



CSN-254
Software Engineering
Group-4
Project Report

under

Prof. Sandeep Kumar Garg

Date: 26 April, 2024

Contributions

Name: Meet Sindhav

Roll No: 22114053

Email: meet_js@cs.iitr.ac.in

Mobile No: 7984221079

Contribution: Worked on gameplay design, modifications, character design and UI of the game. Also, integrated audio and visual assets for better user experience.

Name: Mohammed Haaziq

Roll No: 22114055

Email: mohammed_hj@cs.iitr.ac.in

Mobile No: 8126061554

Contribution: Controller application development specifically in establishing connection between game and controller via web sockets

Name: Aditya Mundada

Roll No: 22114058

Email: mundada_am@cs.iitr.ac.in

Mobile No: 7720079142

Contribution: Working in game mechanics, implementing attack and movement features, adding special abilities to players in the game and mapping websocket data recieved to required game data.

Name: Nayan Kakade

Roll No: 22114060

Email: nayan_rk@cs.iitr.ac.in

Mobile No: 7420936972

Contribution: Developing the welcome page and IP Address key page of the controller application, working on websocket connection on controller side, and integration of game and controller.

Name: Polasa Hariharan

Roll No: 22114068

Email: polasa_h@cs.iitr.ac.in

Mobile No: 7670839921

Contribution: Controller application front-end development (main page) and integration with web-sockets, game mechanics design

Name: Roopam Taneja

Roll No: 22125030

Email: roopam_t@cs.iitr.ac.in

Mobile No: 9810788629

Contribution: Adding websocket functionality to game controls while adding necessary features for ensuring seamless integration with app. Also contributed in adding characters and designing different screens in the game.

BattleBlade: Smartphone console based gaming experience

Introduction

In today's modern era, gaming is becoming an increasingly popular hobby. There is a sudden emergence in the number of people who indulge in gaming online with their friends as well as casual gamers who play games in their leisure time. Currently, the gaming services available are of three types:

- Playing online games on the phone.
- Playing online games on PC/laptops.
- Playing games using gaming consoles with friends.

The third category of gaming services is a very exciting domain. This is because it is dominated majorly by hardware consoles that are very expensive and not available easily for casual gamers-teenagers and young adults who want to just spend fun time with friends and play these games in groups.

Aim

We propose a one stop solution for these customers: **BattleBlade**

A very fun and intuitive fighting game developed by us that runs on the PC, with multiple characters, great moves and multiple arenas, all of them controlled by a controller android app developed by us specially for this game. This controller incorporates all of the user requirements and two players can simultaneously play this fighting game against each other and enjoy. This eliminates the need for expensive consoles and solves a big headache for casual gamers. This creates a "Small To Big Screen Adventure" for our users where users can play the game on a big laptop screen and control it with phones on their fingertips with the only need being a common Internet connection.

Proposed System

The key features of our system:

1. **Smartphone-Controlled Multiplayer Gameplay:** The system enables two-player gameplay where smartphones serve as controllers. This occurs through an android app with great Dpad and attack controls that also support haptic feedback. The controller icons are very intuitive so that any person can understand just by looking at buttons as to what use they are of.
2. **Character Selection:** Users can select characters from a grid displayed on the screen. The various characters added include Batman, Superman, Psyduck and characters from some popular gaming and comic universes which players love controlling and managing. This feature enhances user engagement and contributes to a more personalized gaming experience.
3. **Real-Time WebSocket Communication:** The system utilizes WebSocket communication for real-time interaction between the game server and connected clients (smartphones). This ensures seamless communication, enabling responsive gameplay and synchronization between players. WebSocket connections are connections where all we need is the same ip address for all devices that must communicate.

4. **Dynamic Backgrounds and Character Animation:** The game incorporates dynamic backgrounds and character animations, enhancing visual appeal and immersion. We provide backgrounds of various types from forest to snow to beaches and more. These elements contribute to a captivating gaming experience, keeping players engaged throughout gameplay sessions.
5. **Various attack types:** The players can choose multiple types of attacks, some that are more powerful than the other but can only load after certain time intervals. Thus more powerful attacks are available less frequently. Also players can choose to use superpowers they can become invincible to attacks for five seconds or they may choose to boost their health to 50 percent, if they are below it. However, these abilities can only be used once per round and this makes the game more strategic keeping players involved throughout.
6. **Scalability and Flexibility:** The system's architecture is designed to be scalable and flexible, accommodating potential expansion in terms of characters, backgrounds, and game features. This scalability ensures that the game can evolve over time, providing continuous entertainment to players.
7. **Security:** The game requires us to connect our controllers to the PC via IP address verification. To keep Ip address secure we have encrypted it to form a game key using Vignere cipher and its Beaufort variant. Also we verify number of players connected and disallow more than 2 users to be connected to the game to avoid unnecessary security issues.

Algorithms Used

There are multiple algorithms we have used in this project from Vignere cipher for encoding and decoding ip address to handling websockets. Here are the details of a few of them below:

Algorithms boil down to the very minute detail of our project like in our controller buttons we have used vibrate which uses an asynchronous function called Future which is used to represent a potential result or error that will be available at some point in the future. This asynchronous function checks if the device has a vibrator. If available, it triggers a brief 25-ms vibration. For implementing the hold feature in the buttons we repeatedly send the control key as per the timer.

For encryption of the ip address we use the Beaufort variant of the Vignere cipher.

Vignere cipher: The Vigenère cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. It employs a keyword to determine the shift for each letter in the plaintext, allowing for more secure encryption than simpler substitution ciphers. To encrypt, the sender selects a keyword and repeats it to match the length of the plaintext. Then, each letter of the plaintext is shifted according to the corresponding letter in the keyword. The receiver decrypts the message using the same keyword.

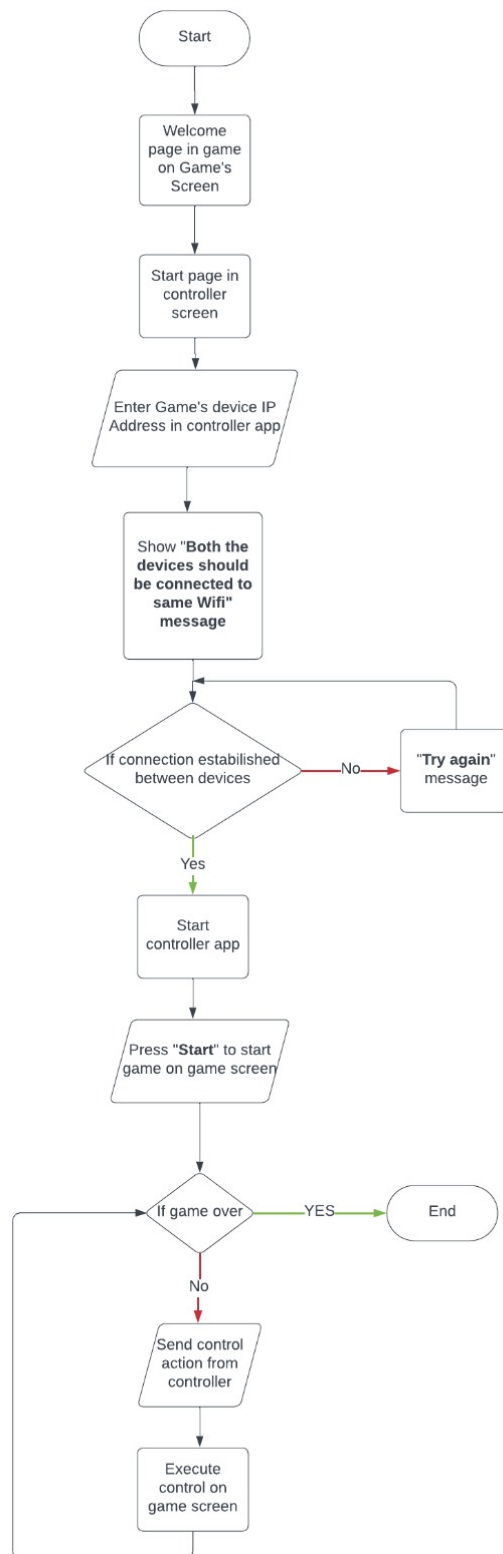
Innovation and Application

1. **Accessibility Enhancement:** By eliminating the need for traditional hardware consoles and leveraging smartphones as controllers, the system enhances accessibility for casual gamers. This innovation democratizes gaming, making it more accessible to a wider audience.
2. **Immersive Multiplayer Experience:** The combination of smartphone-controlled multiplayer gameplay, dynamic backgrounds, and character animations creates an immersive gaming ex-

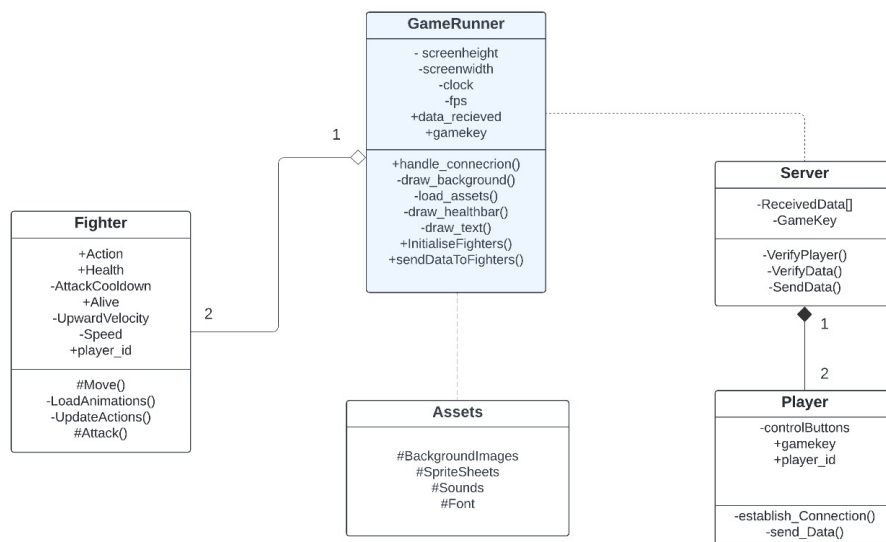
perience. Players can engage in intense battles with friends or family members, fostering social interaction and camaraderie.

3. **Cross-Platform Compatibility:** The system's use of smartphones as controllers promotes cross-platform compatibility, allowing users with various devices to participate in multiplayer gaming sessions seamlessly. This feature expands the reach of the game beyond specific gaming platforms, catering to a diverse user base.
4. **Real-Time Interaction:** The real-time WebSocket communication facilitates instant interaction between players, enabling responsive gameplay and fostering competitive dynamics. This real-time aspect adds excitement and intensity to multiplayer battles, enhancing the overall gaming experience.
5. **Adaptability to Emerging Technologies:** The system's design allows for adaptation to emerging technologies and trends in the gaming industry. It can integrate future advancements such as augmented reality (AR) or virtual reality (VR), further enriching the gameplay experience and staying relevant in a rapidly evolving landscape.

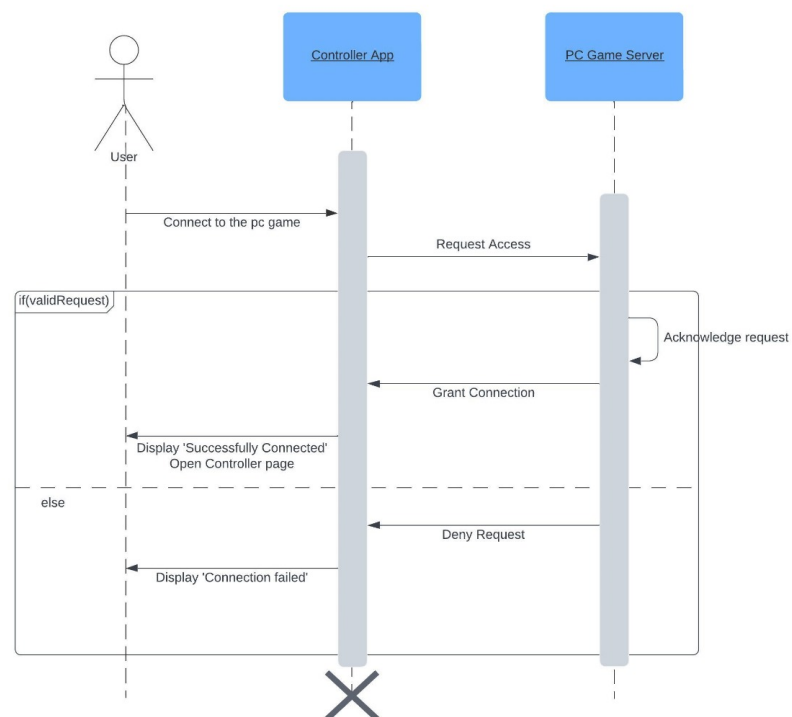
Diagrams

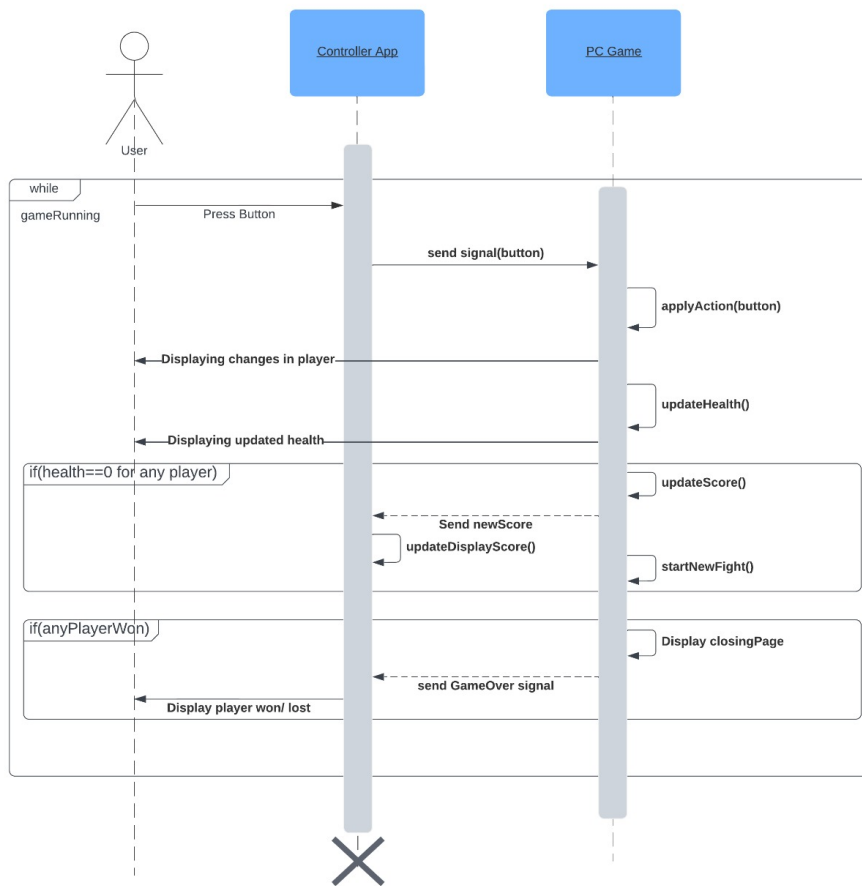


FlowChart

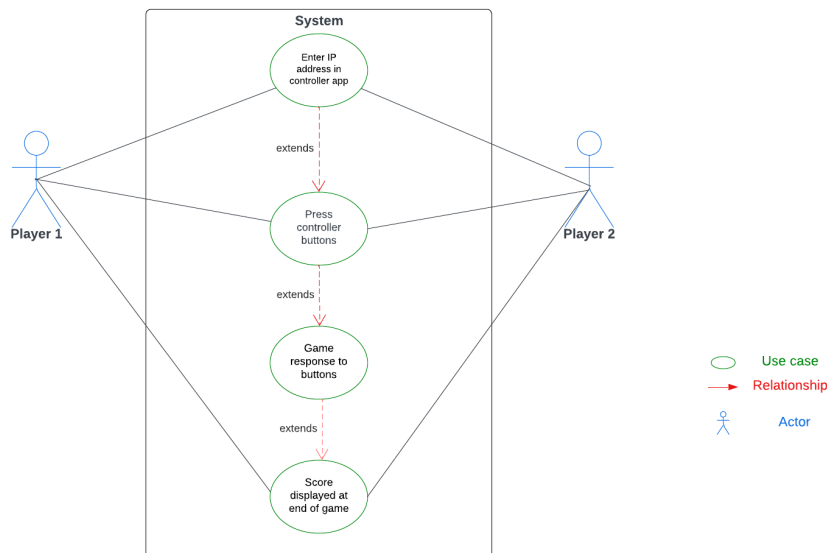


Class Diagram





Sequence Diagram



Use Case Diagram

Implementation

Fighter Game

1. We have designed the game using pygame, an inbuilt library of python.
2. The game starts with a graphic entry of the app logo along with the welcome page. The game encrypts the ip of the computer to generate a game key.
3. Only when two players connect , by entering the correct game key then we move ahead to the select screen where both players can choose characters of their choice. This is checked using player select and player connected variables which check each incoming request as to whether it follows the required json encoding corresponding to these options. Thus, every movement of the player and every game decision occurs if the specified mapping of that request is encoded correctly and sent by the controller.
4. Then the game begins. The game consists of 5 rounds and the player who wins 3 out of 5 rounds first wins. Each round starts in a different arena that has great visual appeal. Further the narrator also adds his voice at certain occasions that add spice to the game.
5. Each player has his own unique way of attacking. Even the attacks are of two types. We keep track of what the player wants to do and move the character animations accordingly. The health of the characters is reduced according to the intensity of the attack in the backend.
6. Each character is an object of the fighter class which has attributes corresponding to game dynamics: the object's health, location, size, superpower usage, invincibility criteria, speed. In order to make the moves smooth and allow for the next move to enter animation only after the previous move has completed we have attack cooldown variables which do just that.
7. The superpowers of healthboost and invincibility have also been implemented similarly.
8. After a player wins we simply move to the end screen and showcase winner details.

Controller App

1. We have designed the controller mobile application using flutter. We have various pages in our app.
2. On starting the app, you are welcomed with a Get Started page.
3. Once you click, you go to the IPAddress page where you have to enter the encrypted key of the ip address displayed on the game screen. After you enter the encrypted key, in the backend of the app, the encrypted key is decrypted using the decrypt function for the cipher text. The function will generate the ip address. After this, it will try to connect to the game through a websocket connection using the decrypted ip address. If the ip address is correct, the connection will be established successfully and a success popup will appear, on clicking which you will be directed to the selection page of the app. If the ip address is incorrect, then a failure popup will appear.

4. On arriving on the selection page, selection buttons (right, left, up and down) will be available for you to select characters visible on the game screen. Both the players can select their characters and click on the Let's Play button on the selection page.
5. On clicking the button you will be directed to the controller page. On the controller page, you have a DPAD (having right, left and jump buttons) on the left side and special buttons (two types of attack buttons, one shield button and one heal button) on the right side. DPAD buttons on the left side are for movement (right, left and jump) in the game.
6. Out of the two types of attack buttons, one is normal attack and another is special attack. The special functionality of the second attack button is that it provides more damage compared to the first button but the limitation implemented is that the special attack once clicked, becomes disabled for some amount of time and gets enabled for use after some time.
7. Next is the shield button which creates a shield around you for some time during which no attack can provide damage to you.
8. At the last, there is the heal button which can increase the health to 50 percent of the total health, once health becomes less than 50 percent.
9. The message sending through websocket channel is achieved using the sendControl function which sends the messages as and when the button is clicked. But for the DPAD buttons, we have also implemented one more function when we have to continuously tap the movement buttons since players require smooth and fast movement during the game. This function is the StartSendingRepeatedly function which sends the messages through the websocket channel every 10 milliseconds if we continuously press the button. This ensures smooth movement of the fighter in the game.

Tools and Technologies Used

Pygame

1. Pygame is a popular Python library used for creating video games and multimedia applications. It provides functionalities for handling graphics, sound, input devices like keyboards and mice, and more.
2. We have used pygame for developing the fighter game.
3. Programming Language - **Python**

Flutter

1. Flutter is an open-source UI software development toolkit created by Google. It is used for building natively compiled applications for mobile, web, and desktop from a single codebase.
2. We have used flutter for developing the controller mobile application.
3. Programming Language - **Dart**

Websocket

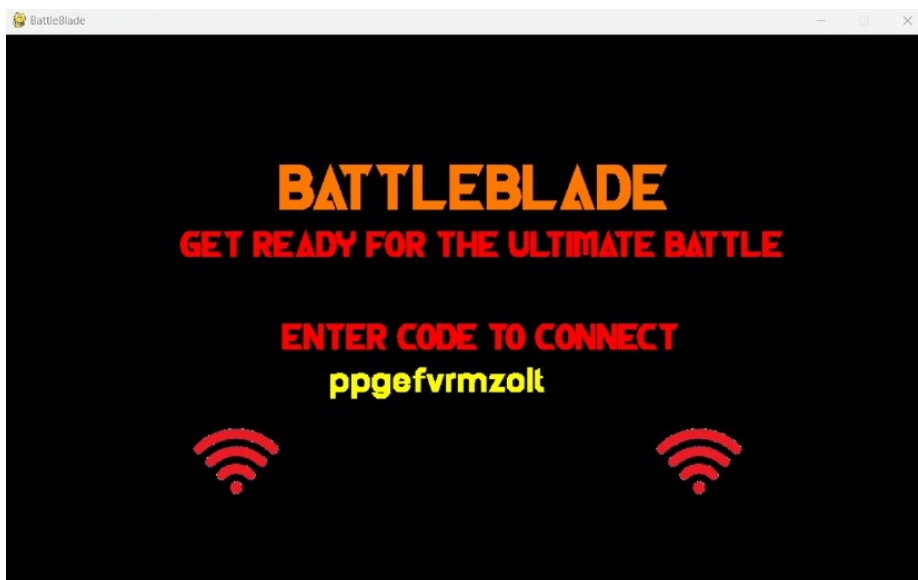
1. WebSocket is a communication protocol that provides full-duplex communication channels over a single, long-lived TCP connection. It enables real-time, bidirectional communication between clients (such as web browsers or mobile apps) and servers, allowing for interactive web applications that can send and receive data asynchronously without the need for constant polling.
2. We have used websocket to establish and maintain a wireless connection between the game and controller mobile app for real time message sharing between the two devices.
3. Programming Language - **Python**(Game side) and **Dart**(Controller Side)

Asyncio

1. Asyncio is a powerful library in Python that enables concurrent execution of asynchronous I/O-bound tasks. It leverages coroutines, which are functions that can pause and resume their execution, allowing for efficient handling of multiple operations without blocking the program's flow. Its event loop mechanism efficiently manages task scheduling and ensures optimal resource utilization, making asyncio a valuable tool for building responsive and efficient Python applications.
2. We have used async programming in the game so that we can trigger character movements asynchronously as we receive move requests from controller even when game environment is active continuously.
3. Programming Language - **Python**

Screenshots of the System

Game



Welcome Page

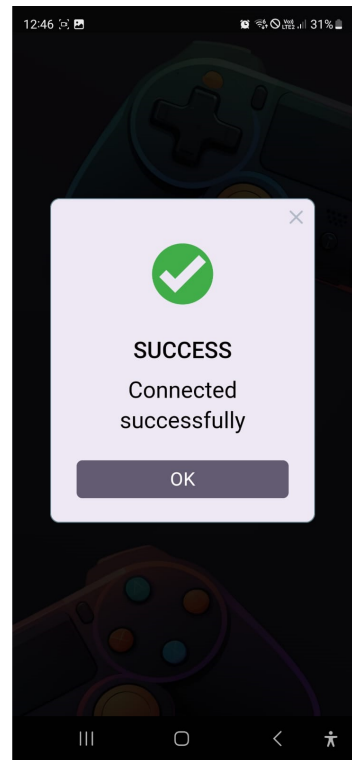
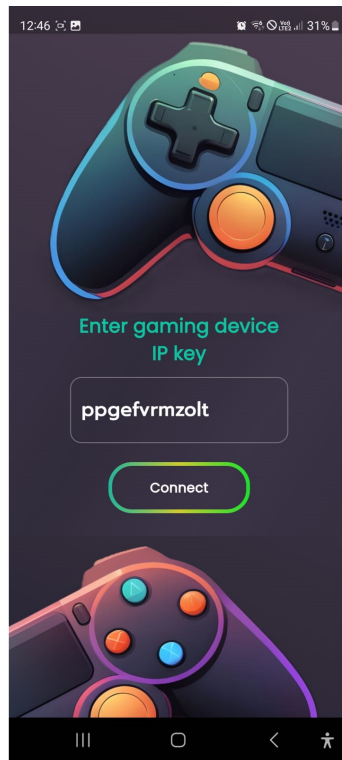
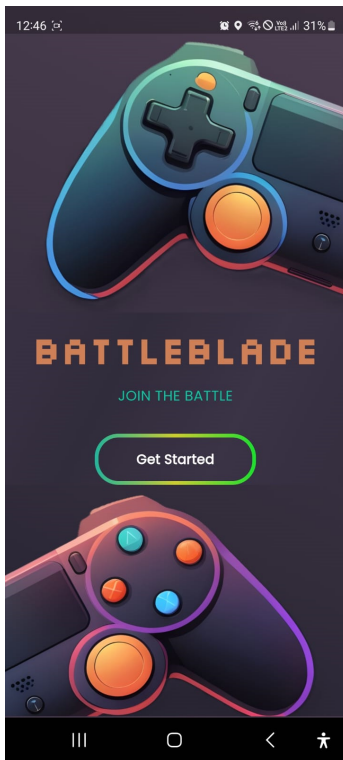


Character Selection Page

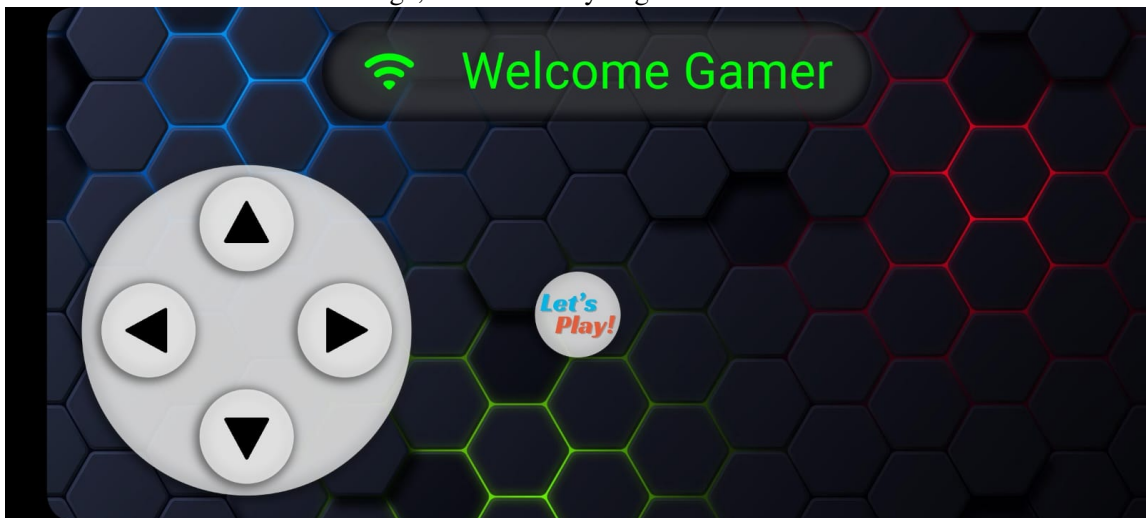


Gameplay

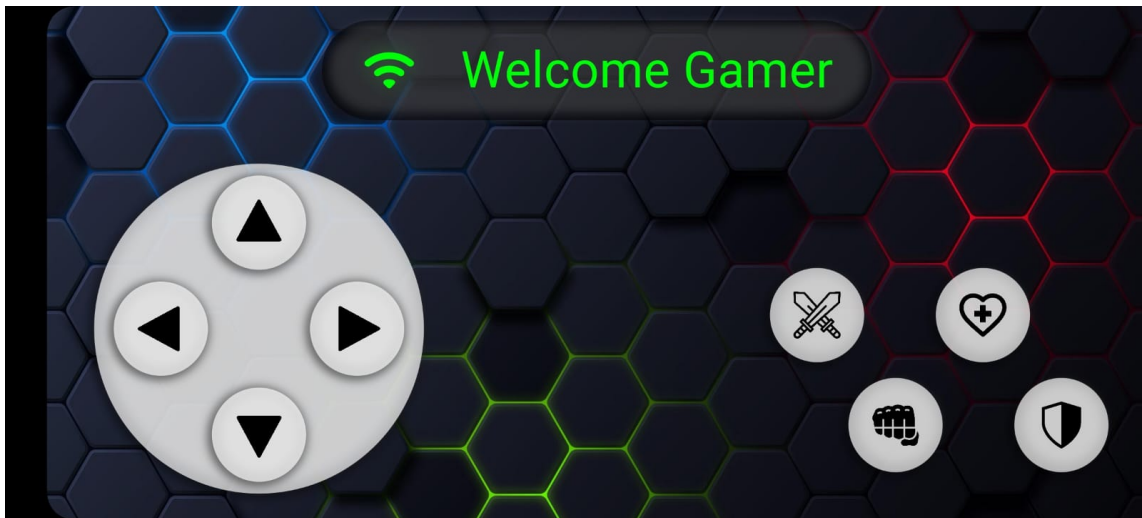
Controller Mobile App



Welcome Page, IPAddress key Page



Character selection controller buttons



Gameplay controller buttons

Comparison with existing similar Systems

We will compare **BattleBlade** with three established multiplayer gaming solutions from different domains: Traditional consoles like PlayStation and Xbox, Mobile gaming applications and AirConsole

- **Traditional consoles like PlayStation and Xbox:**

Strengths

1. **Established Brand:** PlayStation has a strong brand presence and loyal user base, built over decades of successful gaming consoles.
2. **Exclusive Titles:** PlayStation and Xbox often secure exclusive rights to popular game titles, attracting gamers who want access to unique content.
3. **Reliable Hardware:** Consoles like PlayStation and Xbox are known for their reliable hardware and controller design, ensuring a consistent gaming experience.

Weaknesses

1. **Cost:** Traditional consoles come with a high initial cost, which may deter casual gamers or those on a budget.
2. **Lack of Accessibility or Availability:** Gamers need to purchase the console and physical controllers, which may not be possible in remote areas.
3. **Limited Mobility:** Consoles are stationary devices, limiting gameplay to specific locations and environments.

- **Mobile Gaming Apps:**

Strengths

1. **Accessibility:** Mobile gaming apps are widely accessible to a large audience with smart-phones, making it easy for users to download and play games
2. **Low Barrier to Entry:** Many mobile games are free to play or offer low-cost options, making them accessible to a wide range of players.
3. **Convenience:** Mobile gaming allows players to enjoy games on the go, offering flexibility and convenience.

Weaknesses

1. **Limited Graphics and Gameplay:** : Mobile games may have limitations in terms of graphics quality and gameplay depth compared to console or PC games
2. **Lack of convenience :** Multiple players clicking buttons on the same screen can be quite inconvenient and spoil the gaming experience. Lags may further ruin the experience.
3. **Battery Life:** Extensive gaming on mobile devices can drain battery life quickly, requiring frequent recharging.

• Air Console:

Strengths

1. **Cross-Platform Compatibility:** AirConsole allows players to use smartphones as controllers to play games displayed on a larger screen, offering a seamless cross-platform gaming experience.
2. **Convenience:** Separate devices as consoles enhance the experience of gamers and avoid the load on a single device by distributing it on different smartphones.
3. **Accessibility:** Players can easily access AirConsole through a web browser on their PC or smart TV, eliminating the need for additional hardware.

Weaknesses

1. **Paywall on many features:** Several features like access to all games, no ads and unlimited players can be unlocked only by a paid subscription which acts as a huge deterrent.
2. **Dependency on Internet Connection :** AirConsole requires a stable internet connection for gameplay, which may be a limitation in areas with poor connectivity.
3. **Potential Latency Issues:** Depending on internet speeds and connectivity, players may experience latency or lag during gameplay, affecting the overall gaming experience.

Conclusion

In summary, our two-month project has delivered promising outcomes, showcasing notable strengths such as seamless smartphone integration, cost-effectiveness, and cross-platform compatibility. Leveraging smartphones as controllers not only enhances accessibility but also significantly reduces financial barriers for players. Moreover, the game's compatibility across various smartphone models and PC operating systems widens its reach to a diverse audience.

However, we acknowledge key weaknesses that demand attention for future improvements. Building brand recognition in a competitive gaming market requires substantial marketing efforts, while potential latency issues and user adoption hurdles highlight the necessity for further optimization.

Despite these challenges, our team remains dedicated to refining our strategies and technology to overcome these weaknesses and enhance the project's success in the long run.

Bibliography

1. <https://docs.flutter.dev/>
2. https://pub.dev/packages/web_socket_channel
3. <https://websockets.readthedocs.io/en/stable/>
4. <https://docs.flutter.dev/cookbook/networking/web-sockets>
5. <https://docs.python.org/3/library/asyncio.html>
6. <https://www.pygame.org/docs/>
7. <https://www.python.org/doc/>