Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Neophyte

Final Report

Ali Soyaslan, Oğuz Liv, Umut Akös, Gülce Karaçal

Supervisor: Halil Altay Güvenir

Jury Members: Uğur Güdükbay and Hamdi Dibeklioğlu

Final Report
December 20, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction 3

2 Sytem Overview 4

3 Algorithmic Design of Neophyte 5

3.1 Matchmaking Algorithm 5

3.2 Scoring Algorithm 5

3.3 Timing Algorithm 6

4 Final Architecture and Design 6

4.1 Overview 6

4.2 Subsystem Decomposition 7

4.3 Subsystem Services 10

4.3.1 Client 10

4.3.1.1 ScreenController Subsystem 11

4.3.1.2 GameGraphics Subsystem 12

4.3.2 Server 13

4.3.2.1 Logic Tier 14

4.3.2.2 Data Tier 14

5 Impact of Engineering Solutions 15

5.1 Global Impact 15

5.2 Economic Impact 16

5.3 Societal Impact 16

6 Engineering Solutions and Contemporary Issues 17

6.1 Technology-Oriented Issues 17

6.2 Children-Oriented Issues 18

7 Tools and Technologies Used 18

8 Resources Used 20

8.1 Library Resources 20

8.2 Web Resources 20

References 21

Appendix A – User Manual 22

1.0 Introduction

In a rapidly digitizing world, having technical skills is very crucial since, nowadays; almost everything requires some form of programming. As technology has been developing, we have become more dependent on it and use various technologies to accomplish specific tasks in our daily lives. Technology is being implemented in almost every section of our lives and business structures. This is the reason why, many countries such as England, Singapore, Estonia and US have started programming education in early ages, because the sooner a person learns how to create programs, the stronger their problem solving abilities get. This education also amplifies their computational and analytical thinking skills. For instance, UK made the most ambitious attempt to get kids coding, with changes to the national curriculum in 2013. ICT – Information and Communications Technology – is out and replaced by a new “computing” curriculum including coding lessons for children as young as five [1]. Such knowledge is important not only to individual students’ future career prospects, but also for their countries’ economic competitiveness and the technology industry’s ability to find qualified workers [2].

However, it appears that Turkey is a little belated to educate children about programming compared to other countries. According to International Computer and Information Literacy Study (ICILS), who conducted among students between 6-15 years from all over the world in 2013, it has been acknowledged that only 1% of students from Turkey have advanced computer knowledge. On the other hand, 35% of students from Korea, 34% of students from Australia and 33% of students from Poland have advanced knowledge about computers and programming [3]. In order to offer an effective and simple solution for this problem, the project Neophyte will be proposed. With Neophyte, we aim to teach children how to code while making them entertained by playing different kinds of games they like. Neophyte creates a platform where children can interact with each other in an exciting way and improve their programming skills.

2.0 System Overview

Project Neophyte is a learning tool for children in elementary school and middle school ages. Strictly speaking, ages ten to fourteen. This tool helps children understand the concept of programming by teaching them the way of computer scientists and basics of simple algorithms. As it was discussed in the introduction, it is important for children to learn algorithm creation in early ages, as it will affect their problem solving and computational thinking skills.

This project aims to raise awareness among children about computer science topics by dragging their attention to these topics with games and fancy graphics. It is important for these games to be easy to understand as it should be challenging enough. Our baseline for these games will be psychological researches and pedagogical reports on child informatics.

Our application will create a real life simulation that will have real life problems that kids may have. For example, kids might lose their keys or toys, or even their tablets or phones. There will be a quest for children to play in single player mode or multiplayer mode. Throughout the quest, there will be mini-assignments for them to solve. In this quest, they may have to send a message to their friend for him to come down to, for instance, basketball court to play with them. If they want to send this message, they have to get the charge cable from their parents first, then they have to find their tablet in their messed up room by cleaning it up. While cleaning their room, they have to grab their belongings that are scattered around in closest element first fashion.

For kids in that age, it might be a little hard to figure out the code at once, twice or even ever. That’s why we will implement an acting game mode for the first levels. They will move their character around the room with arrow keys or W,A,S,D keys and pick up items with space key. This will teach them how to use keyboard as well, since we want to raise kids according to needs of the day. This type of games will get them to recognize the algorithms they will encounter in their education or professional life. Then the games will become more abstract and second levels will contain a box view for coding. It will still hold the concreteness by having blocks all around. Children can play with these boxes in a sandbox fashion. They will create movements and figures by changing block attributes and arrangements. The final levels will be completely abstract and will contain coding fields for children to finally start real coding.

The language of this project is in Turkish and English for now. Normally, making these applications in another language than English is not good for many students, because the keywords of programming languages are all in English. Studying on foreign language basics are sometimes complicating for programmers. On the other hand, we are making this project for children in early ages of education, as our main mission is to endear coding to children, we will have no such apprehension on language. Plus, many children in Turkey still do not get proper English education in Elementary and Middle school.

Finally, this project also helps psychologists and pedologists in many ways, such as, understanding a new way of children and a different way of communication with them. This project may take the initiative for a new research area as children will be motivated to use this system for both entertainment and educational purposes.

3.0 Algorithmic Design of Neophyte

3.1 Matchmaking Algorithm

Neophyte uses a matchmaking algorithm which plays a significant role on the game flow. Because of the fact that scores that each player gains show their knowledge about the subject, it is crucial to match players with similar propagations. To do so, Neophyte implements a matchmaking algorithm whose operational side is responsible to compare the final scores of players.

3.2 Scoring Algorithm

The scoring algorithm, on the other hand, is much more complex. Scores should be given based on topics of the questions. In other words, scoring of a very simple algorithm question and a simple algorithm question is required to differ. For instance, assume there are two similar questions in which player is asked to move on the map 4 times in a row. Yet, on one of the questions, player should use a function 4 times while on the other one, a while loop is used. Considering this, the question with the loop structure will have higher score, because the understanding of a programming concept is a little bit harder than the simple algorithm question.

Furthermore, by enhancing our scoring algorithm, we had the chance to use a simpler matchmaking algorithm. Since, the operational part of this algorithm follows a linear system in which we can make EigenVector calculation. Eigenvectors are the set of basic functions that are the most efficient set to describe data variability and they give use the direction of spread of data [4]. To make the calculations, we assume the questions as contributors to the total score.

$$r = M.r \quad \text{(the main form of the scores)}$$

Only in this case, M is not a coefficient matrix, but an array of coefficients for the questions.

Moreover, before a multiplayer game starts, players are gathered in a lobby at the server and they start the game in a synchronized manner. When a player is doing better in the game, his score will increase faster than other player. In this manner, he will be able to collect much more score.

3.3 Timing Algorithm

Neophyte uses a timing algorithm which can be considered as a helper algorithm for the scoring algorithm. However, when the game finishes, the remaining time to play is multiplied by a factor per second and added to the score. On the other hand, when the player cannot finish the game before the timer runs out, some points are removed from the current score of the player.

4.0 Final Architecture and Design

4.1 Overview

This section explains the proposed software architecture of Neophyte in detail. The partitioning of the system and the classes in each layer is shown with diagrams and explained in the

Subsystem Decomposition section. Afterwards, services of subsystems are explained in detail and class diagrams are provided to give a clear understanding of our architecture.

4.2 Subsystem Decomposition

Neophyte follows a Client-Server architectural style in order to effectively respond/process concurrent user requests. The web application will constitute the client part of the system. The client requests services from the server to function and to respond the needs of the users.

On the server side, the system will be managing database and compiler. This server must be as efficient as possible, because, it processes too much data when command is given by the user. The primary goal is to achieve highest performance while sustaining the lowest response time for each user request. In other words, the client will be focusing on taking the requests of user and on responding them while server will be handling data retrieval tasks and compiler compiles if necessary.

Our system will be divided into 3-Tier system architecture which are: Presentation Tier, Logic Tier and Data Tier. The Presentation Tier, which is the topmost tier, includes the visual components of the system and it can be found mostly on our client-side application. This tier is responsible for managing the interactions with the user. The Logic Tier contains the fundamental operations and all the functionalities of the Neophyte in the Server side of system. Moreover, the Data Tier is responsible for database management and primarily resides on the server side.

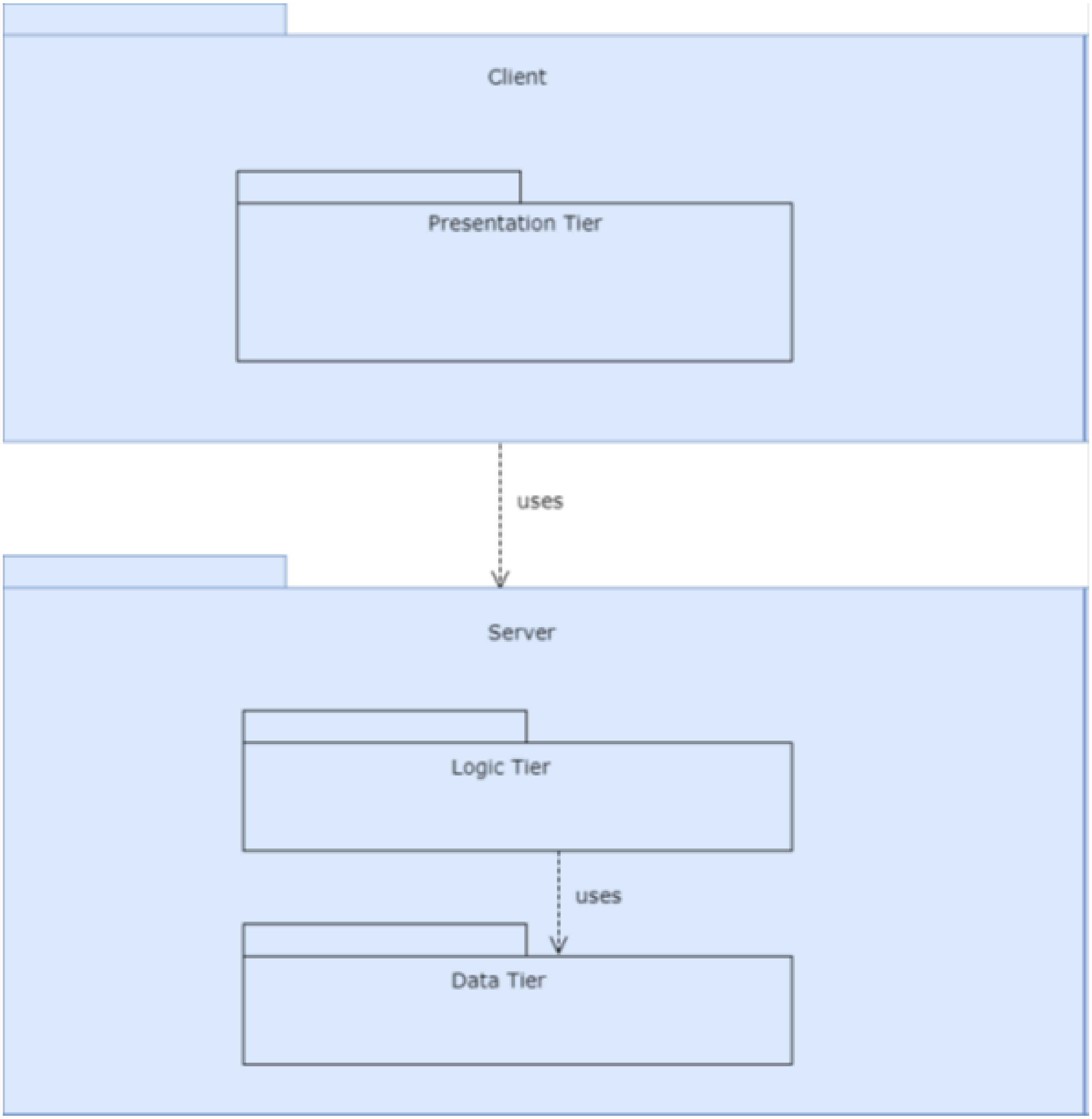


Figure 1: Subsystem Decomposition

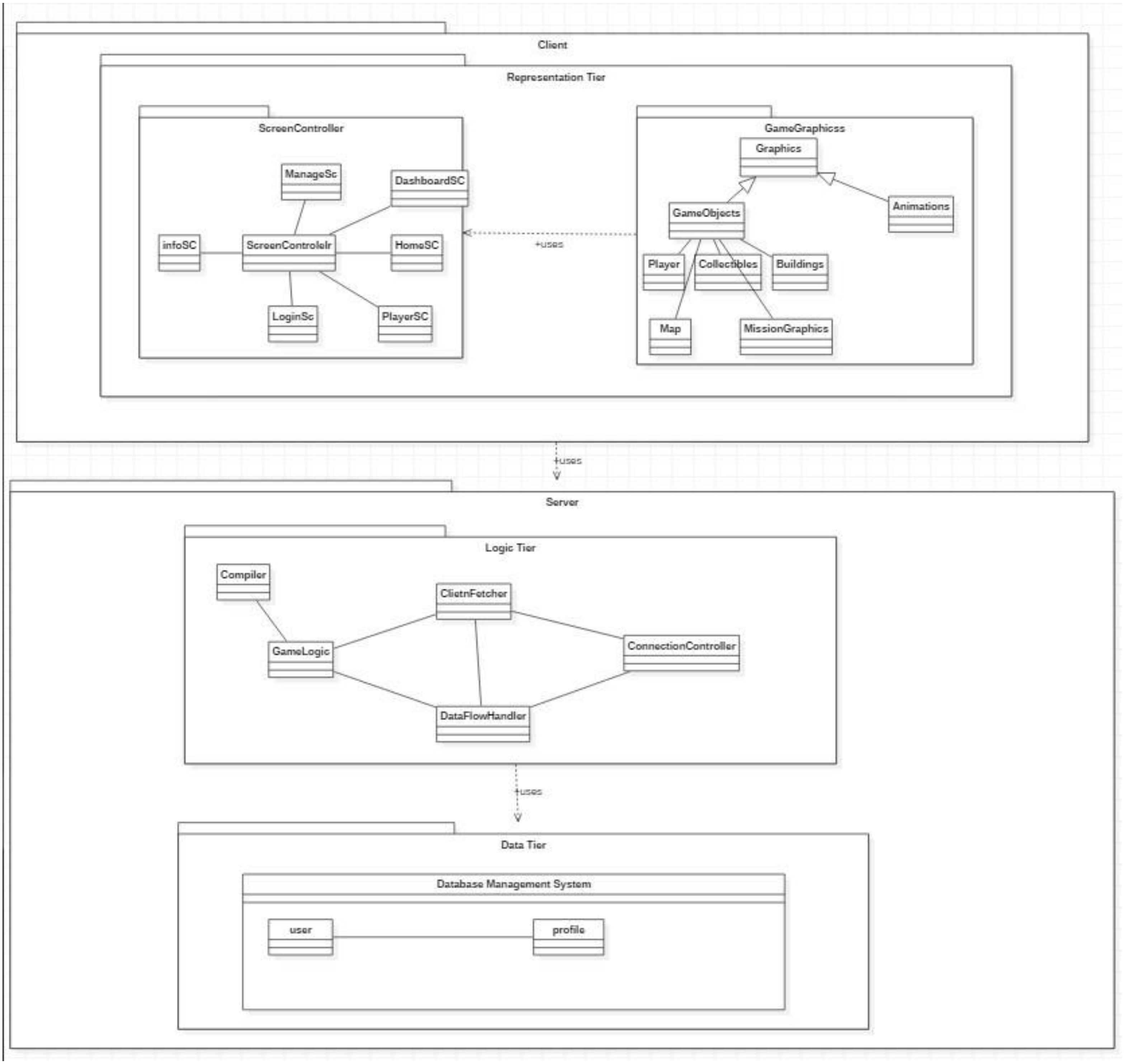


Figure 2: Detailed View of Subsystem Decomposition

4.3 Subsystem Services

4.3.1 Client

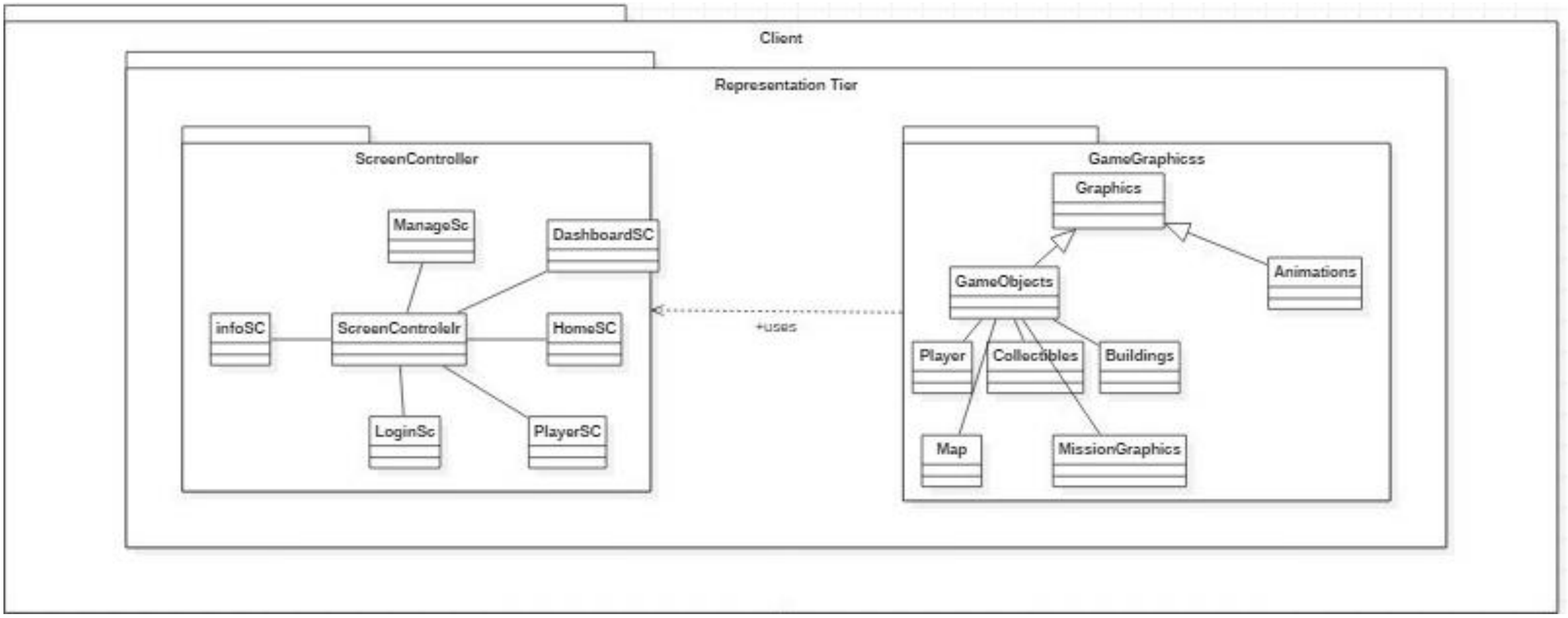


Figure 3: Detailed View of Client Subsystem

Client side is the web application of our system and consists of Presentation Tier. Client side depends on the server side for functioning in terms of collecting data and requesting login authentication.

Client side manages user’s account information and direct messages. In addition to that, client side also presents the data to the user that it collects from the server, gets the user inputs and directs user operations to the server.

Client side involves Presentation Tier that includes ScreenController and GameGraphics packages. Main role of GameGraphics package is to manage all the user interface operations of the user such as presenting the data and getting inputs from the user. ScreenController package is responsible from providing the interaction between client and server by controlling the data transfer. It sends the data to the server that it collects from GameGraphics subsystem and retrieves data from the server for presenting. Other than that, ScreenController manages account of the user, messaging, adding/removing friends etc.

4.3.1.1 ScreenController Subsystem

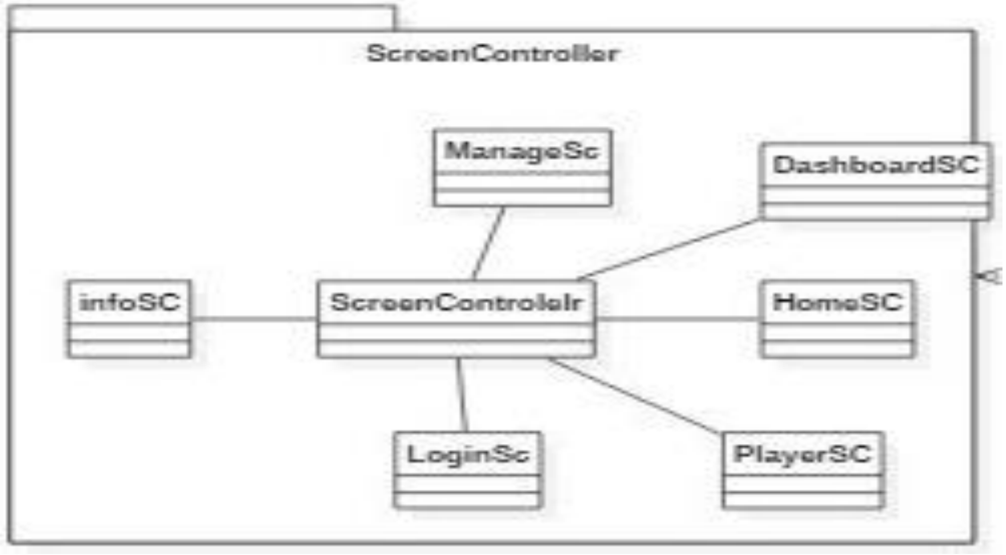


Figure 4: ScreenController Subsystem in Client

- **ScreenController :** It **c**ontrols which screen to be loaded and also handles loading process.
- **LoginSC**: It loads login screen when login button is clicked on home screen.
- **PlayerSC:** It loads the player information such as his/her points, rank and collectibles.
- **HomeSC**: It is the initial opened screen of our program. It offers player to login or get some information about program and its creators.
- **InfoSC:** It loads credits of the game on the screen.
- **DashboardSC**: It loads rankings of the all players in our program. Also, It allows to view player screen when any player clicked on the list.
- **GameSc:** It loads the game when New Game button or Continue button clicked.
- **MessageSC:** It loads message screen when message button clicked on player screen for a specific player.

4.3.1.2 GameGraphics Subsystem

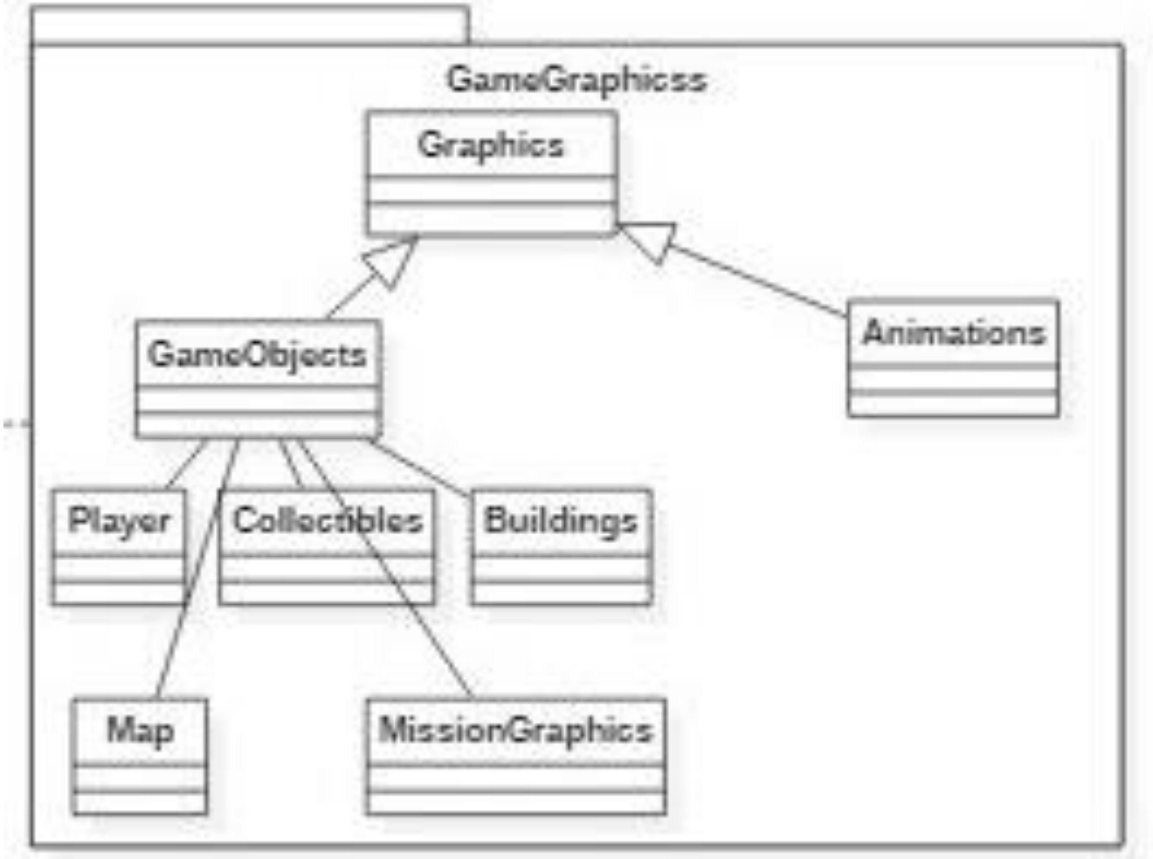


Figure 5: Graphics Subsystem in Client

- **Graphics:** It is the most fundamental attribute for graphics. Other specific graphic objects are inherited from this class and also its functionalities.
- **Player:** It loads player graphics. It depends on the user, so proper graphics are loaded from DB according to player.
- **Collectibles:** It includes the representations of collectibles in the game. Those are cannot be changed so they are also loaded from database according to game flow.
- **Buildings:** Another crucial graphical content of our game is buildings. There are not much building species; however their designs are controlled by this class.
- **MissionGraphics:** According to ongoing mission, graphics are loaded from this class.
- **Map:** It loads all contents of the Map according to selected game mode, game stage and player.

4.3.2 Server

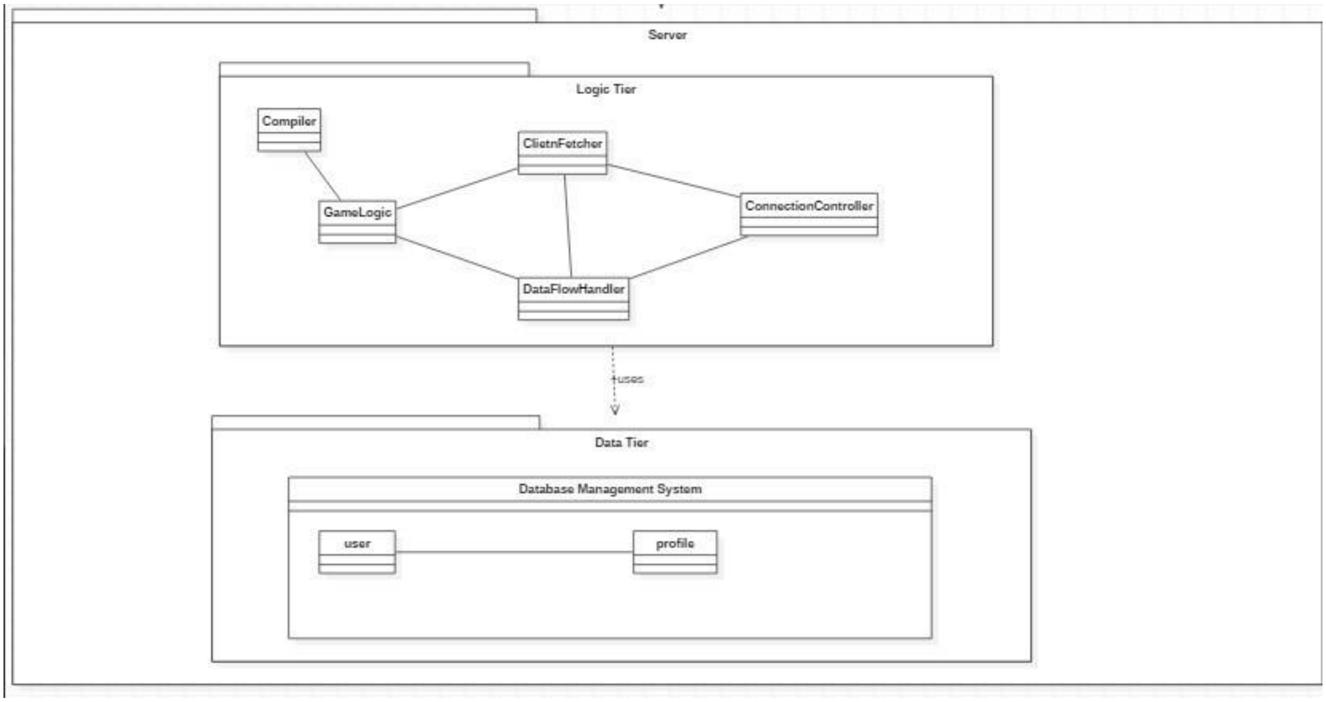


Figure 6: Detailed View of Server Subsystem

Server side is one of the hearts of our project and allows our program to connect to players and its database. Also, in our server architecture we have our compiler in order to compile in game codes. This compiler is going to test in Java.

In our database, all of the entities about players are going to be stored. Also graphics are going to be stored in our database.

Client Side is going to interact with API directly. In order to interact, API is going to fetch instructions from client, and loads proper graphics and content of the program according to request. This process guarantees that there are no direct interactions with data storage. Furthermore, according to game stage, API directs flow to compiler and then continues to control flow according to user commands.

4.3.2.1 Logic Tier

Logic Tier is the layer which communicates with the database and client. It processes the incoming data. This tier contains Client Fetcher and Data Flow Handler, GameLogic and the ConnectionController, which deal with the core business logic.

- **ClientFetcher:** This class communicates with the client. According to client’s commands, this class directs data to Data Flow Handler.
- **Data Flow Handler:** According to command and its perception in the program, this class directs data to proper classes.
- **Game Logic:** This class controls game commands according to user commands. This class directly communicates with client in order to execute client’s commands immediately.
- **Connection Controller:** This class handles connection according to game mode. If single player mode is selected, then the server establishes the connection for the single player. If it is selected as multiplayer, then the connection will be established for multiplayer mode.
- **Compiler:** This class compiles codes and returns the result to the data flow handler, if the code is working or not. This class can be used only in programming assignments of the game, that’s why it is connected to Game Logic class.

4.3.2.2 Data Tier

Data tier is the layer which manages the data of software. It consists of only Database Management System. Database Management system handles relational database operations.

- **User:** It contains user’s data to be represented in the software.
- **Profile**: It includes in-game user data to be represented according to user in the game.

5.0 Impact of Engineering Solutions

In this section; the impact of engineering solutions of Neophyte, in global, economic, and societal context are discussed.

5.1 Global Impact

Based on our market research, there exist applications offered in the market to teach children programming such as Scratch, Blockly and CodinGame. Although these applications have similar functionalities to our system, Neophyte has different and improved features than existing applications.

First of all, we are living in the era of social media and it has a huge impact on people's daily lives. Considering this fact, Neophyte provides children a platform in which they can follow each other, can see other's personal information and send direct messages to each other. Because of having the properties of a social media platform, Neophyte has an important global impact in connecting people.

Moreover, Neophyte also helps psychologists and pedologs in many ways, such as, understanding a new way of children and a different way of communication with them. This project may take the initiative for a new research area as children will be motivated to use this system for both entertainment and educational purposes.

Finally, it allows users to play the game in multiplayer mode and it offers tournaments based on completing time of the given tasks. Hence, Neophyte changes the way children interact with each other with offered competition environment. Also, due to the fact that competition has a positive effect on creativity, children are oriented to think out-of-the-box in order to beat their opponents. Therefore, this project provides an improved and engaging environment where children may not only learn to program, but also have opportunities to be creative using programming. By offering a high quality game experience and encouraging programming as well as creative thinking, Neophyte breathes a new life into the programming application market and therefore it intends to have a significant global impact.

5.2 Economic Impact

Neophyte may have a considerable economic impact because it is a free application. Considering this, for parents who want their children to gain some programming skills, it is a big opportunity. Instead of registering their children into a private paid course, all they need to do is to introduce Neophyte to their children. In this way, they will not waste any money and will make sure that their children will spend a good time while acquiring programming skills.

Furthermore, Neophyte will be useful for public schools. Because of the fact that private schools have capitals to have access to paid applications, they can easily integrate these applications on their curriculums and make students to learn with the help of these applications. However, it is not the case for public schools. For this reason, students in public schools are not able to have education on equal terms. At this point, Neophyte will be a crucial platform in which students in public schools can also learn programming skills as it is free to register. Therefore, public schools can provide it to its students on their curriculum.

5.4 Societal Impact

Differently than the other programming applications, as discussed under its global impact, Neophyte offers social media environment for its users. In this manner, it provides children a platform in which they can easily socialize instead of becoming distant from their friends while spending time on our game. This is the reason why, Neophyte is an environment where children can make new friends by following each other and also, by sending direct messages to each other. Therefore, Neophyte promotes them to be social and thus it affects daily social lives of children. These functionalities allow Neophyte to have a societal impact.

Furthermore, due to the fact that users of our application are children, we have focused on their psychology while deciding on contents of games. According to the American Psychological Association’s (APA) report on violent videogames, there is a consistent relation between violent video game use and heightened aggressive behavior, aggressive cognitions, and aggressive affect and reduced prosocial behavior, empathy and sensitivity to aggression [5]. Considering this fact and also the fact that there exist many games available for children that have violent contents such as Conan Exiles and Sniper Elite, Neophyte follows a different approach to engage children’s attention such as offering competition based on programming assignments, not based on violent actions taken in the game. In this way, children can pay attention on learning new thinking skills instead of focusing on violence. For this reason, Neophyte has a huge societal impact because it detracts aggressive behaviors from children.

Moreover, as mentioned on its global impact, Neophyte encourages children to be creative as it offers competition within its users. Whether internal or external, competition can increase motivation, improve creativity and performance, and provide accountability as well as validation. Stimulation and learning are helpful for brain growth [6]. As a result, Neophyte has an important social influence on children’s ability to think and to create while interacting with each other.

6.0 Engineering Solutions and Contemporary Issues

6.1 Technology-Oriented Issues

Security

As known, security and privacy of personal information has become one of the most controversial issues recently. In Neophyte, user can login to the system with his/her email and password and can provide additional personal information. Hence, the security is crucial for Neophyte. We strengthen the storing way of user’s information to prevent such kind of problem.

Scalability

As the number of user increases, the system is required to handle increased loads of work. This can be larger data-sets and higher request rates etc. Therefore, the server would need to respond to millions of requests every second and manage a huge amount of data. In order to prevent this problem, we tried to implement the system as efficient as possible such that all the requests can be handled in parallel.

Portability

According to our interview with Nilay Pekel Uludağlı, who is a professor specialized in child development psychology, we have learned that children have a tendency to use desktops, mobile phones as well as tablets [7]. Considering this, Neophyte is implemented as a web-based application since browser compatibility is easily achievable compared to operating systems compatibility. Therefore, we have solved this issue by providing children a Web application that can be used in many browsers on many platforms.

Connection

Internet connection is crucial for the system to be able to function. The application relies on Internet connection since it is a web-based application. Hence, in case of a failure of Internet connection, the application may be out of service until a connection is established.

6.2 Children-Oriented Issues

Learning Tendency

Based on our interview with Uludağlı, we have also learned that children have a tendency to learn quickly and permanently with the help of concreteness [7]. Considering this, Neophyte is implemented in such a way that it engages concrete items with programming concepts. It helps children to visualize what happens concretely if they implement different algorithms. Therefore, we have solved this issue by reducing abstraction.

7.0 Tools and Technologies Used

In this section, we are going to give information about the technologies and the tools we used during the development stage.

- **PlayCanvas:** Neophyte uses PlayCanvas which is a visual development platform for interactive web content. This platform was chosen because it provides an easy environment to build 3D web applications.

- **HTML5:** Thanks to PlayCanvas, Neophyte is powered by HTML5 which gives us more choices for a richer website, along with additional features with better markup (HTML), better style (CSS), and better interactivity (JavaScript) [8].

- **NodeJS:** Neophyte uses NodeJs -an open-source (also free to use) web service development framework in order to build our server in an efficient manner. NodeJS runs Javascript on the background.

- **JSON (JavaScript Object Notation):** It is a lightweight data-interchange format that Neophyte uses. When exchanging data between a browser and a server, the data can only be text and JSON is text. Since we use JavaScript, we chose to use JSON as any JavaScript object can be converted into JSON.

- **Glitch:** Glitch is a platform that allows us to build Node.js applications in the cloud therefore Neophyte uses Glitch as a server.

- **GitHub:** This was used as a version control tool and as a Git repository. Issues and our plans were also published to the GitHub's issue tracking system. As a team of four developers, we needed a platform where we can collaborate and work on applications together.

- **Discord:** Discord was used to communicate via voice between developers. Text channels of the discord server created for this project are also used for sharing important documents and links between developers.

- **TeamViewer:** TeamViewer was used to control and view other developers' screens for debugging purposes.

- **Visual Paradigm:** A software design tool. This was used to draw diagrams for the reports.

8.0 Resources Used

8.1 Library Resources

- **Web Graphics Library (WebGL):** Thanks to PlayCanvas, Neophyte uses WebGL that is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins [9].

- **jQuery:** It is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling and animation much simpler with an easy-to-use API that works across a multitude of browsers [10].

8.2 Web Resources

All Internet resources used to find background information including similar design, component information, and basic engineering principles can be found in References. The following Web resources are the most useful ones for us during the design and implementation phases of the project:

- https://developer.playcanvas.com/en/tutorials/
- https://nodejs.org/en/
- https://developers.google.com/blockly/
- https://www.javascript.com/
- https://html5gameengine.com/

References

[1] “Coding at school: a parent's guide to England's new computing curriculum,” https://www.theguardian.com/technology/2014/sep/04/coding-school-computing-children-programming Accessed February 17, 2018.

[2] “Adding Coding to the Curriculum,” https://www.nytimes.com/2014/03/24/ world/europe/ adding-coding-to-the-curriculum.html Accessed February 17, 2018.

[3] “İlkokuldan itibaren Kodlama dersi geliyor!,” http://www.sozcu.com.tr/egitim/ ilkokuldan-itibaren-kodlama-dersi-geliyor.html Accessed February 17, 2018.

[4]” What is the importance of eigenvalues/eigenvectors?,” https://math.stackexchange. com/ questions/23312/what-is-the-importance-of-eigenvalues-eigenvectors

[5] “The Truth About Violent Video Games and Kids, Part 1,” https://www.psychologytoday.com/blog/growing-friendships/201509/the-truth-about-violent-video-games-and-kids-part-1-3 Accessed Feb 18, 2018.

[6] “This is how competition affects your brain, motivation, and productivity,” https://www.fastcompany.com/90240826/this-is-how-competition-affects-your-brain-motivation-and-productivity Accessed Dec 10, 2018.

[7] Interview with Nilay Pekel Uludağlı, Asst. Prof. Dr., Başkent University, March 22, 2018.

[8] “What Is HTML5?,” https://smallbiztrends.com/2014/02/what-is-html5.html Accessed September 10, 2018.

[9] “WebGL,” https://en.wikipedia.org/wiki/WebGL Accessed September 11, 2018.

[10] “What is jQuery?,” https://jquery.com/ Accessed September 10, 2018.

Appendix A – User Manual

Main Screen

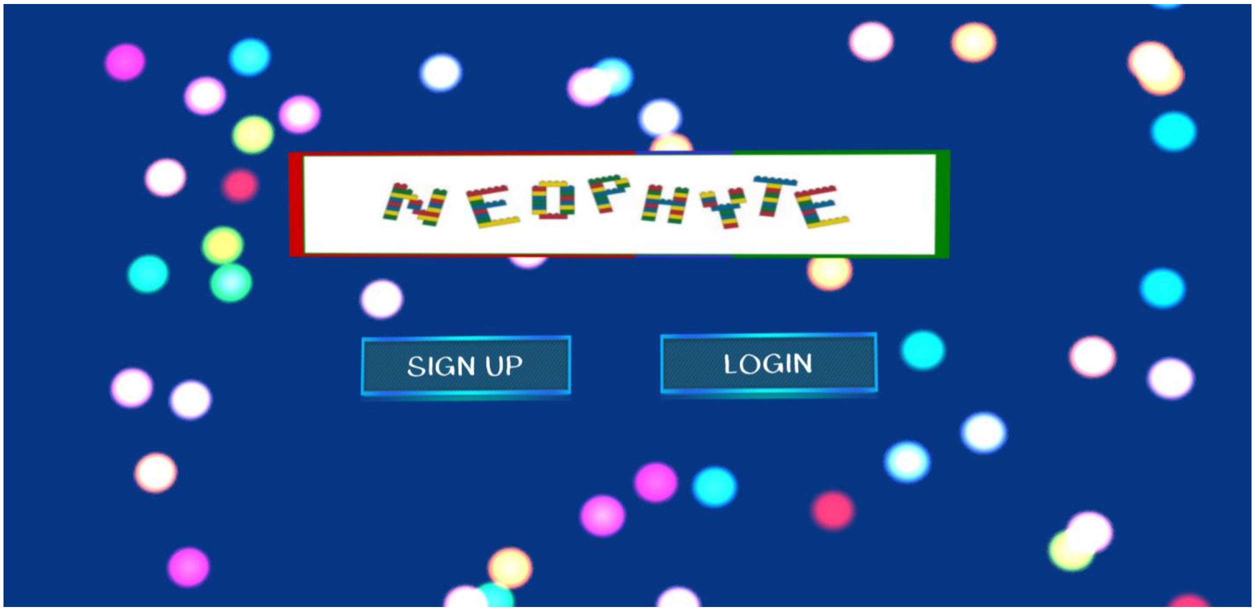


Figure 1: Main Page of Neophyte

This is the main page of Neophyte. You are required to have an account to see the contents. Therefore, if you have a current account, you can login to our system by clicking on the "LOGIN" button. However, if you don't have an account yet, you can easily sign up by clicking on the "SIGN UP" button.

Sign Up Screen

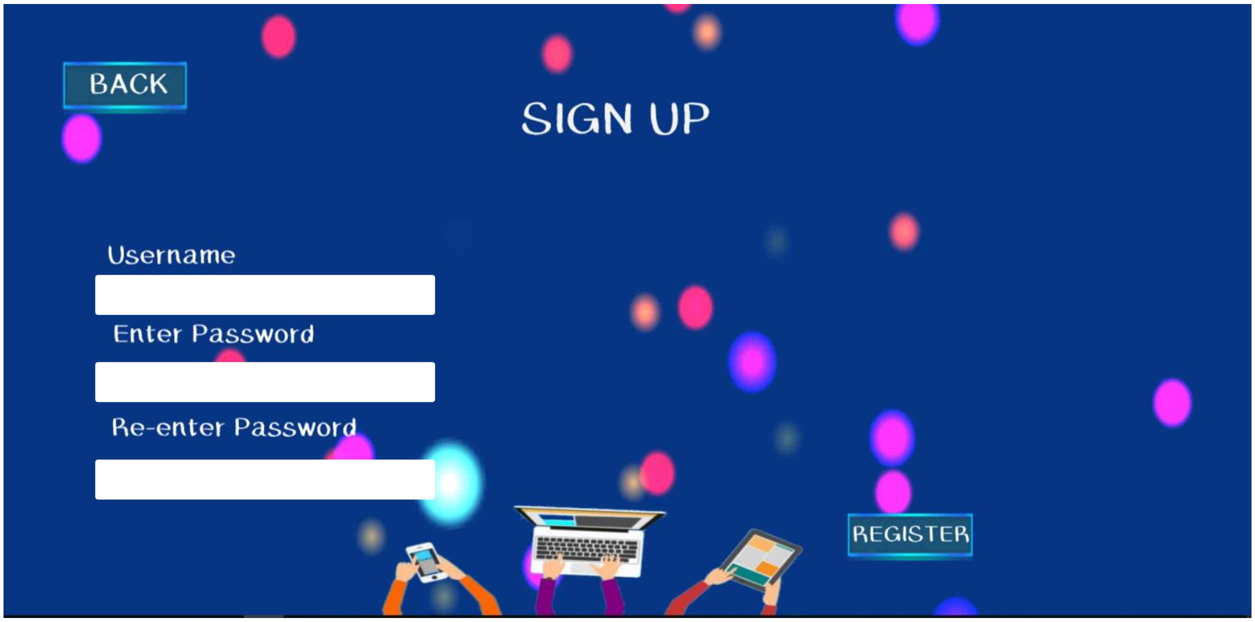


Figure 2: Sign Up Screen

In order to sign up, you are required to create a username which will be the name seen on the system. After providing this information, you should enter a password and confirm the password by re-entering it. In order to create an account, you are supposed to click on the “REGISTER” button at the bottom of the screen. After you logged in, you will be redirected to home page.

Login Screen

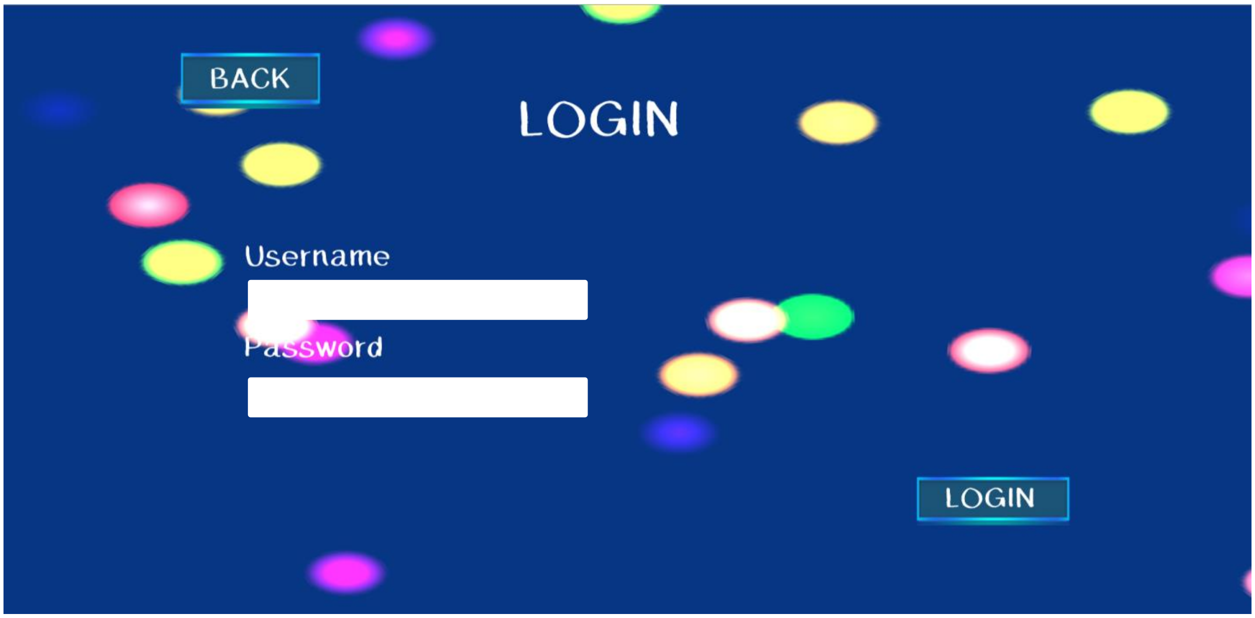


Figure 3: Login Screen

Figure 3 shows the login page of Neophyte. If you are a registered user, you can login to your account by entering your username address and your password. After entering the required information, you should click on the “LOGIN” button in order to proceed.

Home Screen

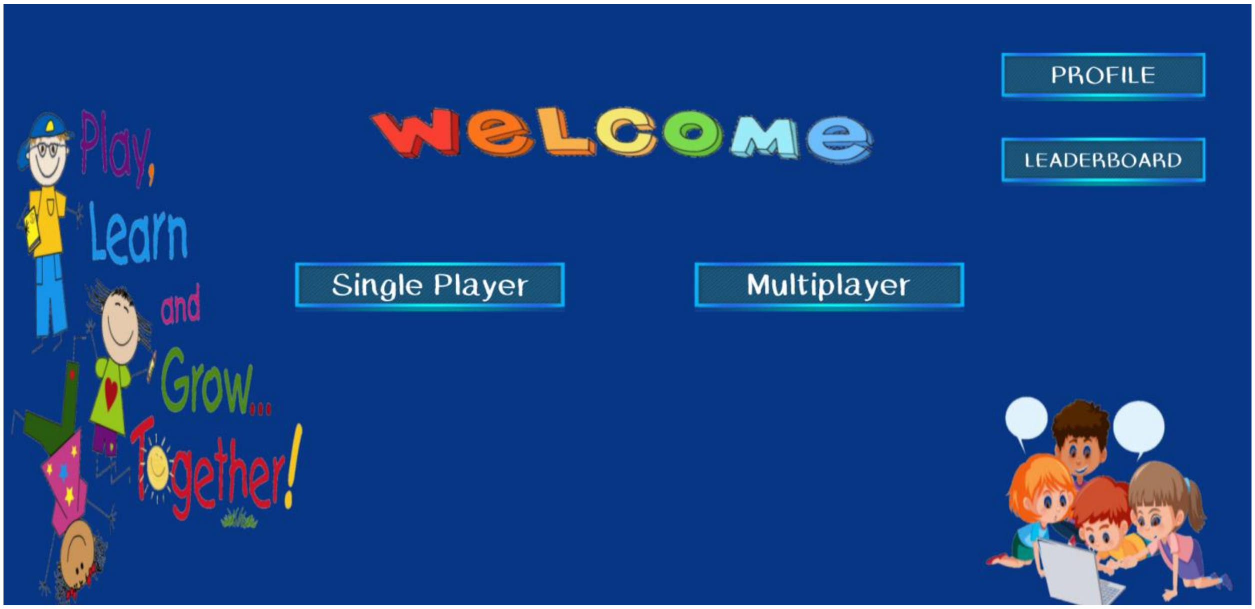
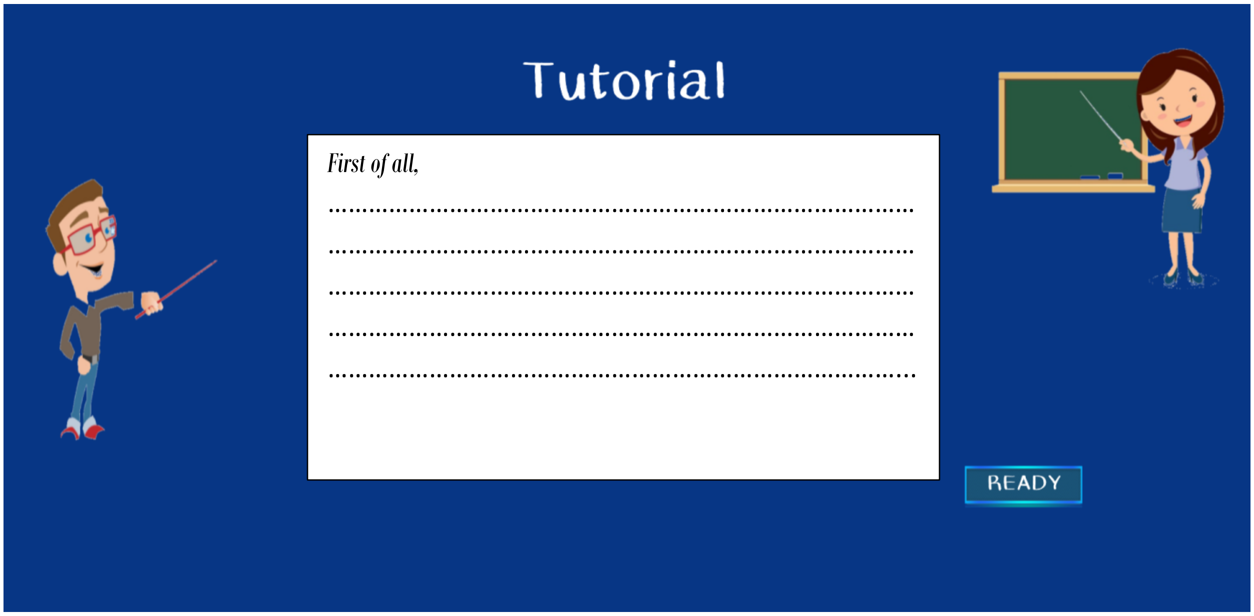


Figure 4: Home Screen

When you login to Neophyte, the home screen on Figure 4 is displayed. You can also reach this page later by clicking the “HOME” button on the top corner of the screen. In this page, you can choose the number of player you prefer to start the game by clicking either “Single Player” or “Multiplayer” buttons. Additionally, you can reach you profile with the “PROFILE” button on the right corner of the screen. Similarly, you can display leaderboard by clicking on the LEADERBOARD” button.

Tutorial Screen



Tutorial

First of all,

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

Figure 5: Tutorial Screen

Before you start playing, a tutorial about the game will be displayed on the screen. Whenever the tutorial page pops up on the screen, you are required to read the tutorial carefully and then click "READY" button in order to proceed to the game screen.

Game Screen

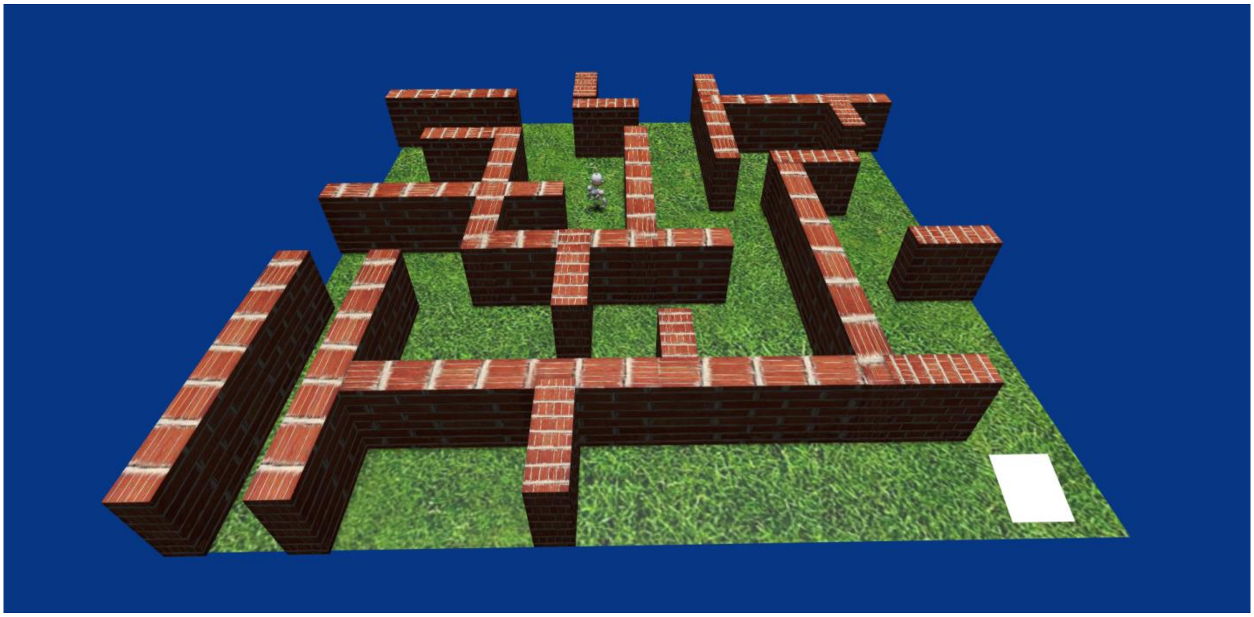
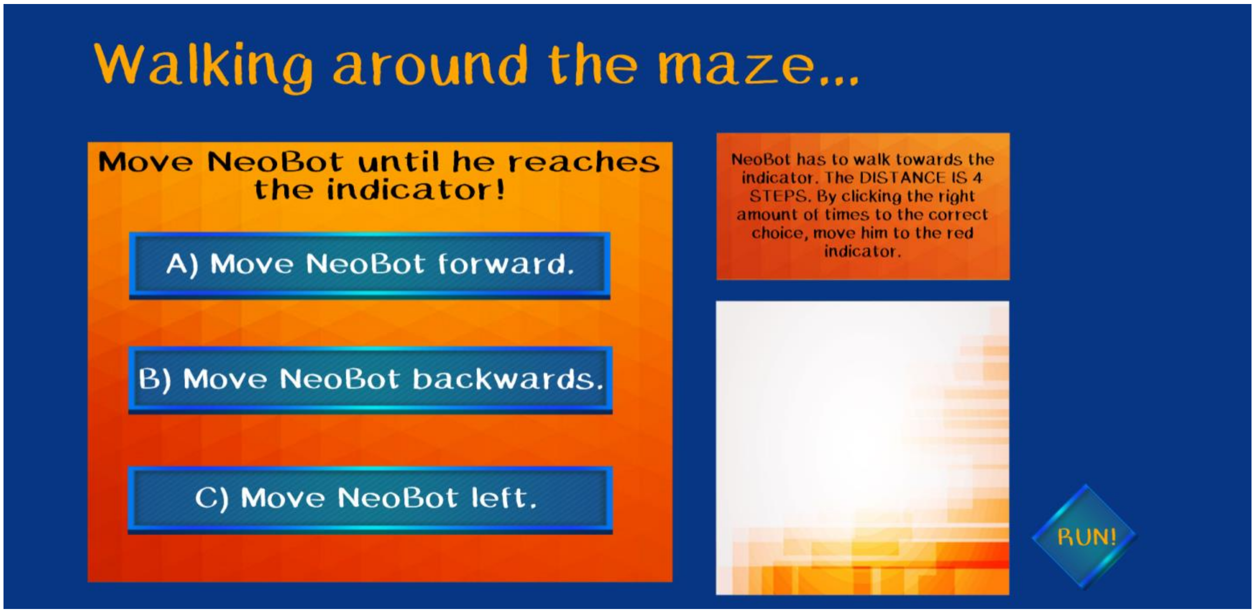


Figure 6: Game Screen

After selecting the number of player, the game screen is displayed where you will see NeoBot which is your character throughout the game. In the game screen, you will see a maze which NeoBot is in. Before you start, a tutorial page is displayed in which you can read necessary information about how to play the game and programming/algorithm concepts that you will acquire within the game. You should read these information very carefully as it affects your performance during the game.

Question Screen



Walking around the maze...

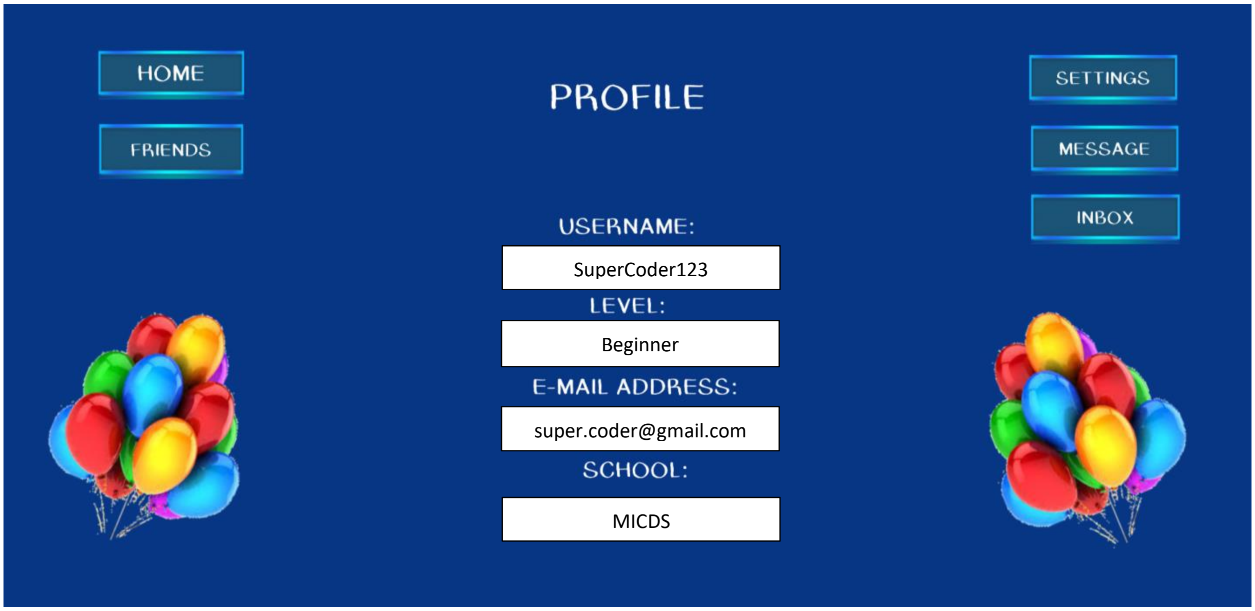
Move NeoBot until he reaches the indicator!

NeoBot has to walk towards the indicator. The DISTANCE IS 4 STEPS. By clicking the right amount of times to the correct choice, move him to the red indicator.

Figure 7: Question Screen

After you read the tutorial, the Question Screen will be displayed. As you see, this screen consists of three main parts: the question on the right, the options on the left and the blank field. First of all, you need to read the question carefully. Then, you are required to select the option that you think is correct. After you click on a choice, you will see that this option will be showed on the blank field. Depending on the question, you may need to select a choice more than once. After you answer the question, you should click on the “RUN!” button to proceed to the game screen.

Profile Screen



PROFILE

USERNAME:
SuperCoder123

LEVEL:
Beginner

E-MAIL ADDRESS:
super.coder@gmail.com

SCHOOL:
MICDS

Figure 8: Profile Screen

In the Profile page, you can see your personal information such as username, level, e-mail and school that you have provided to the system. You can add or delete some of these fields through moving on the Profile Settings Page by clicking on the “SETTINGS” button on the right corner of the screen. Moreover, you can display your friends list by clicking on the “FRIENDS” button. In order to write a direct message to your friends, you should click on the “MESSAGE” button. Similarly, if you want to see your inbox, all you have to do is to click on the “INBOX” button. To go back to the Home page, you should click on “HOME” button on the top left corner of the screen.

Profile Settings Screen

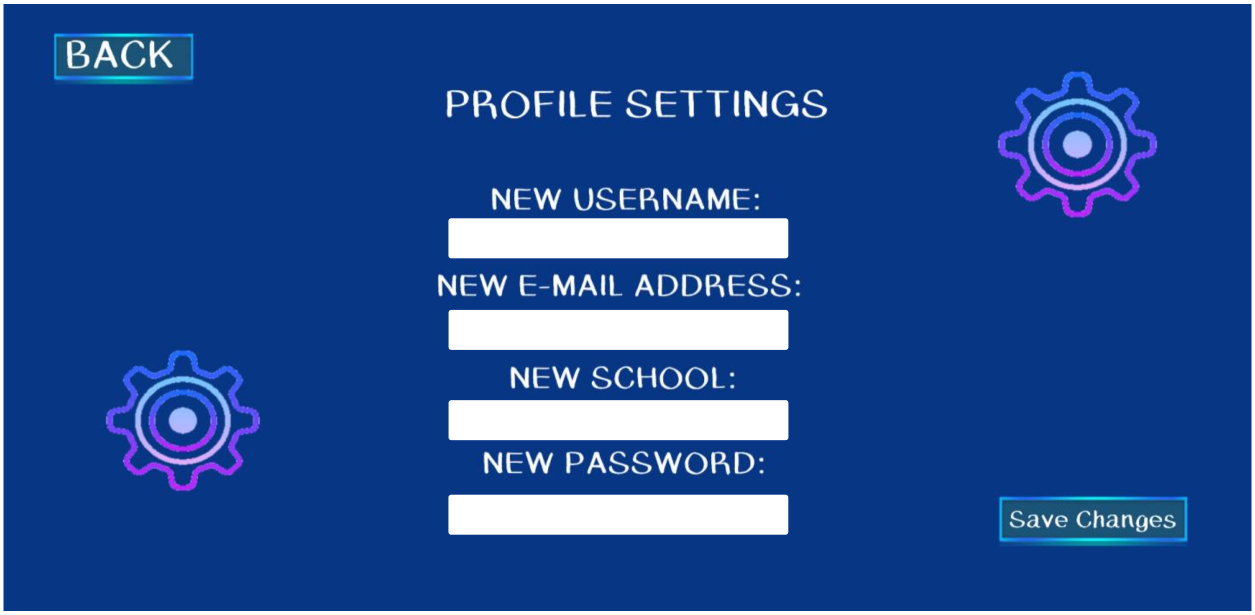


Figure 9: Profile Settings Screen

In this page, you can edit your personal information by entering new information on the text fields. After that, in order save your modification, you are required to click on “Save Changes” button. Then, you can go back to your profile with the “BACK” button.

Message Screen

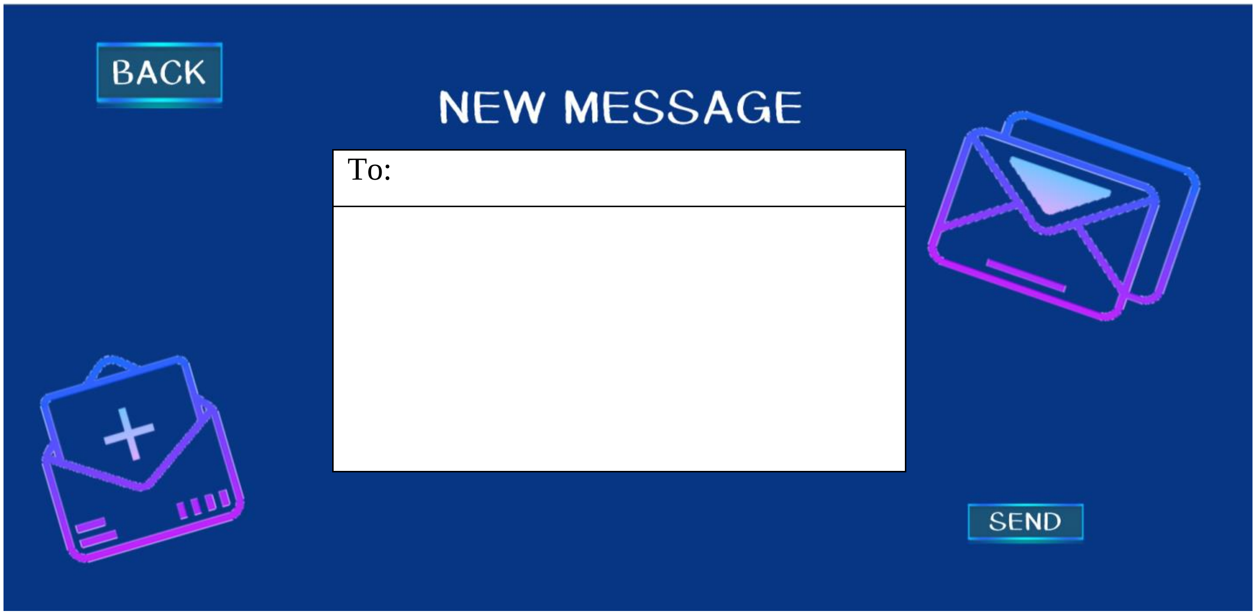
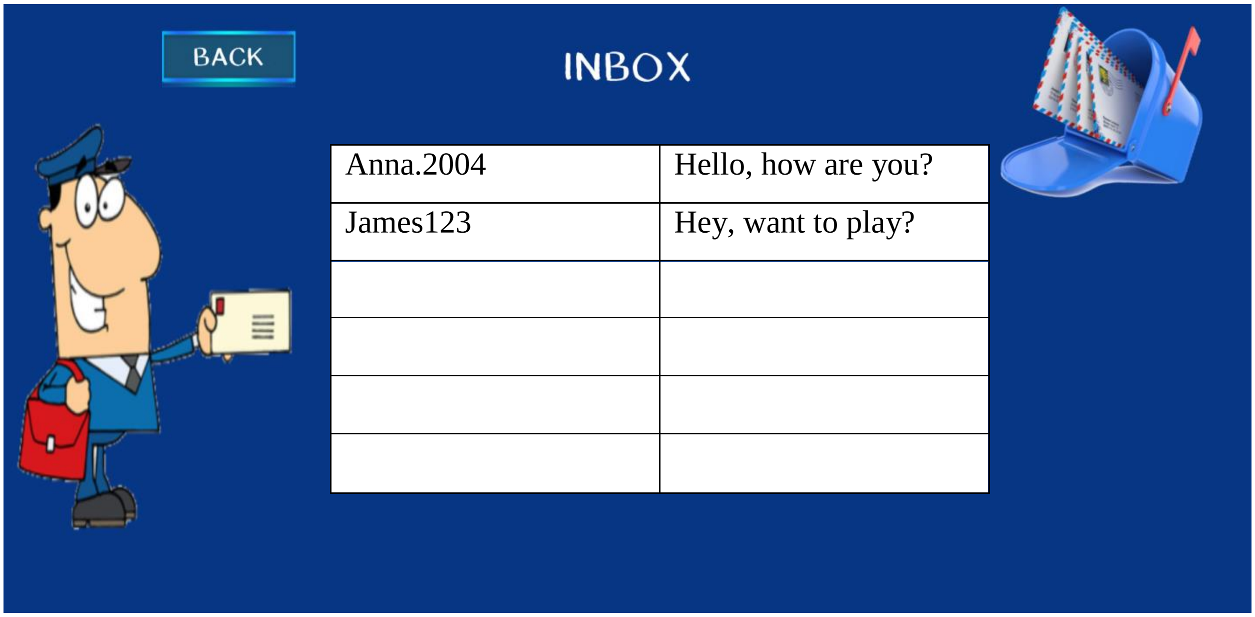


Figure 10: Message Screen

In order to send a message to your friends, you need to write your friends username to “To” field and then write a message on the field provided below. After you complete your message, you should click on the “SEND” button. Then, you can go back to your profile with the “BACK” button.

Inbox Screen

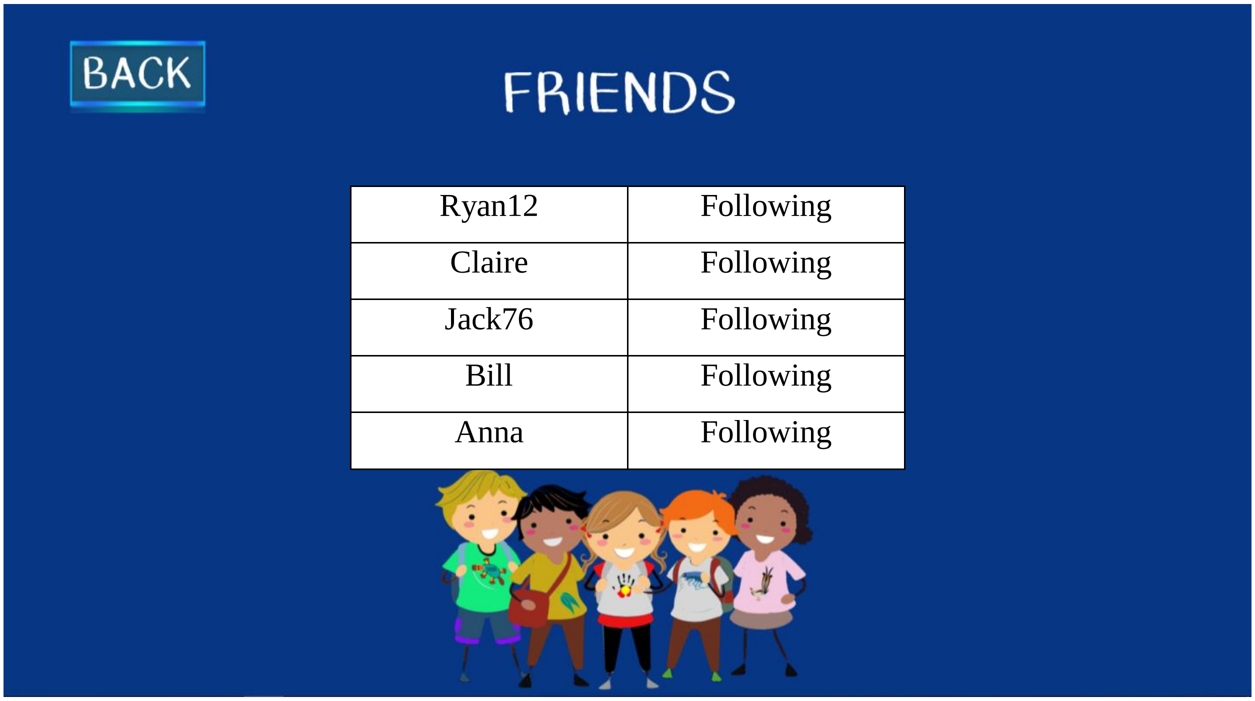


Anna.2004	Hello, how are you?
James123	Hey, want to play?

Figure 11: Inbox Screen

In Inbox page, messages which are sent to you will be displayed in a list with the name of the sender. In order to read a message, you should click on it. Then, you can go back to your profile with the “BACK” button.

Friends Screen

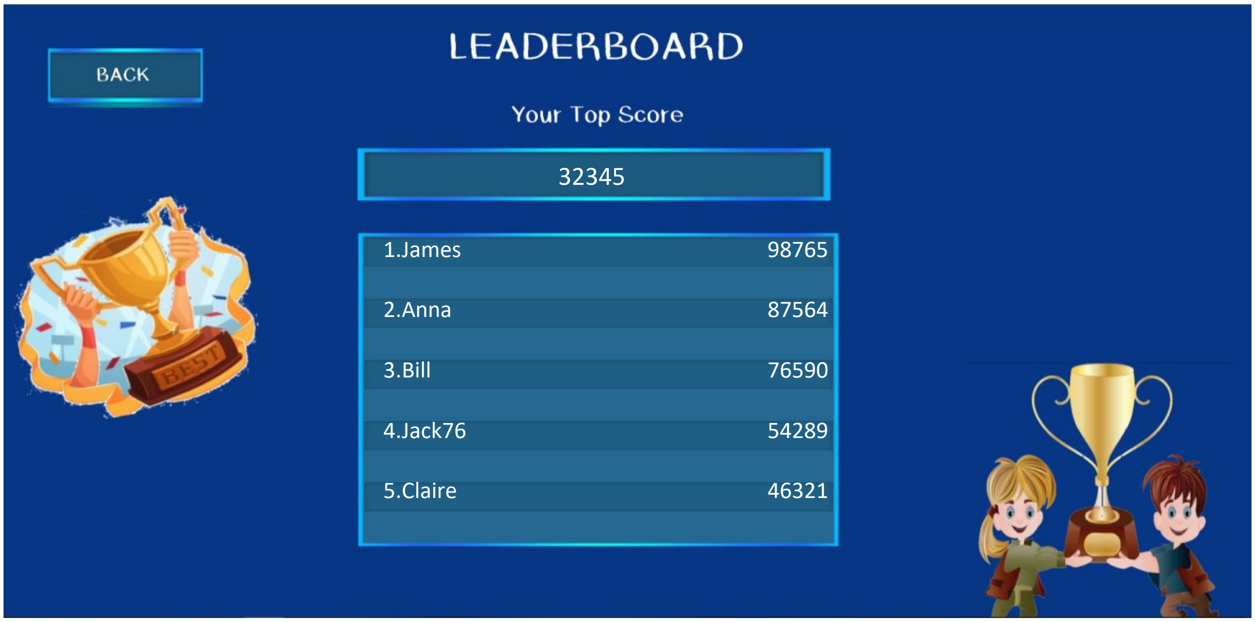


Ryan12	Following
Claire	Following
Jack76	Following
Bill	Following
Anna	Following

Figure 12: Friends Screen

This page displays your friends' name in a list form. You can select a friend to see his/her profile page. Then, you can go back to your profile with the "BACK" button.

Leaderboard Screen



Your Top Score

32345

1.James	98765
2.Anna	87564
3.Bill	76590
4.Jack76	54289
5.Claire	46321

Figure 13: Leaderboard Screen

This screen shows users who have the highest scores. At the top of the page, you can see your highest score as well. Then, you can go back to the Home page with the “BACK” button.