Software Engineering Department
Braude College of Engineering

Capstone Project Phase B – 61999

24-1-D-42

House of Riddles using ML

Supervisor:

PhD. Moshe Sulamy

MosheSu@braude.ac.il

Students:

Seren Hanany- seren.hanany@e.braude.ac.il
Maisalon Safory- maisalon.safory@e.braude.ac.il

GIT:
https://github.com/serenhanany/House-of-Riddles

Abstract

"House of Riddles Using ML" is designed to provide an immersive and intellectually stimulating gaming experience. Players are immersed in a dynamic 3D environment where they navigate through different houses, each containing a series of complex quizzes. As players solve a quiz for each house, they gain scores that contribute to their overall progress. Correct answers increase their score, which in turn raises the level of the game, making future challenges more difficult. The game leverages advanced artificial intelligence and reinforcement machine learning to further enrich the experience. The ML system offers real-time analysis of the player's current state, providing recommendations and hints personalized to their needs while dynamically adjusting the difficulty to keep the game both challenging and fun. This sophisticated approach ensures that players are constantly engaged and motivated. Data from the game is securely managed and stored using an AWS RDS MySQL instance, ensuring that progress and metrics are preserved. The game is developed with Unity, and server-client interactions are managed using a REST API, ensuring seamless connectivity and performance. By combining these elements, "House of Riddles Using ML" aims to provide a unique and engaging alternative to traditional idle games, promote cognitive development and provide a highly interactive gaming experience.

Table of Contents

1.Introduction **3**
2. Gap between Phase A and Phase B **4**
2.1 Model Training Gap 4
2.2 Multiplayer Mode Exclusion 4
3.Development Process **5**
4.Diagrams **6**
4.1 AWS Architecture diagram 6
4.2 Activity Diagram 7
5.Scenes and Flow **9**
5.1 Home Page 9
5.2 Login Page 9
5.3 Register Page 10
5.4 House of Riddles Page 11
5.5 Practice Page 11
5.6 Play Page 12
5.7 Summary Page 15
6.Challenges and Solutions **16**
5.1 Reinforcement Learning Integration 16
6.2 Configuring ML-Agents 16
6.3 Balancing Complexity and Playability 17
6.4 Continuous Adaptation and Learning 17
7.Decisions **18**
8.Tools we used **19**
9.Testing **20**
10.Operation & Maintenance Guide **22**
10.1 Server 22
10.1.1 Accessing the server using SSH 23
10.1.2 Uploading the Server 23
10.1.3 Running the Server 24
10.1.4 Database 24
10.2 Client 24
10.2.1 Server Connection Overview 24
10.2.2 Monitoring and Metrics 24
10.2.3 Server Communication Overview 25
10.2.4 Function Maintenance Tips 25
10.2.5 Troubleshooting Common Issues 26
10.2.6 Maintenance Procedures 26
11.Results & Conclusions **27**
12.Further improvement **28**
13.References **29**

1.Introduction

Our project, “House of quizzes Using ML”, is a single-player quiz game developed in Unity[1], designed to challenge and inspire players through the innovative use of Reinforcement Machine Learning. In this game, the players embark on a journey through a series of houses so that each house contains a quiz that the player must solve to win the house. Progression is closely related to the player's performance, correct answers increase their score, which in turn increases the difficulty level of the following challenges, ensuring a constantly evolving and engaging experience.

“House of Riddles using ML” is a sophisticated machine learning system designed to adapt the game based on the player's skill level and interaction patterns. The game collects real-time data on player performance, such as response time, accuracy, and the number of hints requested. This data is processed by reinforcement learning algorithms that adjust the difficulty of future quizzes, ensuring a balance between challenge and accessibility.

The machine learning system receives input data including player success rates, time spent on each quiz, incorrect answers, and hint usage. Based on this, it outputs personalized hints that guide the player without directly revealing answers, and it dynamically modifies the difficulty of future riddles by altering quiz complexity, response time constraints, or the number of possible answers. This allows the game to provide an experience that evolves with the player’s learning curve, preventing frustration or boredom. The system continuously updates the game's difficulty, making it more or less challenging in response to the player's progress, creating an engaging, personalized experience.

Beyond gameplay, the game is built on a robust technical framework that ensures reliability and security. Player progress and data are securely managed through an AWS RDS MySQL instance, while server-client interactions are streamlined via a REST API. This ensures that the game performs seamlessly, providing players with a smooth and uninterrupted experience.

By continually analyzing the player's quiz-solving behavior, such as the types of errors made, the time taken to solve quizzes, and the frequency of hint usage, “House of Riddles using ML” adjusts the complexity of future quizzes. For instance, if a player struggles with logic-based quizzes, the game provides slightly simpler ones of the same type before gradually increasing the difficulty as the player improves. The quizzes include a mix of logic, memory, and pattern recognition challenges, each dynamically modified based on the player’s current performance to ensure an optimal balance of challenge and learning.

[1] https://learn.unity.com/

2. Gap between Phase A and Phase B

2.1 Model Training Gap

We were unable to train the model as planned in Phase B. For example, the model currently lacks the ability to identify specific strengths in related fields. If a player answers 10 math questions correctly, it cannot infer that the player might also have strengths in physics or other similar areas.

2.2 Multiplayer Mode Exclusion

In Phase A, we planned to implement a multiplayer mode. However, due to time and resource constraints, we were unable to include it in Phase B. Moving forward, refining these aspects, particularly adding the multiplayer mode will enhance the game's ability to engage players, foster deeper cognitive development, and improve problem-solving skills.

3.Development Process

The development process of our game focuses on enhancing players' knowledge through an adaptive, machine learning (ML) powered quiz experience. Games like Lumosity have shown a 97% improvement in users' mental sharpness, while Duolingo equates 34 hours of practice to a semester of university-level language learning[2]. This demonstrates the effectiveness of digital platforms in combining learning with entertainment. Our game uses ML to adapt to players' skill levels, providing personalized and evolving challenges that enhance learning outcomes and engagement. Early studies support using ML-enhanced quiz games to improve learning and skill retention.

The primary beneficiaries include educational institutions, game developers, and players interested in intellectual growth[3]. Academic institutions can use "House of Riddles" to assess and improve cognitive development, while developers explore adaptive gaming.

One bottleneck in educational games is offering real-time feedback and tailored experiences. Machine learning, particularly reinforcement learning (RL), as seen in Unity's ML-Agents Toolkit[4], helps dynamically adjust quiz difficulty based on real-time data[5]. This approach enhances the calibration of challenges according to players' performance.

Non-technology-based quizzes like books and board games remain popular but lack adaptability. Our game demonstrates that digital, ML-powered quizzes provide a more engaging, personalized experience.

Our development process was refined through research in scientific papers, internal discussions, and prototyping sessions using Unity's ML-Agents Toolkit[6]. Feedback from early testers informed the difficulty calibration, ensuring age-appropriate challenges.

By integrating machine learning into the quiz game space, our game dynamically adjusts to players' abilities, guaranteeing a fun and educational experience that promotes cognitive development tailored to a diverse audience.

---

[2] https://www.languagezen.com/pt/about/english/Duolingo_Efficacy_Study.pdf

[3] https://www.mdpi.com/2078-2489/12/4/161

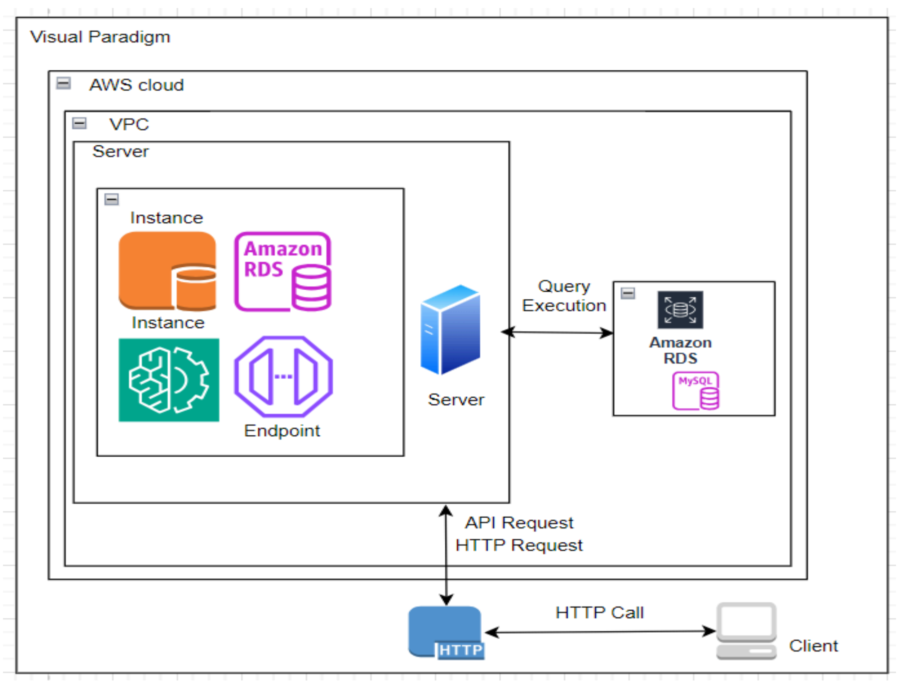
[4] https://github.com/bascoul/Ml-Agents/blob/master/docs/Training-ML-Agents.md

[5] https://www.mdpi.com/2078-2489/12/4/161

[6] https://arxiv.org/abs/1809.02627

4.Diagrams

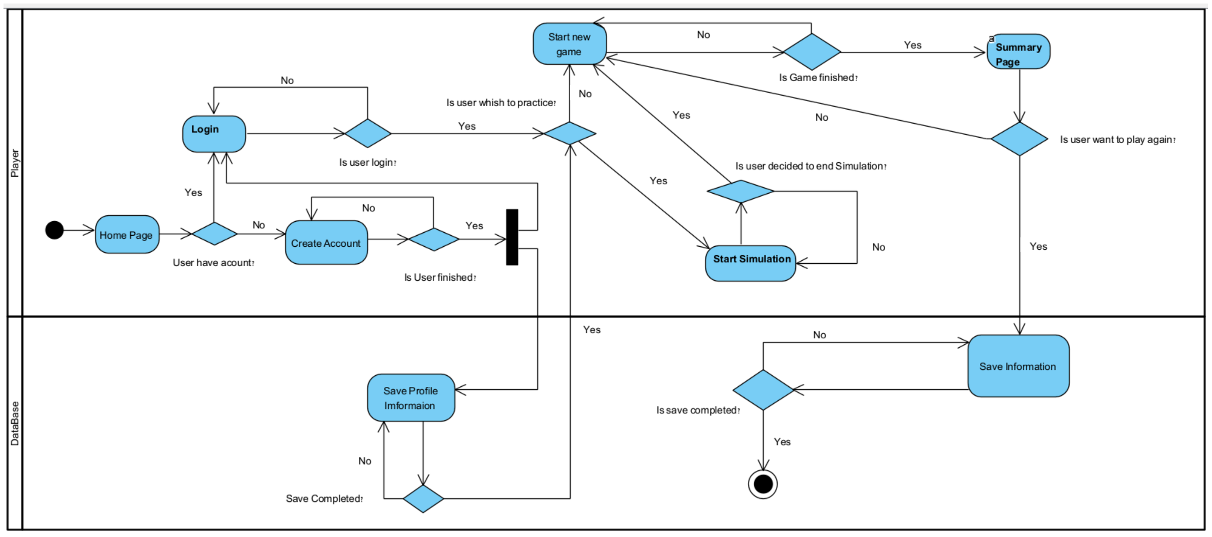
4.1 AWS Architecture diagram



The diagram illustrates the architecture of a cloud-based application hosted on AWS. At the center is an instance, representing a small, cost-effective virtual server designed for handling low to moderate traffic. This server, identified by C#, indicates that it's running a C# application and is connected to persistent block storage for data management.

The EC2 server interacts with a MySQL database hosted on Amazon RDS. This setup allows the server's data to be stored and managed within the RDS service, providing reliable data storage and the ability to scale effortlessly as needed.
On the right, the server connects to a client via a REST API, using HTTPS to ensure secure communication. The client-side is depicted as a web interface, which users interact with, indicating that the application likely includes a web-based front-end.

4.2 Activity Diagram



This activity diagram represents the flow of user actions in a game application, broken down into two main parts: the “Player” actions and “Database” interactions. Here is a detailed explanation of the diagram:

1. Home Page

- The process starts on the ‘Home Page’. Here, the player is prompted to either log in or create a new account.
- Decision:The diagram checks if the user already has an account:
  - Yes:The flow proceeds to the ‘Login’ activity.
  - No:The flow goes to the ‘Create Account’ activity.

2. Login

- In the ‘Login’ activity, the user attempts to log in.
- Decision: Is the user logged in successfully?
  - Yes:The flow moves to the "Start new game" process.
  - No:The process loops back, allowing the user to attempt to log in again.

3. Create Account

- If the user doesn't have an account, they enter the ‘Create Account’ activity.
- Decision: Is the user finished with creating an account?
  - Yes:The user profile information is then saved (handled in the database section).
  - No:The process loops back, allowing the user to complete the account creation.

4. Save Profile Information

- The Save 'Profile Information' step occurs once the user has finished creating an account.
- Decision: Is the save operation successful?
  - Yes:The flow ends here for account creation.
  - No:The process loops back to retry saving the profile information.

5. Start New Game

- Once the user logs in or creates an account, they can start a new game.
- Decision:Does the user wish to practice first?
  - Yes:The flow goes to 'Start Simulation'.
  - No: The game moves forward without simulation.

6. Start Simulation

- The user can go through a practice session or simulation.
- Decision:Has the user decided to end the simulation?
  - Yes:The flow moves to the decision regarding game completion.
  - No:The simulation continues, looping back to the start of the simulation.

7. Is the Game Finished?

- After the simulation or game, there is a decision point to check if the game is finished.
- Decision:
  - Yes: The flow moves to the Summary Page.
  - No: It loops back to allow the user to continue the game.

8. Summary Page

- If the game is finished, the player is directed to a 'Summary Page'.
- Decision: Does the user want to play again?
  - Yes:The flow returns to 'Start new game'.
  - No: The game session concludes.

9. Save Information

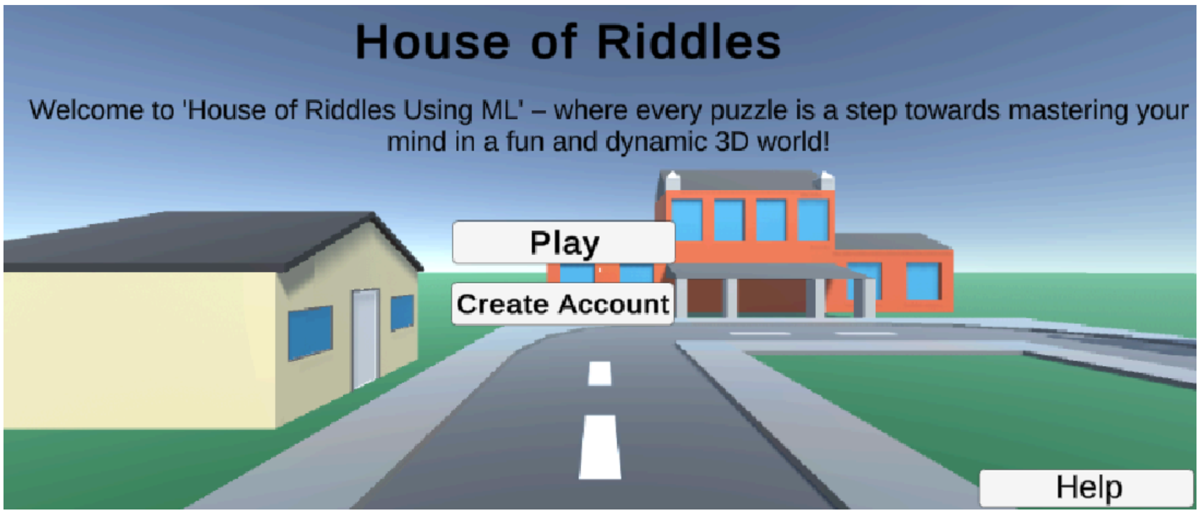
- If the player decides to end the game, any progress or information is saved.
- Decision:Is the save operation successful?
  - Yes:The process ends successfully.
  - No:The system attempts to save again.

5.Scenes and Flow

5.1 Home Page

The **Home Page** serves as the central navigation hub for the player. It includes three buttons:

- **Play**: Takes the player to the **Login** page, where they can enter their credentials and start playing.
- **Create Account**: Directs users to the **Register** page to sign up for the game.
- **Help**: Provides useful instructions or information about the game mechanics or troubleshooting. This page helps orient the player, making it easy to get started or get help as needed.

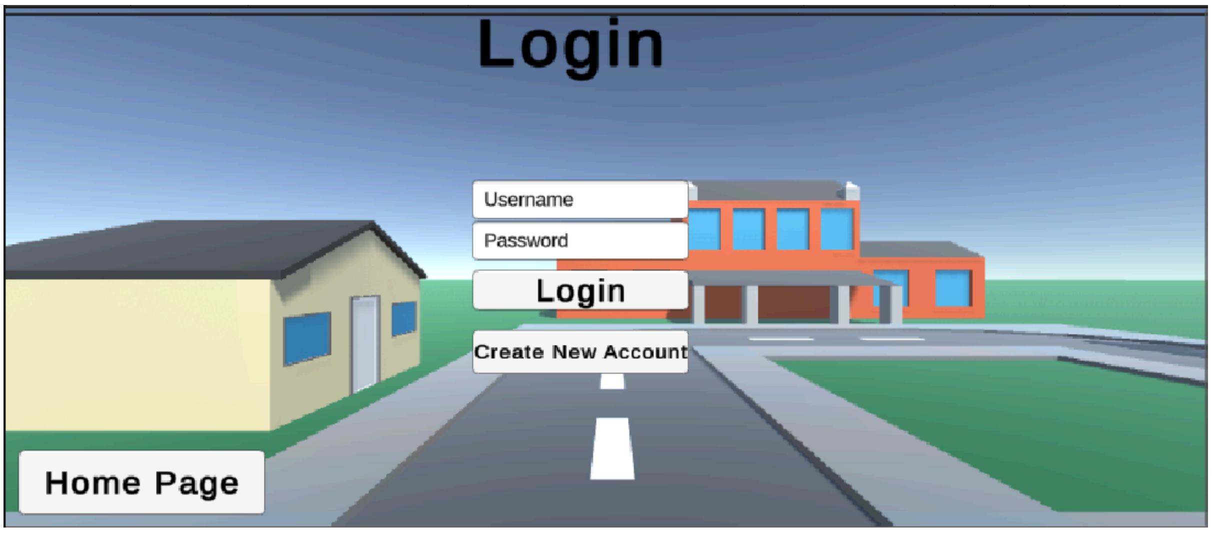

House of Riddles

Welcome to 'House of Riddles Using ML' – where every puzzle is a step towards mastering your mind in a fun and dynamic 3D world!

5.2 Login Page

The **Login Page** allows returning users to access their accounts. It contains:

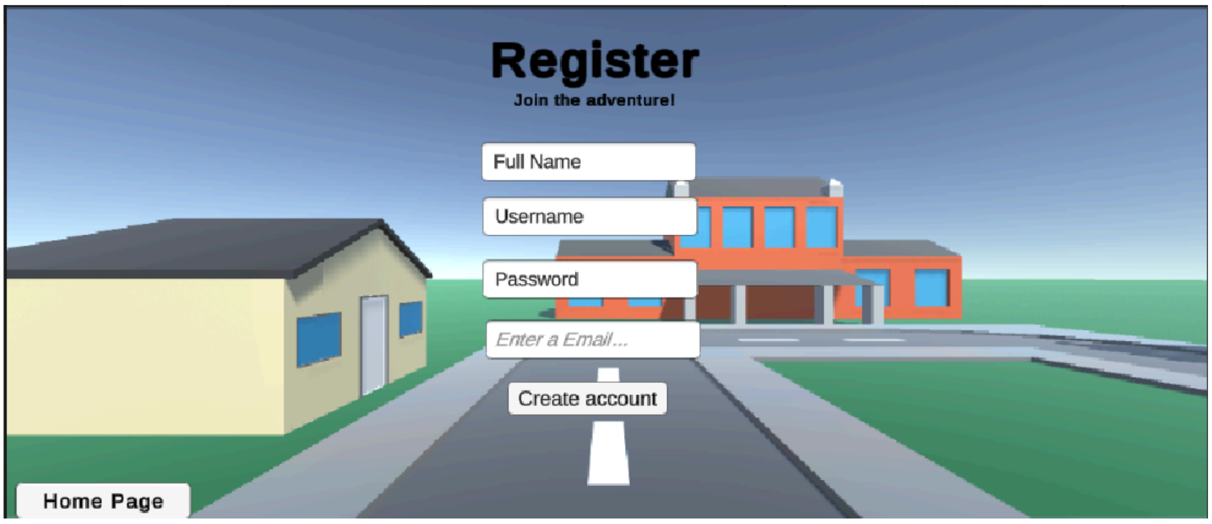
- Two labels for **Username** and **Password** where players input their credentials.
- Three buttons:
  - **Login**: Verifies the entered credentials and grants access to the **House of Riddles** page.
  - **Create New Account**: Directs users who don't have an account to the **Register** page.
  - **Return to Home Page**: A simple navigation button that lets users go back to the **Home Page** if needed.



5.3 Register Page

This page facilitates the registration of new users, collecting essential data through several fields:

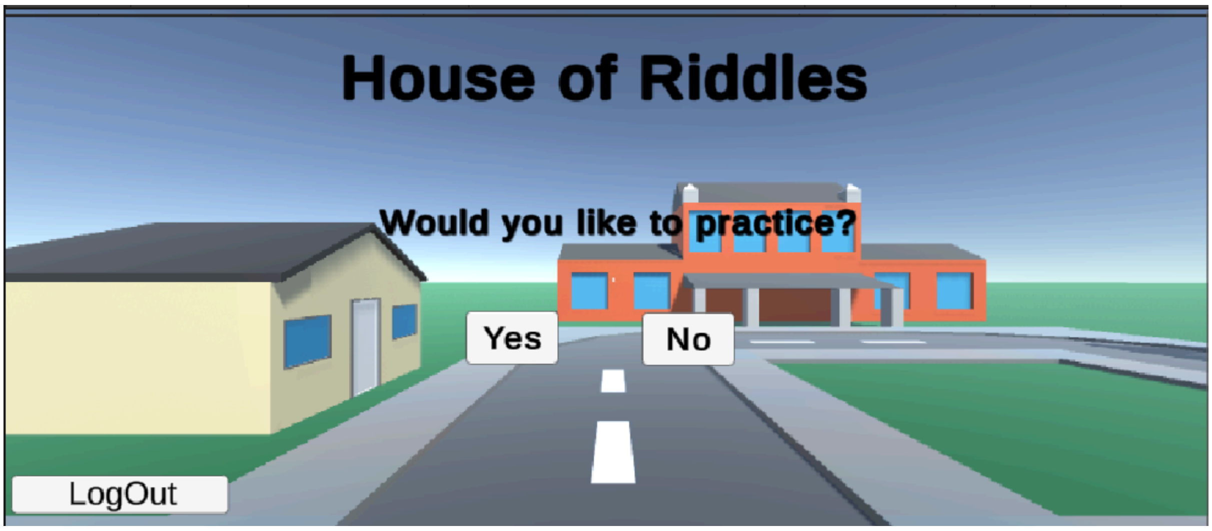
- Fields include: **Full Name**, **Username**, **Password**, and **Email**.
- It also has two buttons:
    - **Create Account**: After filling in the details, clicking this button submits the information to create the account.
    - **Return to Home Page**: Provides the option to go back to the **Home Page** if they decide not to register.
- This page ensures that new players can sign up with the necessary information, establishing a personalized experience.



5.4 House of Riddles Page

After logging in, players arrive at this screen, where they are greeted with a question: **"Would you like to practice?"**

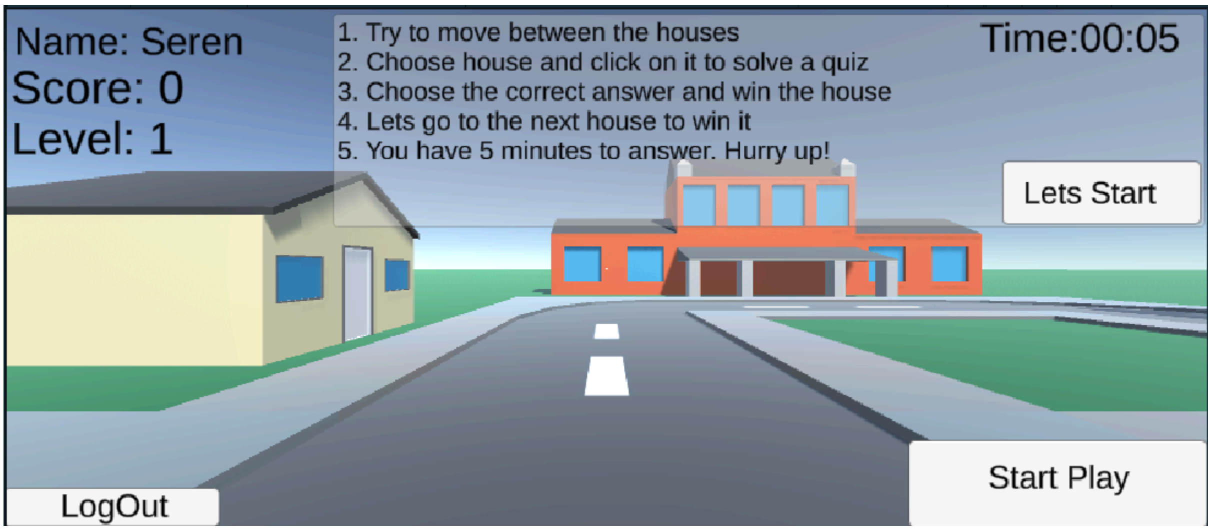
- There are two buttons for answering:
  - **Yes**: Takes the player to the **Practice Page**, where they can get familiar with the game mechanics.
  - **No**: Takes the player directly to the **Play Page** to start playing.
- There is also a button to return to the **Home Page** if the player wishes to go back.



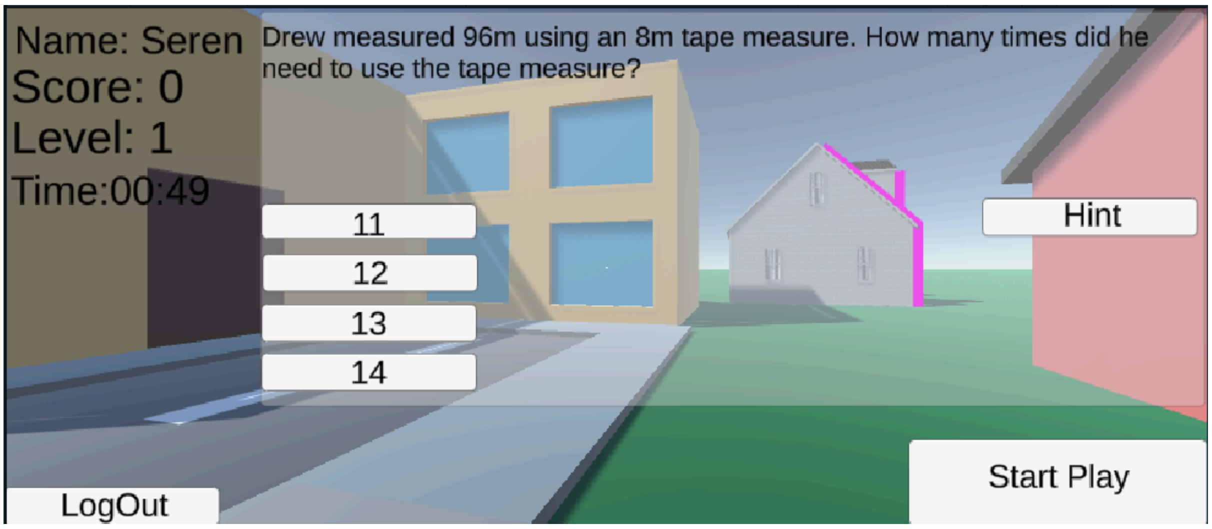
5.5 Practice Page

This page provides a simulation of the actual game, giving players the opportunity to practice without affecting their score or level.

- It includes an explanation of how the game works, describing the key elements like:
  - The **questions** that appear in the houses.
  - The function of the **houses** and how they are related to the game's challenges.
  - The **hints** system, explaining how it assists players when needed.
- The goal of this page is to help players understand the game flow, including answering questions and using hints effectively before diving into the real game.
-
- The Logout Button allows the player to exit their current session and return to the Home Page screen. When clicked.



1. Try to move between the houses
2. Choose house and click on it to solve a quiz
3. Choose the correct answer and win the house
4. Lets go to the next house to win it
5. You have 5 minutes to answer. Hurry up!



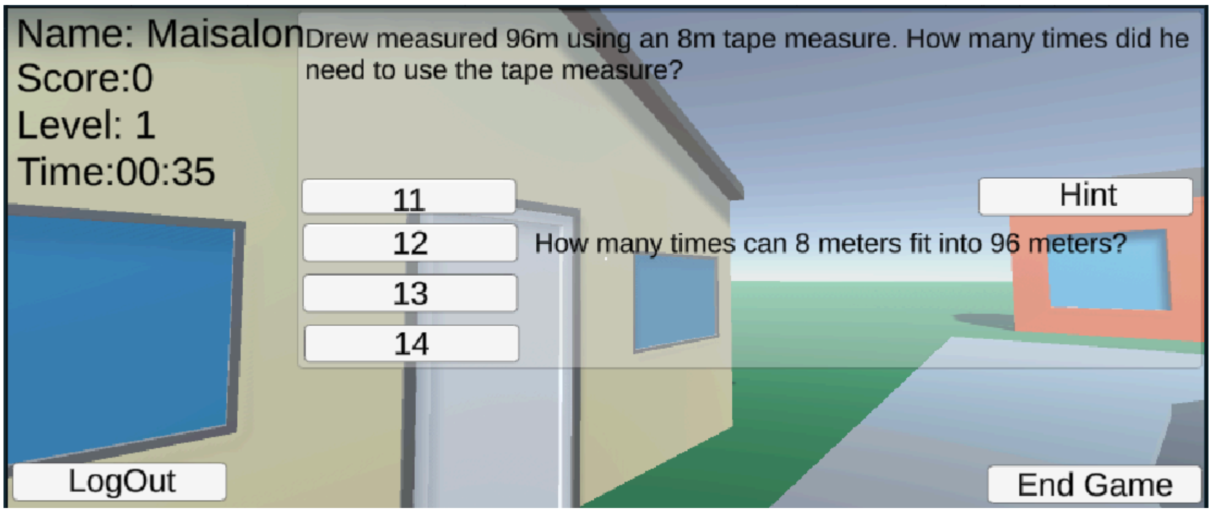
Drew measured 96m using an 8m tape measure. How many times did he need to use the tape measure?

5.6 Play Page

This is the core gameplay screen where the player experiences the actual challenges of the game. It includes several elements:

- **Player details**: Shows the **Name**, **Scale** (progress or achievements), and **Level** of the player.
- **Neighborhood**: Displays the houses that the player navigates through.
- **Questions**: Each house contains a series of questions, and when a player clicks on a house, these questions appear.
- **Answer buttons**: Players choose from several possible answers to the questions presented.
- **Hint button**: This provides personalized recommendations, leveraging the machine learning system to help the player with difficult questions.

- This page fully immerses players in the game experience as they progress through houses, answer questions, and unlock new levels. It keeps track of their advancement in terms of both challenges and scores.
- The Logout Button allows the player to exit their current session and return to the Home Page screen. When clicked.



Name: Maisalon
Score:0
Level: 1
Time:00:35

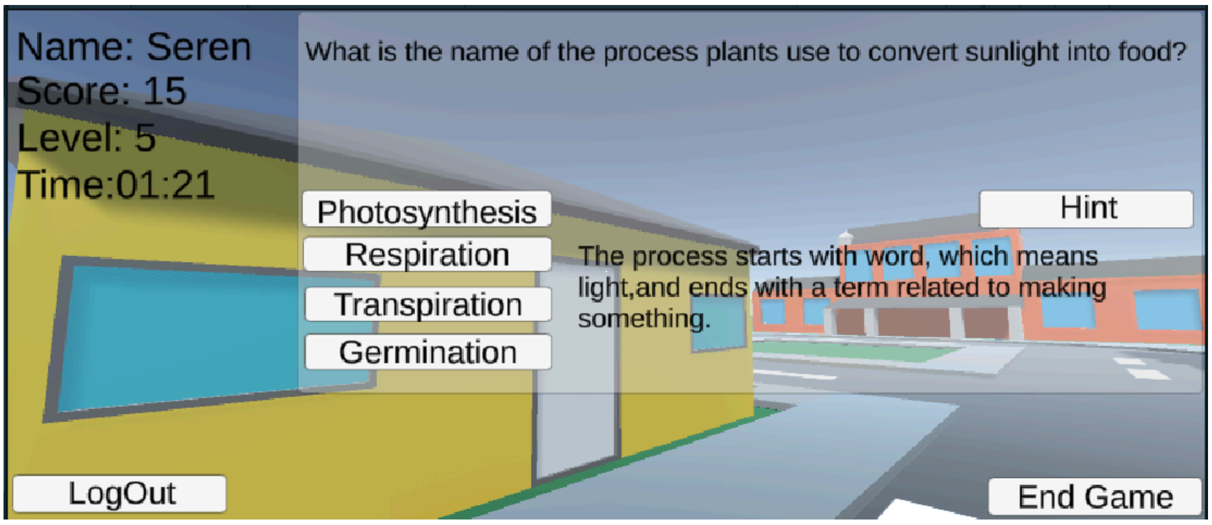
Drew measured 96m using an 8m tape measure. How many times did he need to use the tape measure?

11
12
13
14

How many times can 8 meters fit into 96 meters?

In this example, the player named "Maisalon" is new to the game, indicated by the fact that their level is 1. Because they are at the starting level, the reinforcement learning (RL) model provides an easier hint to assist in solving the question. The hint given, "How many times can 8 meters fit into 96 meters?" simplifies the problem by breaking it down into smaller steps, making it easier for a beginner to understand.

The RL system takes into account the player's level and performance data, such as the time taken to answer and the number of attempts made. By monitoring these factors, the RL model adjusts the hints' difficulty level. If the player struggles or takes a longer time, the model can provide more straightforward hints to help them progress, while gradually increasing the challenge as their level and skills improve. This adaptive approach ensures a personalized learning experience tailored to the player's current abilities.

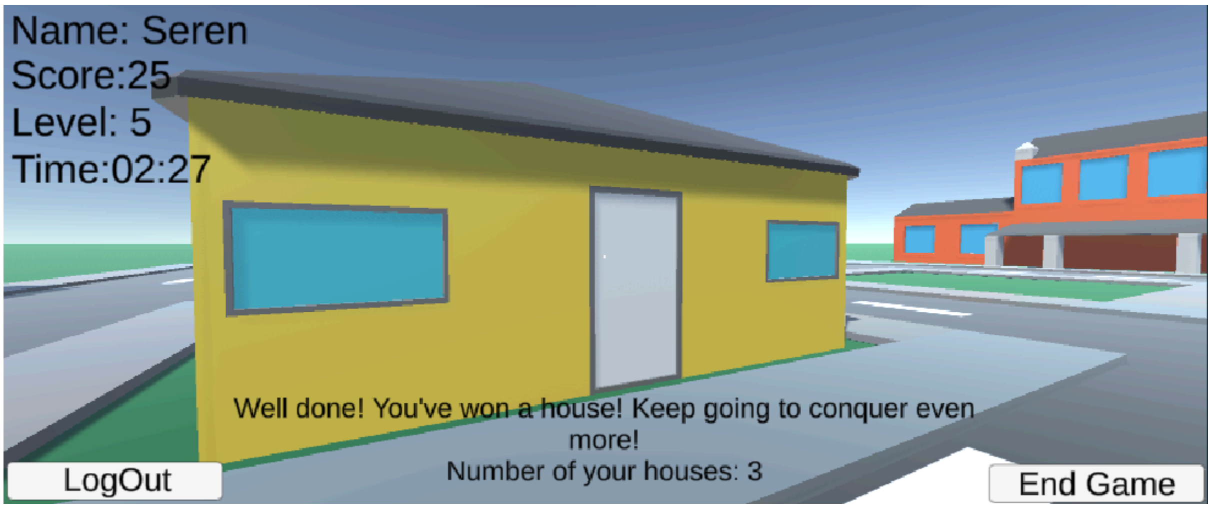


What is the name of the process plants use to convert sunlight into food?

The process starts with word, which means light,and ends with a term related to making something.

In this example, the player named "Seren" is not new to the game, as indicated by their level being 5. Since Seren has progressed to a higher level, the reinforcement learning (RL) model recognizes their experience and provides a more challenging hint: "The process starts with 'photo,' which means light, and ends with a term related to making something." This hint requires the player to connect more advanced concepts, making it suitable for someone at a higher skill level.

The RL system adjusts the difficulty of hints based on the player's level, performance, and history. It also takes into account the time taken (in this case, 1 minute and 41 seconds) and the number of attempts to decide on the next hint's complexity. Additionally, the RL model understands that this hint might be considered hard for Seren, but it could be easy for another player at a different skill level. This dynamic adjustment ensures that hints are tailored not just to the player's current level but also to their individual learning pace and ability.

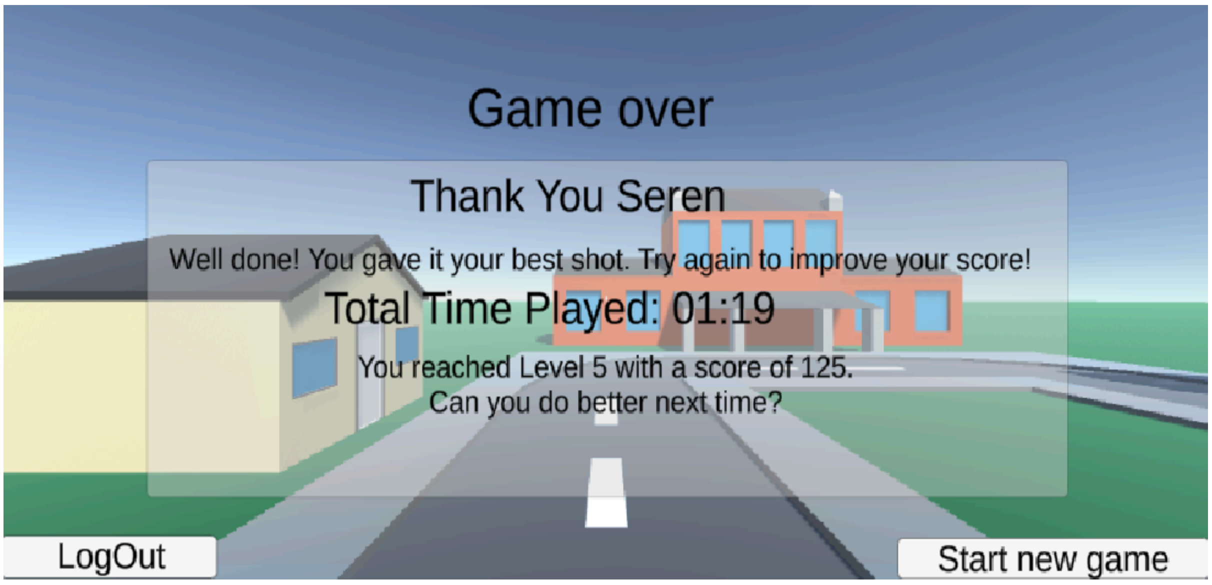


Well done! You've won a house! Keep going to conquer even more!
Number of your houses: 3

In the text displayed at the bottom, it congratulates the player, "Seren," for successfully solving the current challenge: "Well done! You've won a house! Keep going to conquer even more!" This message serves as motivation for the player to continue progressing through the game.

The subsequent line, "Number of your houses: 3," indicates that Seren has already solved three houses in total. Each house represents a set of challenges or quizzes that the player has successfully completed. This running count serves as a way for the player to track their progress and achievements in the game, providing a sense of accomplishment and encouraging them to keep pushing forward to conquer additional houses.

5.7 Summary Page


Game over

Thank You Seren

Well done! You gave it your best shot. Try again to improve your score!

Total Time Played: 01:19

You reached Level 5 with a score of 125.
Can you do better next time?

Summary Page provides an overview of the player's game session, showing:

1. **Time Played**
2. **Final Score**
3. **Level Reached**

It also includes two action buttons:

- **Start New Game:** Begin a new session.
- **Logout:** Exit the game and return to the login screen.

This page lets players review their performance and decide their next step.

6.Challenges and Solutions

Challenges and Solutions

5.1 Reinforcement Learning Integration

.

Challenge: Lack of expertise in RL for Unity and neural network interaction.
Solution: We deepened our understanding of reinforcement learning (RL) by studying key concepts like reward functions, state-action spaces, and policy optimization techniques[7]. We ensured the proper integration of Unity's ML-Agents toolkit—a specialized plugin that allows for the implementation of RL in Unity's 3D environments[8]. The toolkit enabled us to train agents to make decisions in real-time by simulating environments where they learned through trial and error[9].

6.2 Configuring ML-Agents

Challenge: Difficulty in configuring ML-Agents, including reward functions and hyperparameters.
Solution: To resolve this, we mastered key RL concepts and applied them specifically within Unity's ML-Agents framework. ML-Agents is a tool that enables game developers to train machine learning models to act as agents in Unity environments, using techniques like reinforcement learning and imitation learning. We configured reward functions that incentivized correct actions, such as solving quizzes efficiently, while penalizing incorrect answers. We also adjusted hyperparameters, such as learning rates, batch sizes, and exploration parameters, to optimize the learning speed and performance of our RL agents. This process involved extensive research and testing to

[7]https://books.google.co.il/books?hl=ar&lr=&id=OMNiDwAAQBAJ&oi=fnd&pg=PP1&dq=Reinforcement+Learning+Algorithms+with+unity+game&ots=CkBONy-A92&sig=OHdCTANtyNth_N5-Auc_sduxZq8&redir_esc=y#v=onepage&q=Reinforcement%20Learning%20Algorithms%20with%20unity%20game&f=false

[8] https://isprs-annals.copernicus.org/articles/IV-4-W4/161/2017/isprs-annals-IV-4-W4-161-2017.pdf

[9] https://www.youtube.com/watch?v=zPFU30tbyKs

ensure the agents behaved appropriately in various quiz-solving scenarios, reflecting realistic player performance[10].

6.3 Balancing Complexity and Playability

Challenge: Integrating complex ML models while keeping the game accessible and enjoyable.
Solution: This challenge required a delicate balance between advanced machine learning functionality and user-friendly game design. To ensure playability, we engaged in an iterative testing process with a diverse group of test players. We monitored how different players interacted with the game's quizzes and adjusted the learning curve accordingly. To avoid overwhelming players, we designed the user interface (UI) to provide clear feedback on their progress and integrated ML models that gradually increased the difficulty of quizzes based on performance. For example, as players successfully answered more quizzes, the ML model adapted by offering harder, more complex quizzes. However, if a player struggled, the model would lower the difficulty or offer tailored hints to assist. The result was a dynamic difficulty adjustment system that kept the game engaging for both new and experienced players, ensuring a satisfying and accessible gaming experience.

6.4 Continuous Adaptation and Learning

Challenge: Managing constant updates to ML models without disrupting game stability.
Solution: We established a robust version control system to manage changes to ML models and game code. Regular testing was conducted to ensure that updates to the ML system did not introduce instability into the gameplay. This allowed us to continue improving the ML model's adaptability without compromising the overall game experience.

[10] https://www.theseus.fi/bitstream/handle/10024/496604/Breugelmans_Arno.pdf?sequence=2&isAllowed=y

7.Decisions

During the development of *House of Riddles using ML,* we encountered several challenges that led to critical decisions shaping the final game. Initially, our machine learning model struggled to adapt quiz difficulty accurately, leading to a mismatch between the player's abilities and the game's complexity. To address this, we implemented a calibration phase at the start of the game, where players begin with simpler quizzes. This phase allows the model to gather essential data, such as the player’s response time, accuracy, and hint usage, refining its predictions to ensure that subsequent quizzes are appropriately challenging and engaging.

Another challenge involved our dynamic hint system, which was initially programmed to provide hints from a predefined list for each question. However, the system sometimes selected hints that were too specific or even gave away the correct answer due to a flaw in how the hints were chosen. To improve this, we restructured the system to ensure that hints are more tailored to the player's strategy and progress without directly revealing the answer. Now, instead of giving away the correct option, the hints guide players by pointing out strategies or key information they may have missed, such as focusing on particular facts or narrowing down options.

Finally, we faced issues with data synchronization between the game client and the AWS RDS MySQL instance, leading to noticeable latency during gameplay. This latency was primarily caused by frequent database queries for relatively simple operations, such as retrieving player progress and quiz state updates. To resolve this, we restructured our REST API interactions and introduced a caching mechanism. We decided to cache frequently accessed data, such as player scores, quiz states, and session information, on the client side and in a distributed in-memory cache on the server. By reducing the number of direct database calls, we significantly improved performance, ensuring smooth, responsive gameplay while still securely managing player data.

8.Tools we used

In developing the game, we chose Unity as our primary platform. AWS was selected for server hosting and management, with the server built using C#. MySQL Workbench, integrated with Amazon RDS, was used for database management. C# was also utilized for in-game scripting due to its seamless compatibility with Unity, all within the Visual Studio 2022 IDE. For version control, we relied on Git, integrated with GitHub, and used GitHub Desktop to push all updates. Additionally, the REST API was employed to facilitate communication between the client and server.

9.Testing

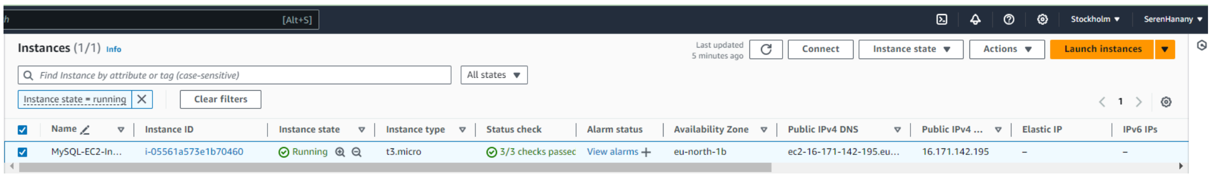
Test	Component	Test case description	Expected Result
1	Login	Correct username and password, press login	Go to the next screen
2	Login	Incorrect username and/or password, press login	Display error message
3	Login	Empty username or password field, press login	Display error message
4	Register	Enter valid details (Full Name, Username, Password, etc.) and press 'Create Account'	Account created
5	Register	Enter existing username and press 'Create Account'	Display error message
6	Register	Leave required fields empty (Username, Password, etc.), press 'Create Account'	Display error message
7	Home Page	Press 'Play' without logging in	Display error message
8	Home Page	Press 'Create Account' from the home screen	Navigate to the register page
9	home Page	Press 'Help' from the home screen	Display the help panel with the correct information
10	Play Scene	Player selects a house and answers a question correctly	House unlocks, and score updates
11	Play Scene	Player selects a house and answers a question incorrectly	Display an error message
12	Play Scene.	Player presses 'Hint' button	Display a personalized hint. progress

13	Logout	Player presses 'Logout' button	Session ends, navigate to home page
14	Practice Scene	Player selects 'Yes' for practice mode	Navigate to the practice game with sample questions
15	Practice Scene	Player selects 'No' for practice mode.	Navigate back to the previous screen
16	Help Panel	Press the 'Help' button	Display the help panel with clear instructions

10.Operation & Maintenance Guide

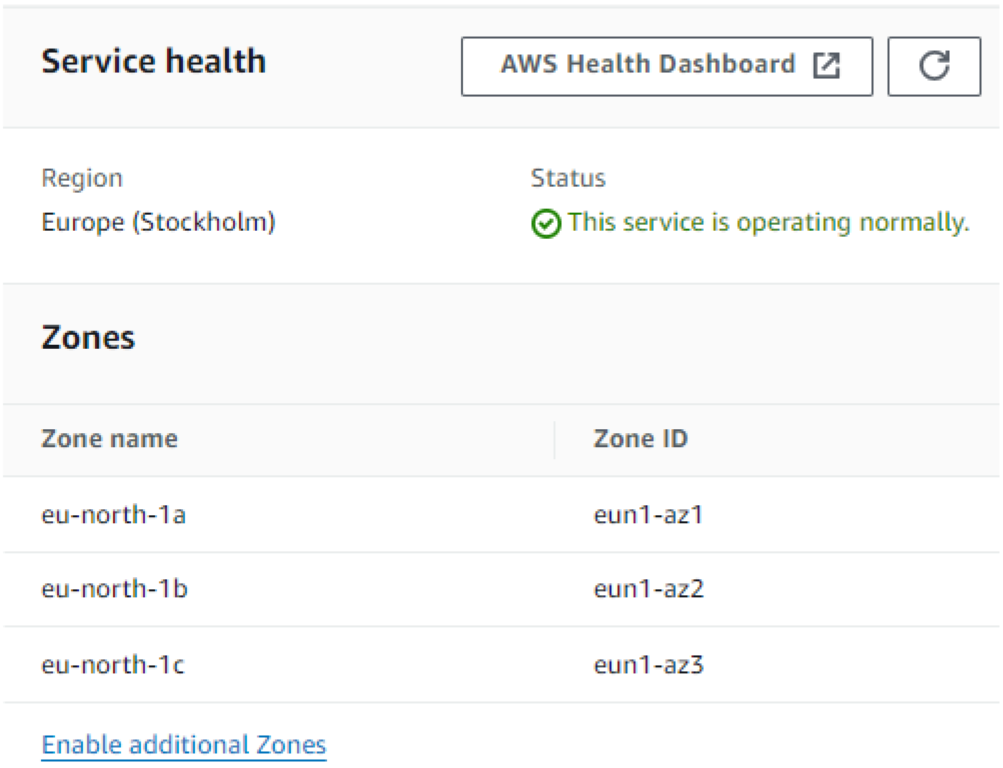
10.1 Server

We chose AWS for our project because it offers industry-standard cloud services and provides a free tier that allows us to use many of its products at no cost for the first year. We selected the Stockholm region (eu-north-1) as it is geographically well-suited to our target audience, ensuring lower latency and better performance. While there is a newer AWS region in Tel Aviv, Israel, it is not included in the free tier, making Stockholm the more cost-effective choice for our project.



Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
MySQL-EC2-In...	i-05561a573e1b70460	Running	t3.micro	3/3 checks passec	View alarms	eu-north-1b	ec2-16-171-142-195.eu...	16.171.142.195	–	–



Service health

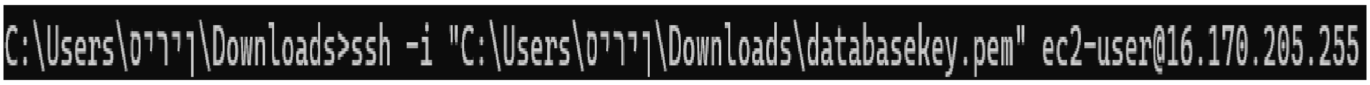
Region	Status
Europe (Stockholm)	This service is operating normally.

Zones

Zone name	Zone ID
eu-north-1a	eun1-az1
eu-north-1b	eun1-az2
eu-north-1c	eun1-az3

10.1.1 Accessing the server using SSH

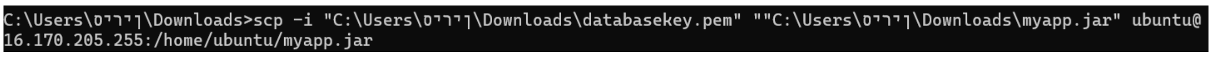
SSH, or Secure Shell, is a protocol that provides a secure method for remote access and management of servers over unsecured networks. It uses strong encryption to ensure both privacy and data integrity. In our project, we use SSH to securely log into our AWS EC2 instance, allowing us to manage the server remotely. This is essential for tasks such as updating the game server, configuring services, and running necessary scripts. We executed the SSH command in the terminal, with the first argument being the private key, followed by the EC2 user and the instance's public IP address, enabling secure access to our project’s infrastructure.



```
C:\Users\סירין\Downloads>ssh -i "C:\Users\סירין\Downloads\databasekey.pem" ec2-user@16.170.205.255
```

10.1.2 Uploading the Server

After writing and locally testing the server code for our project, we created a JAR file. We then deployed the JAR to our AWS EC2 instance using the SCP (Secure Copy Protocol). SCP allows us to securely transfer files between our local machine and the remote server over an SSH connection, providing the same security and authentication as SSH. The SCP command we used included several arguments: `-i /pathToPrivate-key.pem` specifies the SSH key for secure authentication, `/pathToFile.jar` is the path to the JAR file being transferred, `ubuntu@your-instance-ip` represents the user and IP address of the remote EC2 instance, and `/pathToPut/file.jar` is the destination path on the remote server where the JAR file was placed.



```
C:\Users\סירין\Downloads>scp -i "C:\Users\סירין\Downloads\databasekey.pem" ""C:\Users\סירין\Downloads\myapp.jar" ubuntu@
16.170.205.255:/home/ubuntu/myapp.jar
```

```
ubuntu@ip-172-31-35-238:~$
```

10.1.3 Running the Server

After connecting to the remote machine (through SSH) we see the JAR file of the server.

10.1.4 Database

Our database is a MySQL instance hosted on Amazon RDS. It contains several tables, including 'users' for storing user credentials, 'questions' for managing the quizzes and riddles within the game, and 'hints' for tracking the personalized hints provided to players. The database supports account creation, login functionality, and the storage of game-related data for a comprehensive player experience.

10.2 Client

To connect the client to the server in our project, we need to update the serverURL in the to point to the public IP address of our AWS EC2 instance.

10.2.1 Server Connection Overview

In 'House of Riddles using ML,' the client communicates directly with the server to manage gameplay interactions, such as fetching questions, submitting answers, and tracking player progress. The server processes these requests and sends responses to ensure a smooth single-player experience.

10.2.2 Monitoring and Metrics

AWS CloudWatch provides essential tools for monitoring our server’s performance. Regular reviews of these metrics help us maintain server efficiency and identify potential issues like server overload or bandwidth spikes. The key metrics we monitor include:

- All Requests: Total number of requests made to the server by the client, such as login attempts, gameplay actions, and hint requests.
- Requests by Type: Frequency of specific request types (e.g., login, submit answer, request hint).
- Bandwidth Consumption: Total and average data transmitted between the client and server.
- Server Uptime and Response Time: Monitoring server availability and how quickly it responds to client requests to ensure smooth gameplay.

10.2.3 Server Communication Overview

In 'House of Riddles using ML,' the game operates entirely offline. The client interacts with local data to manage the player's progress, provide questions, and display hints. Functions such as SubmitAnswer and RequestHint handle these interactions without requiring an internet connection.

10.2.4 Function Maintenance Tips

- SubmitAnswer: Ensure that the game correctly processes player answers and updates their progress locally. Verify that all player data is saved correctly on the device.
- RequestHint: Confirm that hints are retrieved based on the player's current quiz and stored data, ensuring they are provided accurately during gameplay.

10.2.5 Troubleshooting Common Issues

- Data Processing Errors:

  Problem: Errors when submitting answers may prevent the game from registering player progress.

  Solution: Implement proper error handling in the `SubmitAnswer` function to provide useful feedback to the player.

- Hint Retrieval Errors:

  Problem: Issues retrieving hints can leave the player without assistance during quizzes.

  Solution: Ensure that hints are stored correctly and accessible based on the current game state.

10.2.6 Maintenance Procedures

Even though the game operates offline, regular maintenance tasks are essential for ensuring the local systems work as intended:

- Update Systems: Regularly review and update the game to handle new features or bug fixes.
- Data Backup: Ensure that player progress and game data are saved locally to prevent loss of data.
- Error Handling: Implement thorough error handling within the game to provide feedback to the player and ensure smooth operation during gameplay.

11.Results & Conclusions

We discovered that refining the reinforcement learning (RL) model to enhance the educational impact of the game was necessary but very time-consuming. Training the model in a straightforward environment took significant time, limiting our ability to focus on other important aspects of development, highlighting the need for better time management. Moreover, while the RL model effectively learns player preferences and difficulties, it currently tends to favor a narrow set of responses to player behavior. This suggests a need for more diverse training scenarios to help the model recognize and adapt to a wider range of player strategies and problem-solving approaches, allowing it to provide personalized challenges that reflect individual learning styles and preferences rather than focusing on a single.

12.Further improvement

We discovered that refining the reinforcement learning (RL) model to enhance the educational impact of the game was essential but required extensive time and effort. Training the model in a simplified environment consumed significant time, which constrained our ability to focus on other crucial aspects of development, highlighting the importance of improved time management. Although the RL model effectively generates optimal strategies for solving quizzes, it currently tends to rely on a single approach, which limits its adaptability. This limitation underscores the need for more diverse training scenarios that mirror the variety of strategies players might use in real-world gameplay.

Additionally, while the RL model generates correct solutions, it does not yet fully account for the varying complexities of different quizzes or unexpected obstacles players may encounter. Our objective is to further train the model to adapt to these variables, allowing it to provide more nuanced, context-aware recommendations that better reflect the challenges players face. While this additional training will demand significant time and resources, the resulting improvements will enhance the model’s ability to offer more relevant and personalized quiz-solving strategies.

Looking ahead, we aim to further improve the game by introducing multiplayer functionality. This addition will enable players to engage with others in real-time, providing an opportunity to collaborate or compete in solving quizzes. Incorporating multiplayer features will not only enrich the gameplay experience but also promote social interaction and learning through shared problem-solving.

13.References

1. https://learn.unity.com/
2. https://www.languagezen.com/pt/about/english/Duolingo_Efficacy_Study.pdf
3. https://www.mdpi.com/2078-2489/12/4/161
4. https://github.com/bascoul/Ml-Agents/blob/master/docs/Training-ML-Agents.md
5. https://arxiv.org/abs/1809.02627
6. https://isprs-annals.copernicus.org/articles/IV-4-W4/161/2017/isprs-annals-IV-4-W4-161-2017.pdf
7. https://books.google.co.il/books?hl=ar&lr=&id=OMNiDwAAQBAJ&oi=fnd&pg=PP1&dq=Reinforcement+Learning+Algorithms+with+unity+game&ots=CkBONy-A92&sig=OHdCTANtyNth_N5-Auc_sduxZq8&redir_esc=y#v=onepage&q=Reinforcement%20Learning%20Algorithms%20with%20unity%20game&f=false
8. https://www.youtube.com/watch?v=zPFU30tbyKs
9. https://www.theseus.fi/bitstream/handle/10024/496604/Breugelmans_Arno.pdf?sequence=2&isAllowed=y