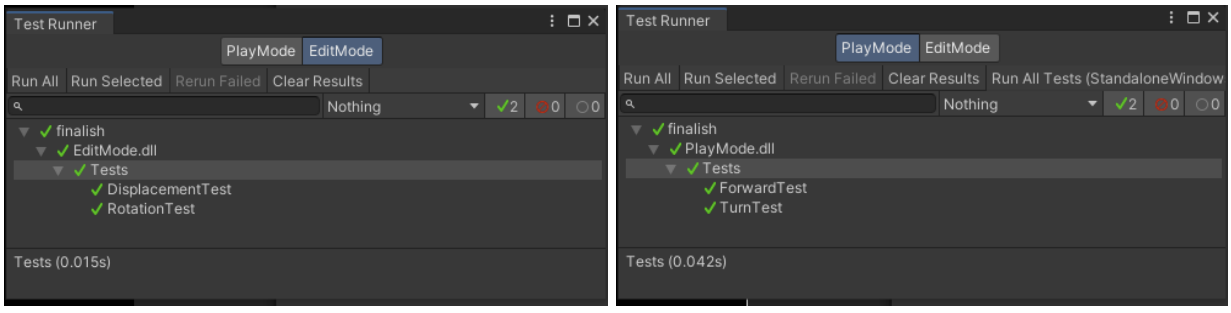


Figure 13: Unity's Test Runner tool shows if a unit test succeeds

Meeting the User Requirements

User requirement	Did we meet it?	Description
Accessible for students with cerebral palsy.	Yes	The game is mainly controlled via touch screen. It would also be compatible with specialist input devices, such as eye trackers, that send mouse inputs.
Able to aid a student's physical therapy.	Yes	Teachers can direct students to play in a way that encourages specific movements. Implementing direct methods of specific motor function will depend on the technology the game is played with.
Uses technology readily available in The Pace Centre school.	Yes	We originally wanted to make a game for Windows, since that's what the Surface Pro runs on, however we decided to develop an Android app. This is because Android apps can be run on Windows using an Android emulator, such as BlueStacks or NoxPlayer. An Android app would therefore be compatible with any Android devices The Pace Centre has, as well as their Surface Pros, while also allowing compatibility with tablets if the Pace Centre chooses to do so.
Suitable for students at around year 1 or 2 education level.	Yes	We were able to create a selection of questions and features suitable for year 1 and 2 students. The questions were based on topics found on BBC Bitesize for year 1 and 2 students. However, we were not able to implement difficulty levels to accommodate for more intelligence levels.
Topics are fun and engaging for CP students.	Needs to be tested	This is a subjective topic, but we chose simple educational questions. While this may not be the most exciting, it shouldn't make any student feel excluded.
Encourages movement around the environment /campus.	No	Unfortunately, due to time constraints, we were only able to implement one level for the game. Thus, there isn't much incentive for moving around the play area. Given more time, we would have implemented more levels, and more scannable images that can then be distributed and the campus/room.
Has a merit recognition system to encourage and engage the student.	Yes	We implemented a coin system to encourage the students to play the game more and earn coins to spend at the Shop. Additionally, confetti upon successful completion of the game is added.
Progress tracking for the teachers to see how the	No	We were not able to connect some player information, such as their coin count important for progress tracking, to the Firebase system due to time constraints.

students are performing, and to be used in reports.		
Accessibility setting for students with cerebral visual impairment.	No	We were not able to implement a high contrast mode. Although this wasn't a high-priority user requirement, it could diminish the enjoyment some of the students can get from our game.
A full game session lasts 20-30 minutes.	No	Currently our one mini game lasts about 1-2 minutes, which is far from our target duration. However, given the nature of our game, scaling the potential play time is quite simple as it's just a matter of implementing more mini games of varying difficulty and associating them with a new scannable image.

Potential improvements

There were many features and parts of the game we had to forego due to time constraints. Given more time, we would like to completely meet our user requirements, and to improve the game overall.

Features undergoing development:

Shop and avatar customisation system

Our shop currently has options to purchase different avatars and they can do so using the points that they earn from completing mini games. We hope to provide more cosmetic changes in the future such as the ability to change avatars and play with different sprites so the students can have a sense of individuality and play with what appeals to them.

Progress tracking

Progress tracking for faculty - tracking both the students' physical therapy and their learning progress - would have been helpful data for teachers and therapists. Additionally, reports of the data in the form of a table or graph for faculty would allow them to optimise their teaching as students progress.

Progress tracking for students would allow them to carry over the progress they made in a previous session of the game, allowing them to retain their points each time they play.

Reset Password

The user should be able to reset their password if they forget it and receive email instructions to reset it. The UI for resetting the password has been developed in our game, however the implementation of sending an email to the provided email is yet to be developed.

High contrast mode

A high contrast mode would make it easy for someone with CP visual impairment to differentiate obstacles and read the questions easier. This would likely be part of a larger settings menu for the user to decide.
Although we have code for the high contrast mode, it needs to be tweaked to be successfully implemented.

Further improvements:

More mini-games

Multiple mini-games with different puzzles (to the obstacle course) could be implemented for a longer and better game duration.

Scaleable level difficulty

Our questions are currently only suitable for young students, at around the year 1 or 2 level. Our format of using multiple choice questions is suitable for all levels, but we were not able to implement a level difficulty setting. Although our UI supports more than 1 difficulty level, the stage design for them is yet to be developed.

Another direction this feature could go is the ability to choose what subjects are being quizzed. This could also be further expanded to a custom question feature, where teachers can create a set of questions according to what the students are currently learning.

Camera improvements

Currently, there is some difficulty with following the avatar in AR while it traverses the level. This may be due to the Wikitude Camera used for AR, however further debugging and research is necessary to determine the cause. Ideally, we would implement a zoom in/out function, so that each user can adjust according to their preference. Currently, the only way to zoom back is to physically move the camera back while facing the scan image.

AR Map

Our AR-based game promotes student movement throughout the building, thus, a system to locate the game pictures using an AR mapping similar to role-playing games, with a quest indicator leading students to their designated game of the day, will be implemented. A UI with directional arrows will be developed for ease of navigation.

How we worked as a team

Overall, everyone contributed to each part of the project, but the following describes the lead of each task:

- Aaruni Arora - the AR game and shop system, GitHub
- Shreya Das - presentation
- Cassius Kua - bug fixing, resource gathering and documentation
- Pranathi Poojary - the Firebase system and LogIn
- Chun Tang - unit testing and documentation
- Everyone - the final report

References

[1] xzippyzachx. (2021) *Unity User Login System Tutorial - Firebase Authentication.* [Video] https://www.youtube.com/watch?v=NsAUEyA2TRo [Accessed 20th December 2022].

[2] xzippyzachx. (2021) *Unity User Data & Scoreboard Tutorial - Firebase Realtime Database.* [Video]. https://www.youtube.com/watch?v=GjyQiq23fBk [Accessed 21st December 2022].

[3] Solo Game Dev. (2022) *Unity Firebase Database Integration - Easy Tutorial*. [Video] https://www.youtube.com/watch?v=59RBOBbeJaA [Accessed 20th December 2022].

[4] OXMOND Tech. (2020) *How To Create a Drop Down Menu - And Get Selected Value [Beginner Tutorial - Unity 2019]*. [Video] https://www.youtube.com/watch?v=5onggHOiZaw [Accessed 21st December 2022].

[5] GDTitans. (2022) *Tilemaps in Unity 2020.* [Video] https://www.youtube.com/watch?v=4eOCTyuOQYo&list=PL0WgRP7BtOezfnz3lPmbs1X6dOu1YsbPh [Accessed December 28th 2022]

[6] Kee Gamedev. (2020) *Unity Victory Effect Tutorial (Confetti)*. [Video] https://www.youtube.com/watch?v=fUly7smuebI [Accessed Decemeber 30th 2022]

[7] The Game Guy. (2021) *How to make a Quiz Game with Multiple Choices in Unity.* [Video] https://www.youtube.com/watch?v=G9QDFB2RQGA&t=877s [Accessed December 30th 2022]

[8] The Game Guy. (2021) *How to make a Quiz Game with Multiple Choices in Unity | Part 2.* [Video] https://www.youtube.com/watch?v=POUemIGCyr0&t=2s [Accessed Decemeber 31st 2022]

[9] Code Cyber. (2022) *How To Make Respawn in Unity 3D*. [Video] https://www.youtube.com/watch?v=gLKPuWzcujE&t=183s [Accessed January 1st 2023]

[10] Penny de Byl - Holistic3D.com (2022) *A Beginner's Guide to Augmented Reality with Unity*. Udemy https://www.udemy.com/course/augmented_reality_with_unity/ [Accessed 28th December 2022]

[11] Thompson, Luke. Interviewed by Das, Shreya et.al. (December 2021)

[12] CPasteGame (2022) *Free 2D Cartoon Parallax Background*. Unity Asset Store https://assetstore.unity.com/packages/2d/environments/free-2d-cartoon-parallax-background-205812 [Accessed 28th December]

[13] Free Icons PNG *White Arrow.* https://www.freeiconspng.com/img/41942 [Accessed 14th January 2023]

[14] Vecteezy - rawain. *Element shape icon* https://www.vecteezy.com/png/9266848-element-shape-icon [Accessed 14th January 2023]

[15] Clipart Library http://clipart-library.com/dinosaur-cliparts.html [Accessed 15th January 2023]