Capstone Project: ft_transcendence

Authors:

Malik AlZubaidi - `maalzuba`
Paula Balbino - `pbalbino`
Samih Abdelrahman - `sabdelra`
Tanvir Ahmed Anas - `tanas`
Tehuan Melo - `tde-melo`

Submittion Date: Nov. 28th 2024

Abstract

This project focuses on developing a web application on which users can play Pong contests by leveraging modern technologies across both backend and frontend development. The core backend is built using Django, with PostgreSQL serving as the database to ensure data integrity and seamless integration. Nginx is optimized for serving static files efficiently, reducing server load and improving performance. Additionally, Nginx enforces TLS to secure data transmission, providing robust protection against man-in-the-middle attacks and ensuring encrypted connections. The frontend is crafted using the Bootstrap toolkit, ensuring responsive and user-friendly interfaces across all devices. A key feature of the project is the implementation of two ways to login (42 login or username) allowing secure remote sign-ins, complemented by Two-Factor Authentication (2FA) and JSON Web Tokens (JWT) for an extra layer of security. The application supports multiplayer functionalities, allowing two or four players to engage in live-controlled games with customizable options. Features such as paddle and ball speed, difficulty levels, and custom score enhance user experience. In addition to security and customization, the project emphasizes accessibility and responsiveness. The platform is optimized for all devices and browsers, with specific support for Firefox and accessibility enhancements for visually impaired users. Server-Side Rendering (SSR) is employed to improve performance and SEO, ensuring fast loading times and content pre-rendering for a consistent user experience. An additional game, Tic-Tac-Toe, is introduced, expanding the application's gaming offerings. Finally, user management is centralized through secure login, profile customization, and tournament participation, enhancing community engagement through friends lists, online status, and detailed match history. This project demonstrates the integration of secure authentication, scalable architecture, and a highly customizable gaming experience, with a strong emphasis on user experience and accessibility.

Contents

List of Figures

List of Tables

List of Abbreviations

2FA Two-Factor Authentication

CI/CD Continuous Integration and Continuous Delivery/Deployment

DOM Document Object Model

DevOps Development Operations

Git Global Information Tracker

JWT JSON Web Tokens

OAuth Open Authorization

OTP One-Time Password

SPA Single Page Application

SDLC Software Development Life Cycle

SSR Server-Side Rendering

SQL Structured Query Language

TLS Transport Layer Security

UI User Interface

URL Uniform Resource Locator

XSS Cross-Site Scripting

Chapter 1

Introduction

1.1 Introduction

This project aims to provide a highly interactive and accessible multiplayer gaming experience through the development of a web application centered around the classic game of Pong, along with the addition of a new game, Tic-Tac-Toe. By incorporating user history tracking and a matchmaking system, the platform not only diversifies its offerings but also enhances user engagement through gameplay statistics. The system ensures that game history and matchmaking data are securely stored and kept up to date, while also optimizing performance and responsiveness for an enjoyable experience. Combining modern web technologies and best practices in both backend and frontend development, the project seeks to deliver a responsive, scalable, and secure platform where users can engage in real-time contests. Standard user management features include secure registration, login, profile updates, tournament participation, and friends lists, with additional functionalities like avatar uploads, unique display names, and online status visibility. User profiles provide insights into gameplay statistics, such as wins, losses, and match history, accessible only to logged-in users. The goal is to offer an immersive and customizable gaming experience that emphasizes performance and security while prioritizing inclusivity, catering to users with visual impairment. Utilizing responsive design principles, and multiple authentication mechanisms, this project aims to push the boundaries of both technical implementation and user experience in web-based gaming applications.

Chapter 2

Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) for this project follows a modified Waterfall model, incorporating Agile elements to enhance flexibility and iterative improvements. While the Waterfall approach provided a structured framework for sequential stages like requirement analysis, design, implementation, and testing, we introduced Agile practices such as ongoing feedback and peer-reviewed GitHub pull requests (PRs). This process of PR reviews allowed team members to give feedback on each other's code, identify potential issues early, and ensure quality and consistency across different modules. By integrating peer-reviewed PRs, we facilitated continuous testing, collaborative refinement, and adherence to best practices throughout the development. This hybrid methodology helped us achieve a balance of systematic progression with opportunities for iteration and improvement based on team insights.

2.1 Requirement Analysis

Requirement Analysis is the first and one of the most crucial phases of the Software Development Life Cycle (SDLC). During this phase, the focus is on gathering detailed requirements and defining the project's objectives. This phase helps ensure that the project starts on the right track, with a clear understanding of what needs to be built, why it is being built, and how it will be used. Proper requirement analysis sets the foundation for all subsequent stages of the software development process. By understanding and addressing potential risks early, such as technical feasibility, resource constraints, or user expectations, the project can avoid delays or even a potential failure. Additionally, proper documentation and agreement on requirements ensure that all members are on the same page, reducing miscommunication and conflicts later. Starting with a solid foundation minimizes misunderstandings and scope changes and sets the stage for successful project execution.

In order to properly gather the requirements, it was necessary to choose which modules best fit the team skills and interests. Initially some modules were set, and others were flagged as “interested”, that the team would do according to the project development.

2.2 Modules Implemented

After a careful analysis, the following modules were implemented:

- **Web**
  - **Major module:** Use a Framework as backend.
  - **Minor module:** Use a front-end framework or toolkit.
  - **Minor module:** Use a database for the backend.
- **User Management**
  - **Major module:** Standard user management, authentication, users across tournaments.
  - **Major module:** Implementing a remote authentication.
- **Gameplay and user experience**
  - **Major module:** Multiplayers (more than 2 in the same game).
  - **Major module:** Add Another Game with User History and Matchmaking.
  - **Minor module:** Game Customization Options.
- **Cybersecurity**
  - **Major module:** Implement Two-Factor Authentication (2FA) and JWT.
- **Accessibility**
  - **Minor module:** Support on all devices.
  - **Minor module:** Expanding Browser Compatibility.
  - **Minor module:** Add accessibility for Visually Impaired Users.
  - **Minor module:** Server-Side Rendering (SSR) Integration.

To achieve 100% project completion, at least seven major modules were required, with each major module being equivalent to two minor modules. The additional modules done are considered bonus points and maximum score reachable is 125%.

2.3 Design the System Architecture

The Design Phase is the second major stage in the SDLC, where the focus transitions from defining system requirements to designing how the system will meet those requirements. The project architecture is divided into independent services, each running in Docker containers, to enhance scalability and maintainability.

For the database, `PostgreSQL` is selected to ensure data integrity and seamless integration with `Django`, while `Bootstrap` is utilized for creating a responsive and accessible UI design on the frontend.

In terms of security design, secure login methods, including 42 login or username, with Two-Factor Authentication (2FA) and JSON Web Tokens (JWT) for authentication, are implemented.

During this stage, the initial draft of the website workflow and the structure of page linkages were established in a group meeting. This draft outlined user navigation through the site, detailing the connections between different pages to ensure a coherent user experience. This early planning phase was crucial for aligning development efforts across various teams, such as frontend design, backend functionality, and game integration, setting the foundation for subsequent implementation stages.

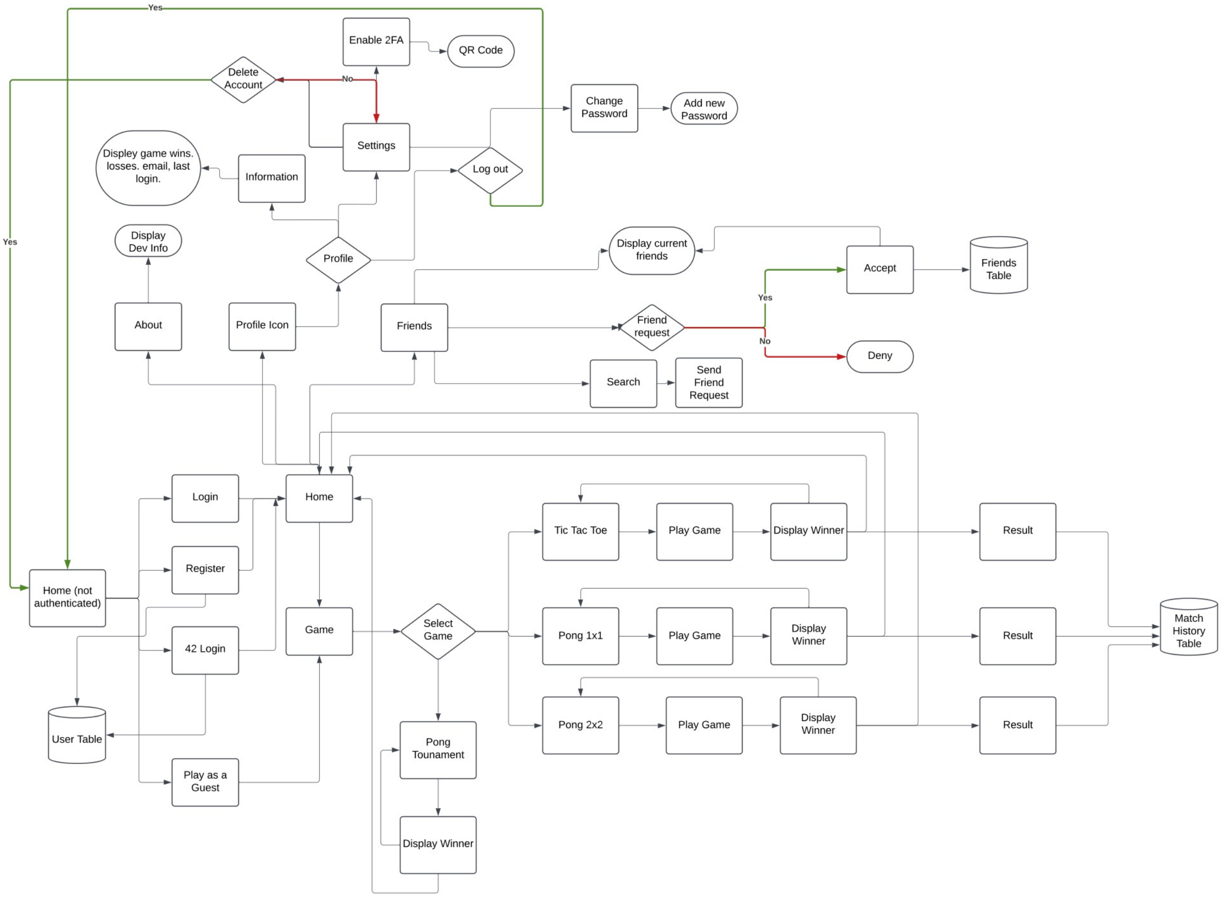


FIGURE 2.1: Workflow

2.4 Implementation

The implementation step involves developing the actual code and functionality based on the design specifications. For this web application project, the implementation step is broken down into the following main aspects:

- **Backend Development:** Development of the core functionality using Django, ensuring the backend supports all game features and security protocols. The backend must also manage user data securely, support multiple game modes, and provide the necessary APIs for frontend interaction. The website is safeguarded against SQL injection and Cross-Site Scripting (XSS) attacks through the inherent security features provided by Django. Specifically, Django mitigates the risk of SQL injection by escaping user inputs and offering secure methods for querying the database. The framework's built-in query functions, such as `.filter()`, `.get()`, and others, incorporate robust mechanisms to prevent the execution of malicious SQL code. Consequently, by using such functions, Django's automatic protection against common security vulnerabilities ensures a safer application environment.
- **Frontend Development:** Build a responsive user interface using Bootstrap, incorporating accessibility features such as high-contrast themes and keyboard navigation using the "tab" key. The frontend should adapt seamlessly across different devices and screen sizes, offering an inclusive experience for users, including those with visual impairments.
- **Game Development:** Implement game mechanics with customization options for different skill levels and settings. This includes developing interactive controls, configuring adjustable game parameters, and ensuring real-time responsiveness. The platform also introduced an additional game, Tic-Tac-Toe, distinct from Pong, to diversify the gaming options available. The game includes user history tracking to record and display individual gameplay statistics.
- **User History and Matchmaking:** Integrate a comprehensive user history tracking system to securely store game history and ensure it is always up to date. These additions aim to enhance user engagement by providing gameplay statistics and facilitating seamless matchmaking for an enjoyable experience.
- **Standard User Management and Authentication:** Implement secure user management features, allowing users to subscribe to the website, log in, and manage their profiles. Users can select unique display names for tournaments, update their information, and upload avatars. Additional features include managing friend lists, viewing online status, and accessing user profiles with stats like wins, losses, and match history. All user management functionalities should comply with best practices for secure authentication.
- To enable simultaneous work on different fronts, such as game development, backend development, and page design, a specific branch was used on GitHub for each task. This approach

allowed team members to work independently while ensuring the integrity of the main branch. Before merging any changes into the main branch, a peer review process was conducted to maintain code quality and catch potential issues early. Figure 2.1 provides an illustration of this workflow, covering the period from October 21st to 24th, 2024, to demonstrate how multiple branches were utilized for collaborative development and quality assurance purposes.

2.5 Testing the application and its functionalities

Testing phase in the SDLC ensures that the web application is functional, secure, user-friendly, and meets all requirements. Focusing on Pong game web application, testing can be broken down into several types, each targeting different aspects of the system:

- **Unit Testing:** Test individual components of the application in isolation to ensure their functionality is correct. Examples include verifying user authentication mechanisms, game logic (like ball and paddle movements), and database interactions. Additionally, unit tests were conducted for the Tic-Tac-Toe game, such as validating game rules, user input, and match outcomes.
- **Integration Testing:** After unit testing, it is necessary to test the interactions between multiple components. Integration testing includes verifying the communication between the frontend and backend (e.g., submitting a login request and receiving the correct response, game management, authentication, user profiles), and ensuring data is correctly stored, retrieved, updated, and deleted from the PostgreSQL database. It was also necessary to perform tests to confirm that the user history and matchmaking system of the game were operating seamlessly, and data flows correctly across services.
- **Functional Testing:** The main goal is to verify that the application meets all functional requirements and works as intended. This is done by testing the game functionality (e.g., in the Pong game, issues like ball collision detection, paddle movement, multiplayer synchronization, adjusting ball and paddle speed, difficulty levels, and sound settings should be analyzed). For authentication and security, it is necessary to check if both login methods (42 login and username), 2FA setup, and JWT session management are properly working, as well as if a non-logged user can access the content restricted to a logged user. In addition, the logout option was also tested to ensure that the session was properly finished. Lastly, user management features like user registration, profile updates, password resets, and tournament participation were analyzed.
- **Usability Testing:** This next step evaluates the application's user interface and experience, ensuring it is intuitive and accessible for users. The main tests were focused on testing the

web application on different devices (desktop, tablet, mobile) to ensure the UI is responsive and easy to use, and checking that all accessibility features (e.g., high-contrast colors, keyboard navigation) work correctly and comply with accessibility standards. After that, the focus shifted to gathering feedback from users on the game's interface, navigation, and overall experience to identify potential improvements.

- **Compatibility Testing:** The goal is to ensure the application functions consistently across different environments. For this application, the browsers "Chrome" (last available version) and "Firefox" were chosen, and in this sense, all tests were performed on both browsers.

In addition to the tests mentioned above, CI/CD was integrated as an automation tool within the GitHub (Figure 2) workflow to prevent commits that would fail the build. This approach proved highly effective, minimizing rework and ensuring that each merge met the minimum standards required for committing to the main branch.

2.6 Evolution

This phase focuses on maintaining the software, enhancing its functionality, fixing bugs, and responding to changing user requirements and technology environments. Evolution ensures that the software remains useful and relevant over time.

2.6.1 Backend Development

- **Initial Phase:** The early development stages focused on setting up the core backend functionality using Django, implementing essential features to support basic game operations, user management, and security protocols. The priority was to establish a secure framework for handling user data and game interactions.
- **Intermediate Phase:** As the project matured, the backend expanded to support multiple game modes, including additional features like dynamic data handling. Using multiple Docker containers allowed independent services (e.g., game management, user profiles) to be developed and scaled separately. The backend was also adapted to handle real-time data exchanges more efficiently, improving the user experience during live gameplay.
- **Current Phase:** The backend now supports a comprehensive API for frontend communication, managing secure user authentication, game history tracking, and matchmaking. It ensures robust security for user data and employs efficient data storage solutions using PostgreSQL.

2.6.2 Frontend Development

- **Initial Phase:** The project began with the development of a basic responsive interface using Bootstrap, ensuring the UI adapted to different devices and screen sizes. Core accessibility features like high-contrast themes when visual impaired mode is selected, and keyboard navigation were implemented from the start.
- **Intermediate Phase:** With user feedback and testing insights, the frontend design was refined to improve usability and inclusivity. Enhancements included optimizing the interface for seamless navigation, adding more sophisticated accessibility features, adjusting the size of the icons in visual impaired mode to appear larger on screen, and analyzing compatibility with the browsers Chrome and Firefox.
- **Current Phase:** The frontend now provides a highly interactive user experience with customizable game settings, smooth transitions, and an adaptive layout. The interface supports real-time updates during gameplay and incorporates comprehensive accessibility measures to accommodate users with visual impairments, ensuring the platform is user-friendly and inclusive.

2.6.3 Game Development

- **Initial Phase:** The first game, Pong, was developed with fundamental game mechanics, focusing on ball and paddle movements, collision detection, and scoring. The game featured basic customization options, such as adjusting paddle speed and difficulty levels.
- **Intermediate Phase:** To enhance user engagement, the game mechanics were expanded with more customization options, including game settings for different skill levels, sound settings, and visual themes. The real-time responsiveness was improved for a smoother gameplay experience.
- **Current Phase:** The platform has diversified its gaming offerings by introducing Tic-Tac-Toe as an additional game option. This addition includes advanced features like user history tracking, which records and displays individual gameplay statistics, and provides players with a more comprehensive gaming experience. The Tic-Tac-Toe game is integrated seamlessly into the platform, allowing users to switch between games.

2.6.4 User History and Matchmaking

- **Initial Phase:** Basic user tracking capabilities were implemented, focusing on recording game results and displaying a simple match history.

- **Intermediate Phase:** The user history system was expanded to track more detailed statistics, such as wins, losses, and game date.
- **Current Phase:** The comprehensive user history tracking now securely stores and displays detailed game history, including past opponents, dates, and scores. The system continuously updates to reflect recent gameplay, and user data is stored in compliance with security standards.

2.6.5 Standard User Management and Authentication

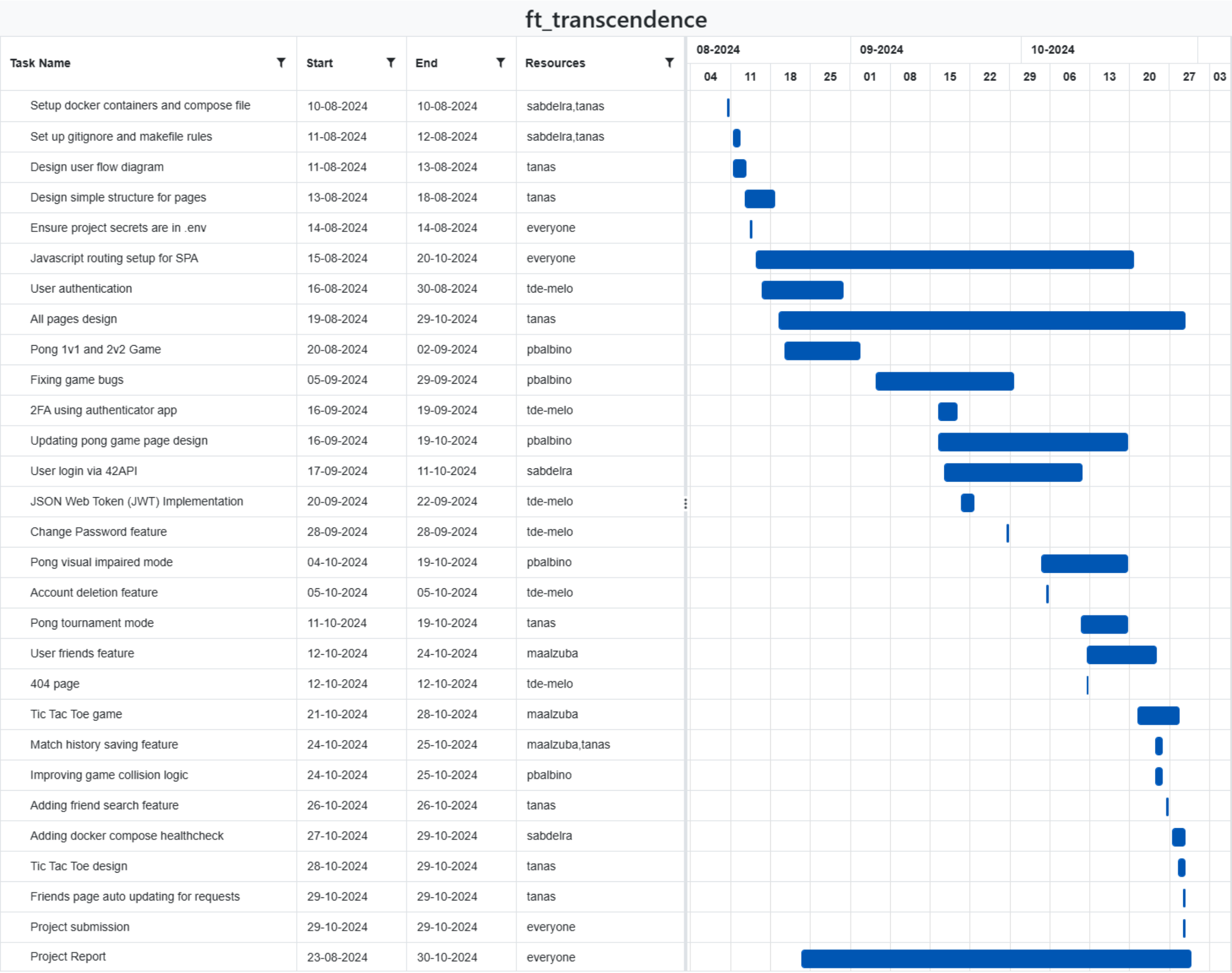
- **Initial Phase:** The user management system was set up with basic functionalities, such as secure registration and login processes. Users could create accounts, set display names, and update their profiles.
- **Intermediate Phase:** Additional features were added, including the ability to manage friend lists, view online status, and delete their account. Security measures were strengthened with the introduction of Two-Factor Authentication (2FA).
- **Current Phase:** The platform now provides a comprehensive user management solution that supports secure website subscriptions, profile updates, and tournament participation. Users can upload avatars, track their gaming stats, and access match history. The authentication system adheres to best practices, employing JWTs for session management and ensuring compliance with secure authentication standards.

Chapter 3

Elements of SDLC

3.1 Gantt Chart

The Gantt Chart delineates the project timeline, outlining tasks, dependencies, and milestones, thereby mapping the project's progression through each phase of the Software Development Life Cycle (SDLC), as follows:



ft_transcendence

Task Name	Start	End	Resources
Setup docker containers and compose file	10-08-2024	10-08-2024	sabdelra,tanas
Set up gitignore and makefile rules	11-08-2024	12-08-2024	sabdelra,tanas
Design user flow diagram	11-08-2024	13-08-2024	tanas
Design simple structure for pages	13-08-2024	18-08-2024	tanas
Ensure project secrets are in .env	14-08-2024	14-08-2024	everyone
Javascript routing setup for SPA	15-08-2024	20-10-2024	everyone
User authentication	16-08-2024	30-08-2024	tde-melo
All pages design	19-08-2024	29-10-2024	tanas
Pong 1v1 and 2v2 Game	20-08-2024	02-09-2024	pbalbino
Fixing game bugs	05-09-2024	29-09-2024	pbalbino
2FA using authenticator app	16-09-2024	19-09-2024	tde-melo
Updating pong game page design	16-09-2024	19-10-2024	pbalbino
User login via 42API	17-09-2024	11-10-2024	sabdelra
JSON Web Token (JWT) Implementation	20-09-2024	22-09-2024	tde-melo
Change Password feature	28-09-2024	28-09-2024	tde-melo
Pong visual impaired mode	04-10-2024	19-10-2024	pbalbino
Account deletion feature	05-10-2024	05-10-2024	tde-melo
Pong tournament mode	11-10-2024	19-10-2024	tanas
User friends feature	12-10-2024	24-10-2024	maalzuba
404 page	12-10-2024	12-10-2024	tde-melo
Tic Tac Toe game	21-10-2024	28-10-2024	maalzuba
Match history saving feature	24-10-2024	25-10-2024	maalzuba,tanas
Improving game collision logic	24-10-2024	25-10-2024	pbalbino
Adding friend search feature	26-10-2024	26-10-2024	tanas
Adding docker compose healthcheck	27-10-2024	29-10-2024	sabdelra
Tic Tac Toe design	28-10-2024	29-10-2024	tanas
Friends page auto updating for requests	29-10-2024	29-10-2024	tanas
Project submission	29-10-2024	29-10-2024	everyone
Project Report	23-08-2024	30-10-2024	everyone

FIGURE 3.1: Gantt Chart

This Gantt chart outlines the project timeline for **ft_transcendence**, showing tasks, their durations, and assigned resources. It spans from August to October 2024, covering development milestones like setting up the Docker environment, designing user interfaces, and implementing essential features such as user authentication, JWT, and 2FA. The project includes building the main pong game modes (1v1 and 2v2), adding game customization options and accessibility options, and implementing user features like friend management, match history, and account settings. Key phases involve testing, debugging, and updating, with the planned submission date due by October 29, 2024. Various team members are responsible for specific tasks, ensuring collaborative progress throughout the timeline.

3.2 Risk Register and Risk Matrix

The risk register table was utilized to identify potential risks early in the project, prioritizing them according to their impact and likelihood. Mitigation strategies were developed to minimize or prevent the risks from impacting the project.

				Impact		
	Risk Matrix	**Insignificant 1**	**Minor 2**	**Moderate 3**	**Major 4**	**Severe 5**
Likelihood	**Almost Certain 5**	Medium 5	High 10	Very High 15	Extreme 20	Extreme 25
	Likely 4	Medium 4	Medium 8	High 12	Very High 16	Extreme 20
	Possible 3	Low 3	Medium 6	Medium 9	High 12	Very High 15
	Unlikely 2	Very Low 2	Low 4	Medium 6	Medium 8	High 10
	Rare 1	Very Low 1	Very Low 2	Low 3	Medium 4	Medium 5

TABLE 3.1: Risk Matrix

The status of each risk was monitored continuously throughout the project's duration to ensure timely responses. To facilitate the analysis and mitigation process, the risks were divided between the *Frontend Team* and the *Backend Team*, considering the project's scope and the number of developers involved. In a real-world project at a well-established company, it would be preferable to allocate risks across additional specialized teams, such as the Security Team, DevOps, and Project Management, etc., to ensure a more comprehensive approach to risk management.

Risk Description	**Impact**	**Likelihood**	**Severity**	**Risk Owner**	**Mitigation Strategy**
	1- Low	5- High			
Security vulnerabilities: Issues in 2FA, JWT, or login mechanisms could lead to unauthorized access.	4	1	4	Backend Team	Perform regular security checks, implement robust encryption, and validate JWT tokens properly.
Data loss or corruption: PostgreSQL database could face corruption or unintentional data loss.	3	1	3	Backend Team	User has a local backup.
Cross-browser compatibility issues: Inconsistent behavior across different browsers.	1	1	1	Frontend Team	Perform cross-browser testing, adopt standardized web technologies, and check if the tool/function that is being implemented is available on multiple browsers.
Frontend responsiveness issues: Problems with the Bootstrap-based design may lead to poor user experience on various devices.	3	2	6	Frontend Team	Test across different screen sizes and use media queries (CSS technique used to apply different styles to a web page based on the characteristics of the user's device) for fine-tuning the design responsiveness.
Docker container misconfigurations: Incorrect configurations may cause services to malfunction.	5	3	15	Backend Team	Implement configuration validation, use environment variables properly, and automate testing in CI/CD.
Accessibility compliance issues: Features for visually impaired users may not meet required standards.	3	3	9	Frontend Team	Conduct accessibility testing, get user feedback, and analyze researches.
SSR implementation issues: Errors in Server-Side Rendering could affect the performance.	3	2	6	Backend Team	Use frameworks that support SSR well, handle edge cases in rendering, and test the rendered output.
Third-party service dependency issues: Problems with external services used (e.g., authentication, APIs).	4	3	12	Backend Team	Implement fallback mechanisms, monitor third-party service statuses, and plan for alternative providers. By implementing the login using 42login the system will rely on a third-party.
User data privacy concerns: Inadequate handling of user data could lead to privacy violations.	1	1	1	Backend Team	All sensitive data is encrypted and the data stored is minimal and most likely already available online (such as name and surname, github and X profile).

TABLE 3.2: Risk Register

3.3 Functional Requirements

User Authentication and Management

- Secure registration, login, and profile management.
- Support for multiple authentication mechanisms (username login, 42 login, and 2FA).
- User profile management features, including unique display names, avatar uploads, and online status visibility.

Gameplay and Game Modes

- Multiplayer game modes, supporting 2 or 4 players in Pong and tournament.
- Additional game option, like Tic-Tac-Toe, to diversify gameplay.
- Customization options for game settings, including ball and paddle speed, score limit, sound controls, and different difficulty levels.

Match History

- Tracking and storing gameplay statistics, including wins, losses, and match history.

Accessibility Features

- High-contrast colors throughout the website, keyboard navigation, and large icons for visually impaired users while playing Pong.

3.4 Technical Requirements

Software Specifications

Operating System

- **Server**: A Linux-based operating system (e.g., Ubuntu Server, CentOS, or Debian) for its stability, scalability, and compatibility with Docker, PostgreSQL, and Django.
- **Client**: Cross-platform compatibility, supporting Windows, macOS, and mobile platforms (iOS/Android) via responsive web design.

Programming Language and Frameworks

- **Backend**: Python with Django framework for fast development, scalability, and built-in security features.
- **Frontend**: Pure vanilla JavaScript and Bootstrap frameworks for a dynamic and interactive user interface.
- **Rendering**: Nginx for static file serving and SSR optimization.

Database

- PostgreSQL for its advanced relational database capabilities, scalability, and support for transactional integrity.

Authentication and Security

- JSON Web Tokens (JWT) for secure token-based authentication.
- Two-Factor Authentication (2FA) to enhance login security.
- TLS for encrypted data transmission.

Hardware Specifications

Server Requirements

• Processor: Multi-core CPU (e.g., Intel Xeon or AMD EPYC) to handle concurrent requests efficiently. • Memory: At least 16 GB RAM, scalable based on concurrent user load (e.g., up to 64 GB for high traffic). • Storage: SSDs for fast read/write operations (minimum 500 GB, scalable depending on database size). • Network: High-speed internet connectivity (1 Gbps or higher).

Client Requirements

• Device: Desktop, laptop, tablet, or smartphone with: 1. Processor: Modern multi-core CPU (Intel i5/AMD Ryzen 5 or higher). 2. Memory: Minimum 4 GB RAM (8 GB recommended for optimal performance). 3. Browser: Latest versions of browsers (Chrome and Firefox). 4. Display: HD resolution or higher (1920x1080 recommended).

System Architecture

Client Layer (Frontend)

The frontend layer is the client-facing part of the system, responsible for user interaction, presentation, and providing a responsive and intuitive user experience. Technologies:

- Styling: CSS3 responsiveness and design.
- Rendering: Server-Side Rendering (SSR) for faster page load times.
- Authentication: Integration of JWT (JSON Web Tokens) and 2FA mechanisms for secure authentication.
- API Consumption: AJAX, Fetch API, or Axios to make HTTP requests to the backend API.

Application Layer (Backend)

The backend serves as the business logic and API layer, handling data processing, authentication, game mechanics, and communication between the frontend and the database. Technologies:

- Web Framework: Django for Python, chosen for its robust features and security practices.
- Game Logic: Core game logic for handling match states, scoring, difficulty settings, etc., implemented in JavaScript.
- Authentication: JWT-based stateless authentication and 2FA, ensuring secure user login and session management.
- Security: Features like input validation to protect against SQL injection and XSS attacks, leveraging Django's built-in security tools.

Data Layer (Database)

The data layer handles persistent data storage, ensuring consistency and integrity across different system components. Technologies:

- Database: PostgreSQL for relational data storage, chosen for its ACID compliance, scalability, and robust querying capabilities.
- Data Models: The database schema would include tables for users, matches, gameplay history, for example.

- Data Integrity: Enforcing constraints, relationships, and foreign keys to maintain referential integrity between tables.

Infrastructure Layer

The infrastructure layer is responsible for the deployment, scaling, and management of the application. This layer includes the servers, cloud services, and containerization that support the system. Technologies:

- Containerization: Docker for packaging the application into containers, ensuring consistency across different environments (development, staging, production).
- Load Balancing: Nginx for load balancing HTTP requests across backend servers, ensuring efficient handling of high traffic.
- Web Server: Nginx for serving static files (such as images, CSS, and JS files) and acting as a reverse proxy to the backend API.

Communication and Networking

External Communication:

- TLS/SSL Encryption: Secure data transmission between the client and the server, ensuring data protection during login and gameplay.
- API Security: OAuth or other token-based methods to authenticate API calls from external services

3.5 Non-functional requirements

Performance and Scalability

- **Fast Operations:** The use of Nginx to optimize static file serving and implement Server-Side Rendering (SSR) ensures fast loading times and improves performance. These techniques minimize server load and enhance responsiveness.
- **Load Balancing and Optimization:** By combining server optimizations with backend technologies, the system ensures consistent response times even during peak usage periods, enabling a high degree of operational reliability.

- **Scalable System Architecture:** The system employs a robust architecture through the integration of PostgreSQL as the database management system and the Django framework. This combination enables the platform to accommodate increasing user demands without significant performance degradation, thereby supporting horizontal and vertical scaling.
- **Infrastructure Scalability:** The use of Docker containers provides a modular infrastructure that supports independent deployment of services, such as frontend, backend, and game mechanics. This containerized approach ensures that services can be scaled independently based on demand, optimizing resource utilization.

Security

- **Robust Authentication Mechanisms:** The implementation of secure login methods, including JSON Web Tokens (JWT) and Two-Factor Authentication (2FA), fortifies the authentication process, reducing risks of unauthorized access.
- **Vulnerability Mitigation:** The adoption of Django's built-in security features protects against common threats such as SQL injection and cross-site scripting (XSS) attacks. These safeguards are critical for maintaining a secure application environment.
- **Data Encryption and Protection:** The use of Transport Layer Security (TLS) ensures secure data transmission, safeguarding sensitive information such as login credentials and gameplay data. Furthermore, secure storage techniques underline a strong commitment to data privacy and security best practices.

Reliability and Data Integrity

- **Consistent Data Handling:** The system ensures the accurate storage and retrieval of critical data, such as game history and match records. PostgreSQL's transactional capabilities contribute to data reliability by ensuring consistency and preventing data corruption.
- **Seamless Integration:** The synergy between PostgreSQL and Django facilitates efficient database operations, enabling seamless data access and updates that enhance system reliability under diverse operational conditions.

User Experience (UX)

- **Device-Responsive Design:** The platform's responsive design ensures a seamless user experience across various devices, including desktops, tablets, and smartphones. This consistency in interaction enhances accessibility and usability.

- **Intuitive Interface:** An emphasis on smooth navigation, real-time responsiveness during gameplay, and a clear layout fosters an engaging user interface that minimizes cognitive load and supports user satisfaction.
- **Inclusive Gaming Experience:** The system complies with accessibility standards, ensuring usability for visually impaired users. This inclusivity expands the user base and aligns the platform with ethical design principles.
- **Continuous Refinement:** Ongoing usability testing and iterative feedback loops enable the refinement of UX, ensuring the system evolves to meet user needs effectively.

Compliance

- **Alignment with Cybersecurity Standards:** The implementation of TLS for secure data transmission reflects adherence to contemporary cybersecurity practices, ensuring confidentiality and integrity of user data.
- **Implicit Compliance:** While specific legal or industry standards are not explicitly mentioned, the system's strong emphasis on security and accessibility aligns with general compliance principles.
- **Accessibility Standards:** Adherence to recognized accessibility guidelines underscores the platform's commitment to inclusive design, further enhancing its compliance profile.

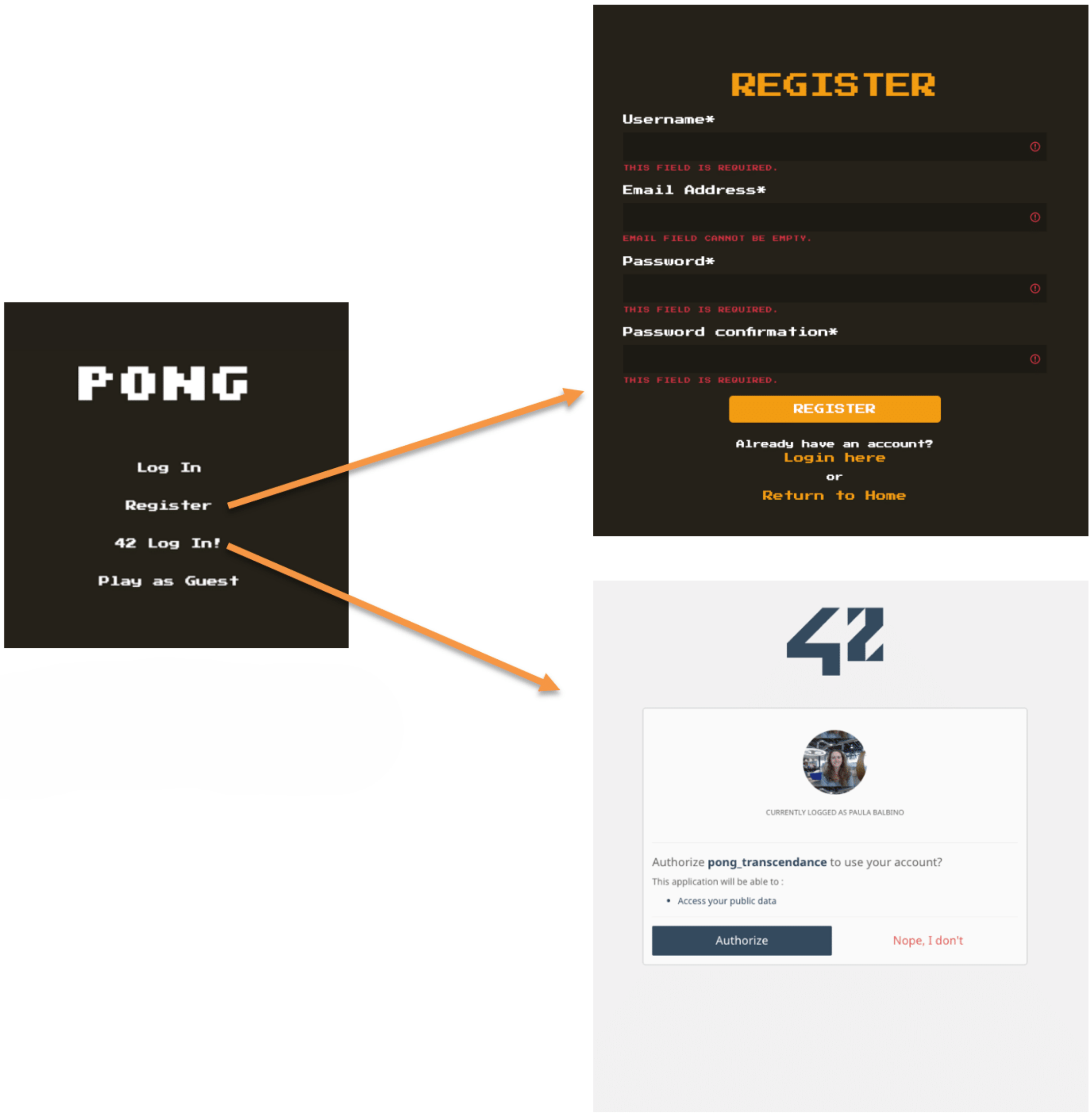
Inclusivity

- Compliance with accessibility standards for visually impaired users, supporting inclusive gaming experiences.
- Ongoing usability testing to refine UX, based on feedback.

3.6 Wireframes

Wireframes provide a visual outline of the project's user interface structure and play a key role in mapping the user journey. They enable early visualization of layout and user interactions, offering insight into user navigation within the system well before the detailed design and coding phases.
In this project, wireframes and key website screenshots illustrate the user journey. Initial screen allows the user to log in (option for registered users), registration with username or through 42 login as well as the option the play as guest. To register the user must provide a unique username,

valid email and a password, basic checks will be performed. Otherwise, the user can login using 42 account, and the respective authorization will be required, as follows:



PONG

Log In

Register

42 Log In!

Play as Guest

REGISTER

Username*

THIS FIELD IS REQUIRED.

Email Address*

EMAIL FIELD CANNOT BE EMPTY.

Password*

THIS FIELD IS REQUIRED.

Password confirmation*

THIS FIELD IS REQUIRED.

REGISTER

Already have an account?
Login here
or
Return to Home

42

CURRENTLY LOGGED AS PAULA BALBINO

Authorize **pong_transcendance** to use your account?

This application will be able to :

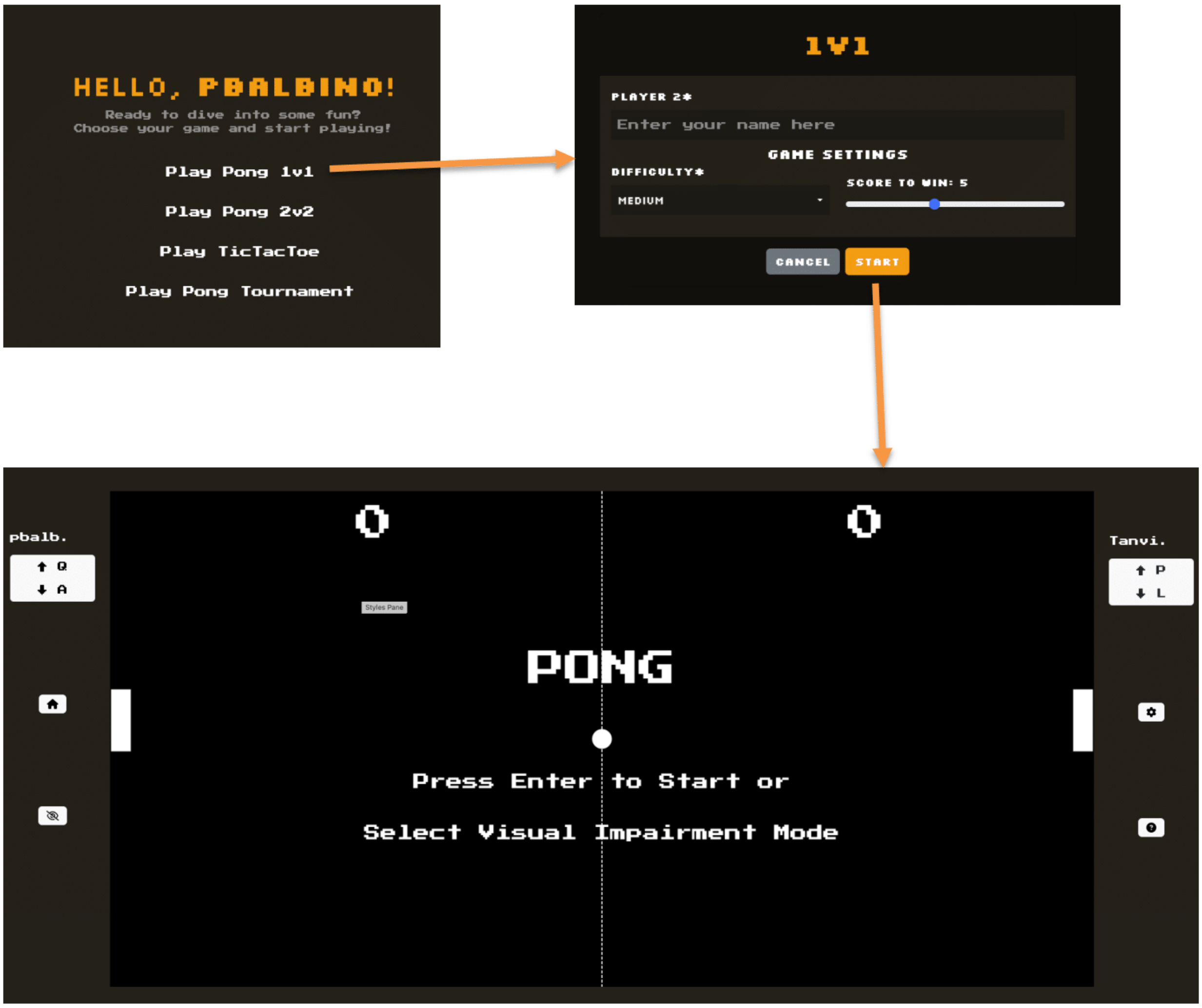
- Access your public data

Authorize

Nope, I don't

Figure 3.2: Initial screen and registration

If the authorization is granted, the user is now logged on the website, the same screen will be available for the user. Now the user can choose between several options: play individual or pair matches, pong tournament or the additional game, Tic Tac Toe. It is also possible to view the profile and add friends (these functionalities will be shown after the game). It will be necessary to add the opponents name, choose a difficulty level (medium by default) and set the score. After adding opponent's name and clicking on "start", the Pong game will be displayed. Player's names are shown close the control buttons, indicating which keyboard keys they should use to move the paddle up and down. Additionally, on the left there are two options, the "home" button return to home page and the eye slashed button represents the visual impairment mode. On the right, the gear button opens the game settings and the question mark, the rules of the game. All these options will be demonstrated on the next pictures.



HELLO, PBALBINO!

Ready to dive into some fun?
Choose your game and start playing!

PLAYER 2*: Enter your name here

GAME SETTINGS

DIFFICULTY*: MEDIUM

SCORE TO WIN: 5

Press Enter to Start or

Select Visual Impairment Mode

FIGURE 3.3: Selecting a game

If the user decides to play the visual impairment mode, colors will be on high contrast, buttons, ball and paddles will be adapted as well:

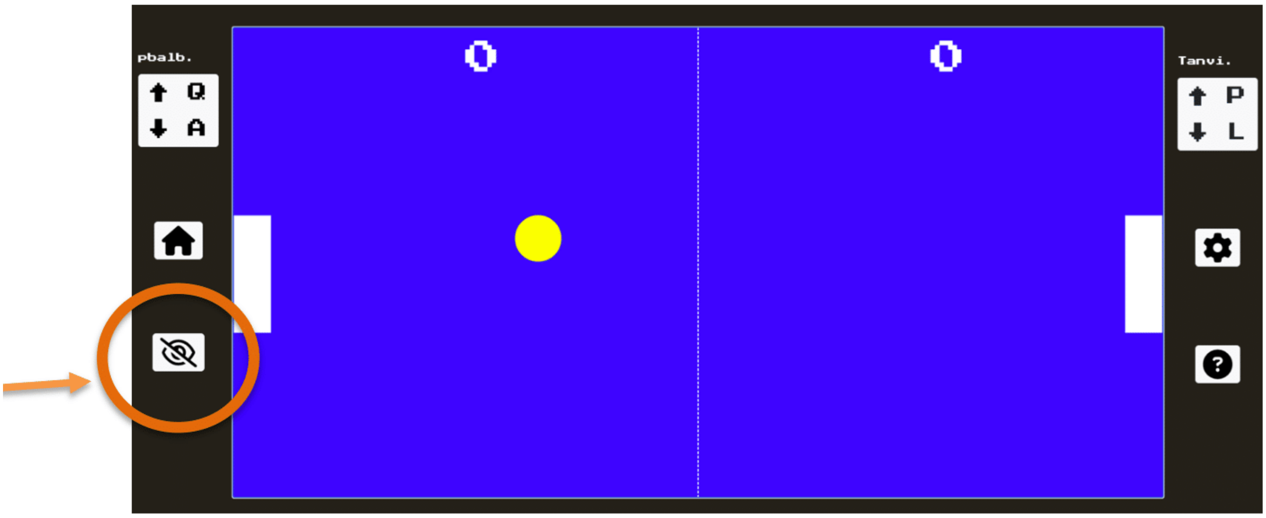


Figure 3.4: Visual Impairment Mode

As soon as the set score is reached, game ends , the winner is show and there are to possible options, go back to home page or play again:

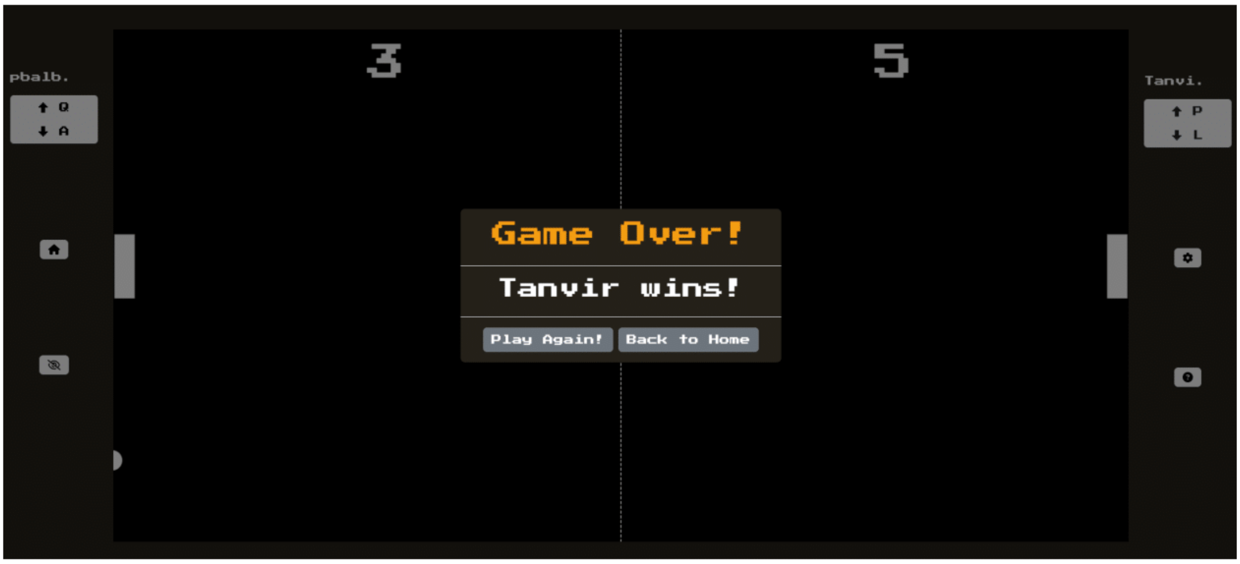
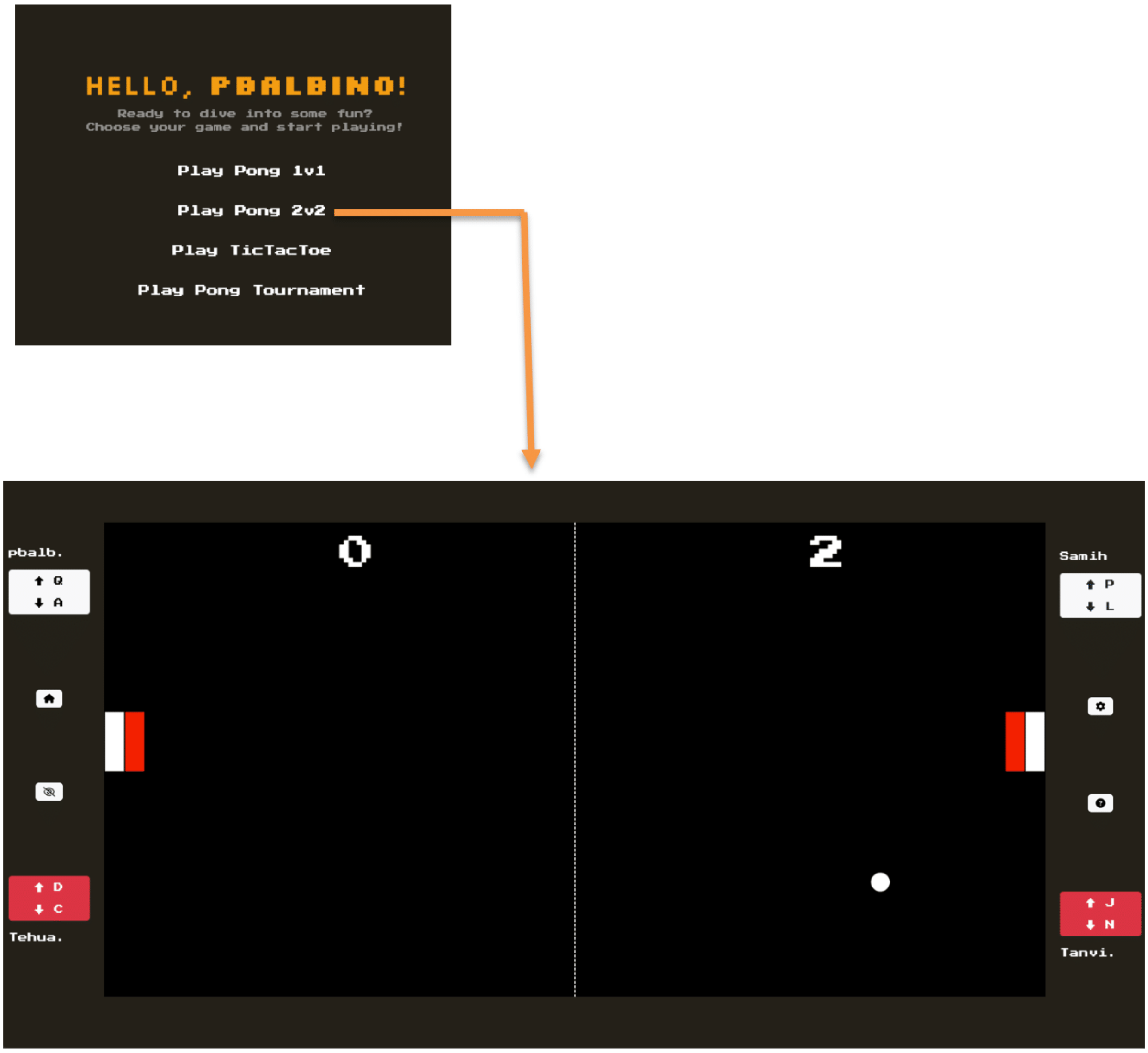


Figure 3.5: 1x1 Pong Match - Gameover



HELLO, PBALBINO!

Ready to dive into some fun?
Choose your game and start playing!

Figure 3.6: 1x1 Pong Match - Gameover

Regarding the possible configurations, user can choose between pre established levels (easy, medium or hard) as well as customize it. The sound can be set off as well:

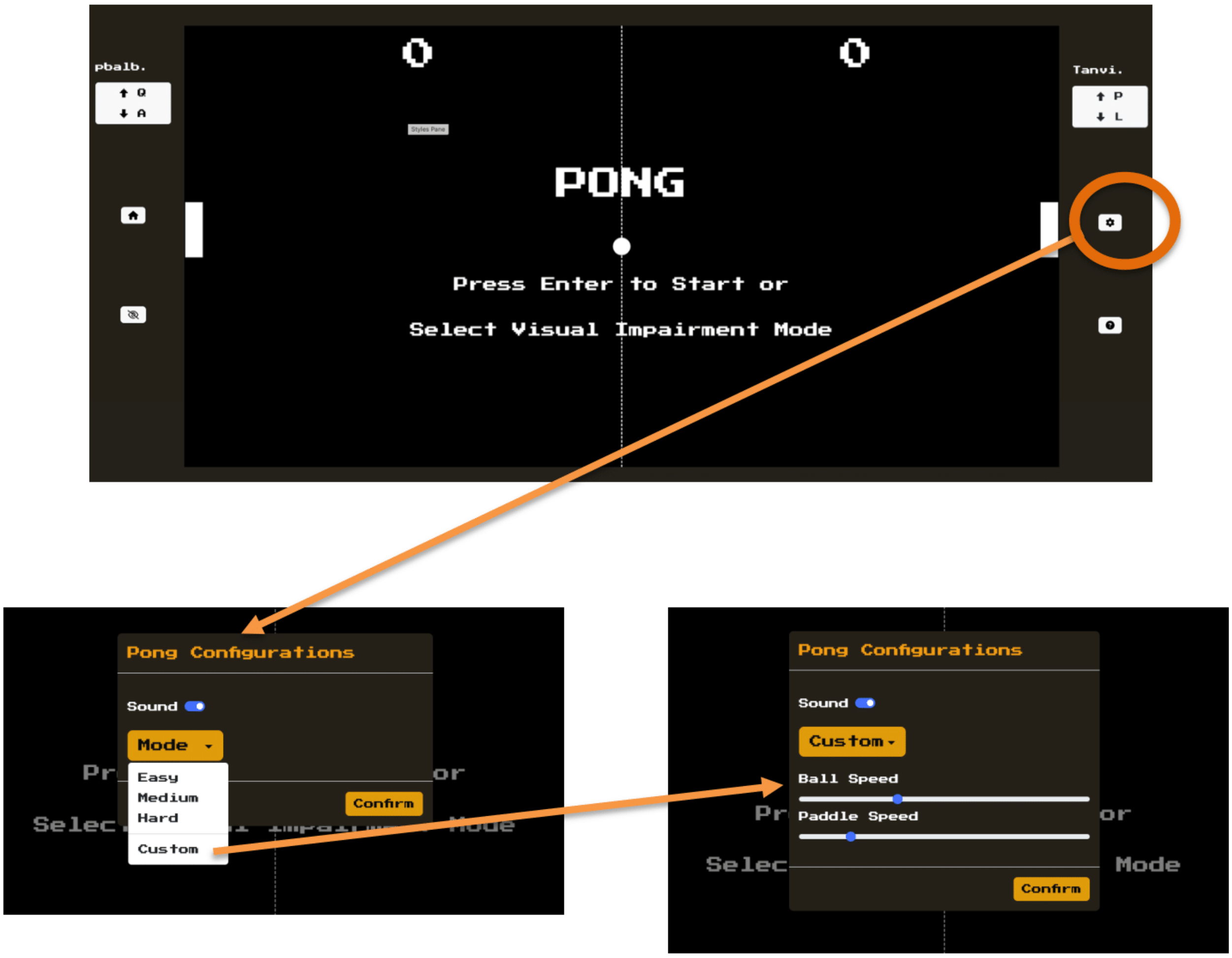
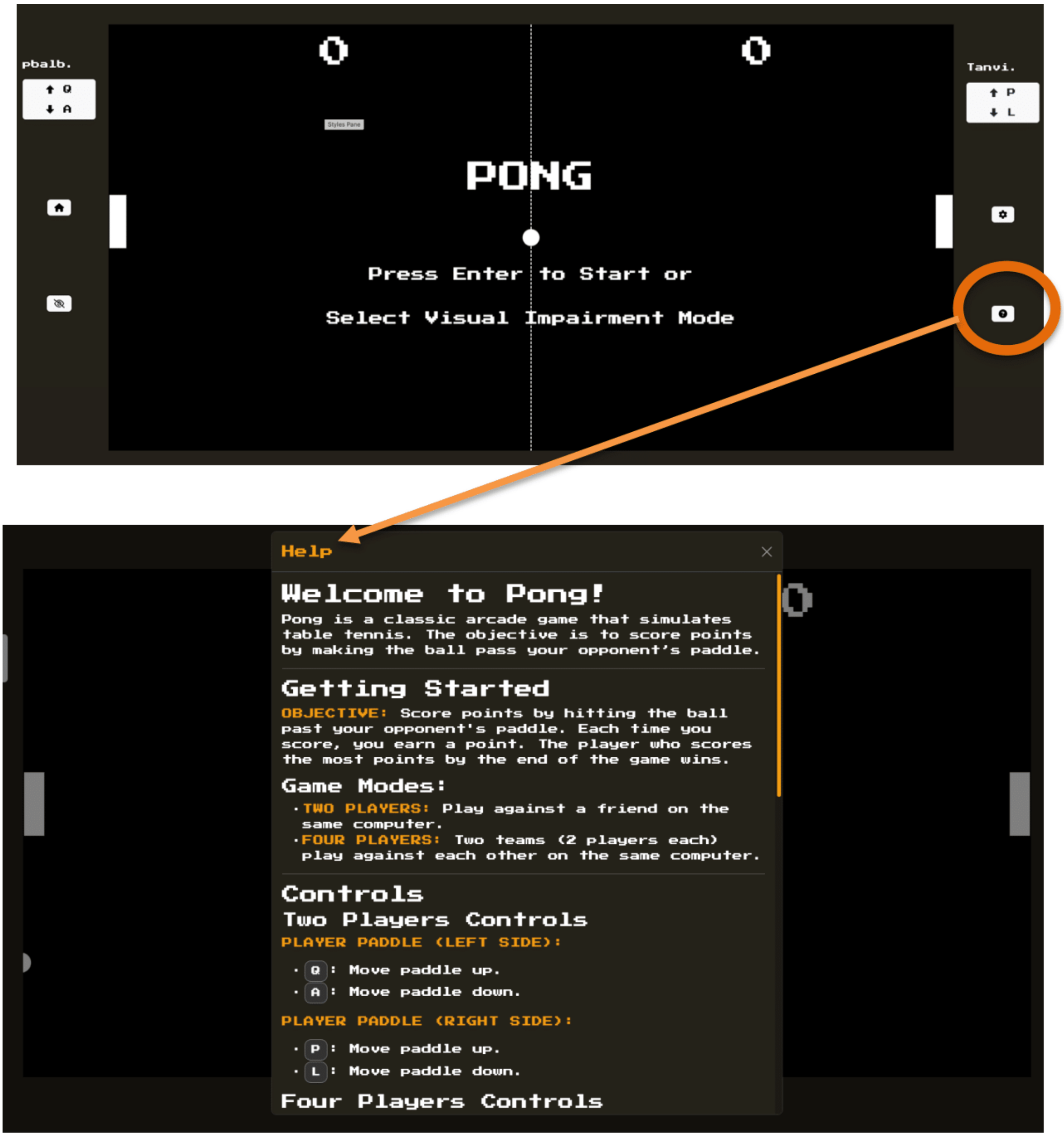


Figure 3.7: Pong Game Configurations

Additionally, for game rules the following pop up is displayed:



Press Enter to Start or

Select Visual Impairment Mode

Help

Welcome to Pong!

Pong is a classic arcade game that simulates table tennis. The objective is to score points by making the ball pass your opponent's paddle.

Getting Started

OBJECTIVE: Score points by hitting the ball past your opponent's paddle. Each time you score, you earn a point. The player who scores the most points by the end of the game wins.

Game Modes:

- TWO PLAYERS: Play against a friend on the same computer.
- FOUR PLAYERS: Two teams (2 players each) play against each other on the same computer.

Controls

Two Players Controls

PLAYER PADDLE (LEFT SIDE):

- Q: Move paddle up.
- A: Move paddle down.

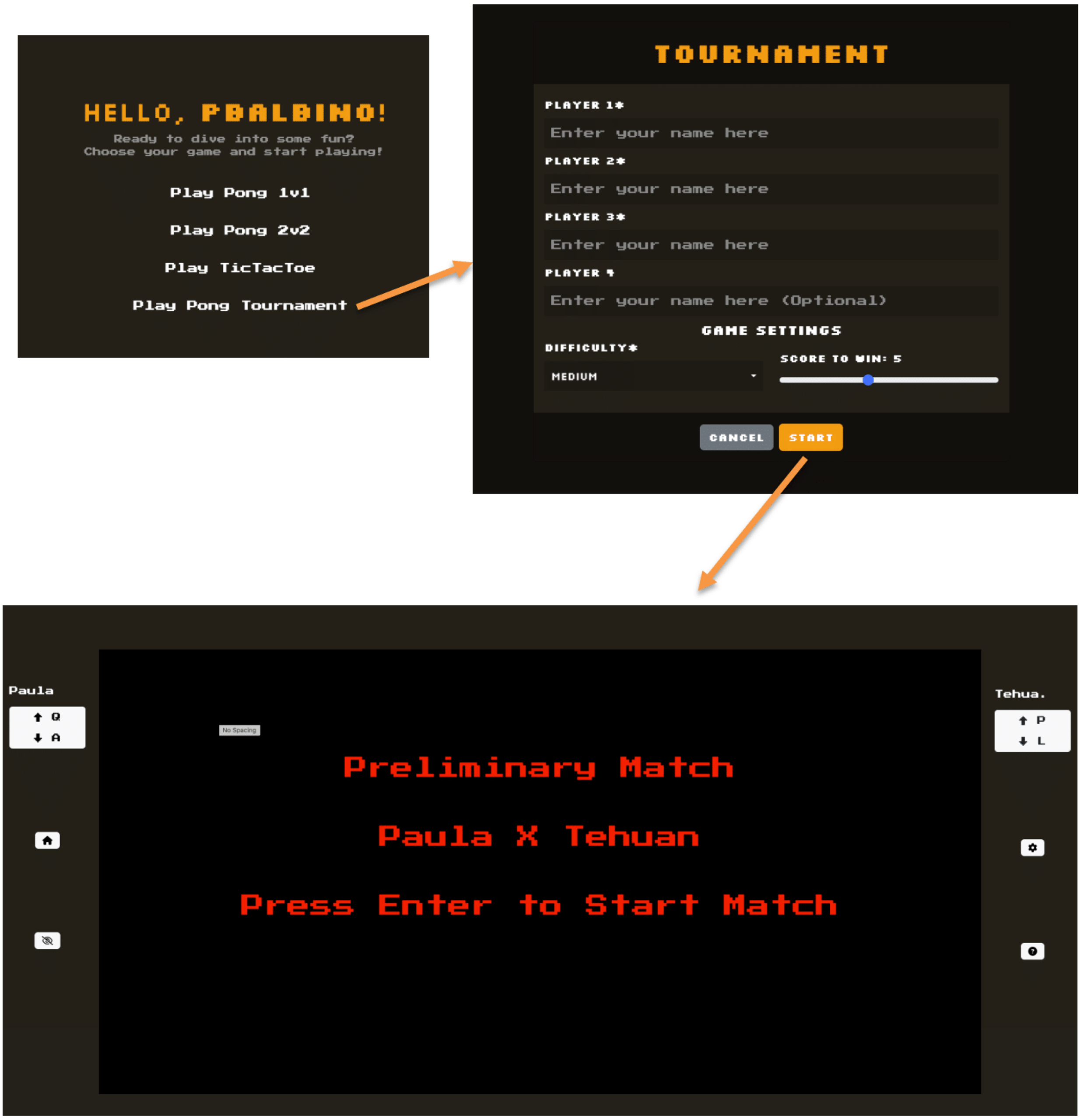
PLAYER PADDLE (RIGHT SIDE):

- P: Move paddle up.
- L: Move paddle down.

Four Players Controls

Figure 3.8: Pong Game Rules

It is possible to set a Pong tournament. Initial step is to enter the player's names and game settings. Player's names are shuffled, and the screen shows the first game. The same happens for the other preliminary match and the final match will be between the winners of the previous matches:



HELLO, PBALBINO!

Ready to dive into some fun?
Choose your game and start playing!

TOURNAMENT

PLAYER 1*
Enter your name here
PLAYER 2*
Enter your name here
PLAYER 3*
Enter your name here
PLAYER 4
Enter your name here (Optional)

GAME SETTINGS

DIFFICULTY*: MEDIUM
SCORE TO WIN: 5

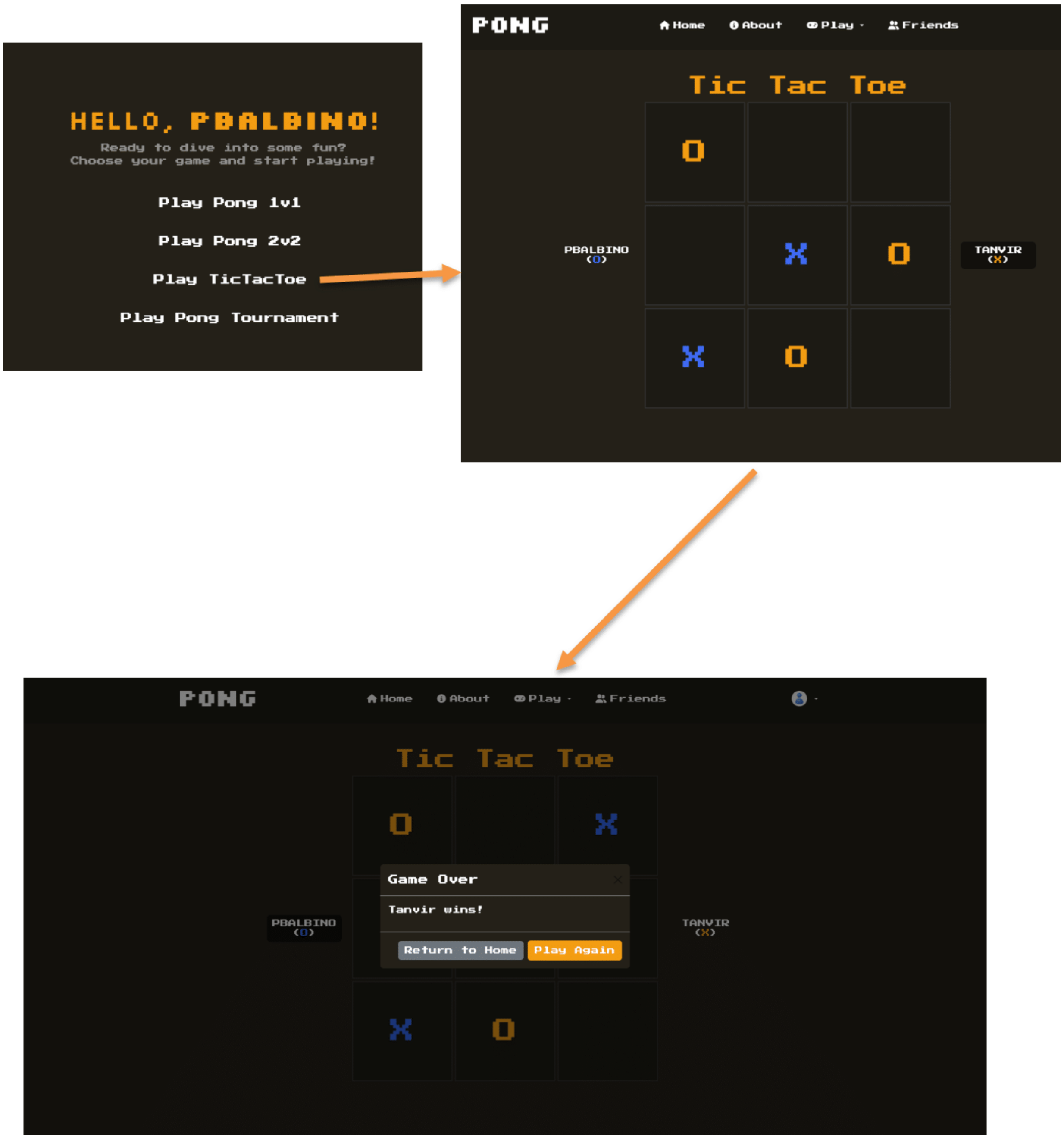
Preliminary Match

Paula X Tehuan

Press Enter to Start Match

Figure 3.9: Pong tournament

In addition to pong, user can also play Tic Tac Toe. Initial screen required the opponent's name (similarly of pong 1x1 match) and after clicking on "start" the game is displayed:


HELLO, PBALBINO!

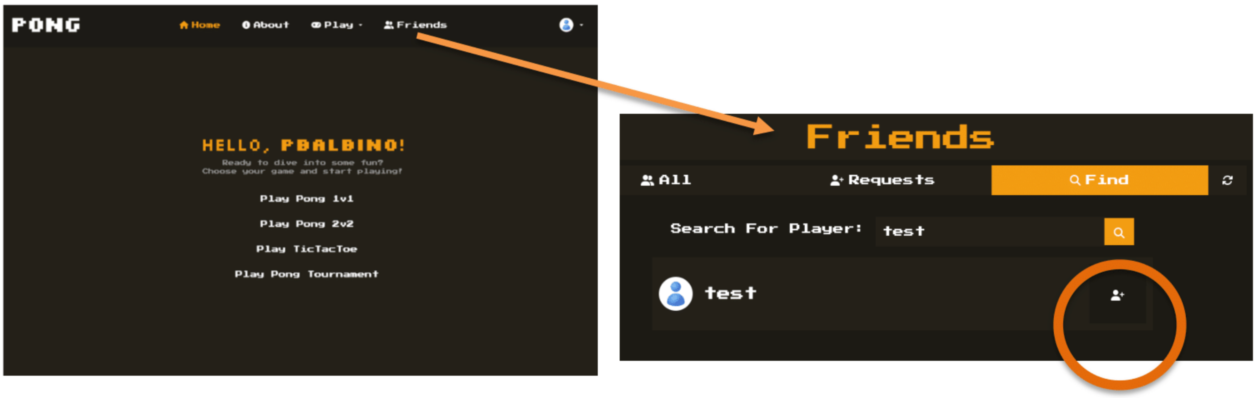
Ready to dive into some fun?
Choose your game and start playing!

Game Over

Tanvir wins!

Figure 3.10: Tic Tac Toe Game

Back to the home page, it is possible to add a friend by clicking on "Friends". This will open the following screen on which it is necessary to add the friend username, for illustration purposes, "test" will be added as a friend. If the user tried to add a non-existing user, the proper error message will be displayed.

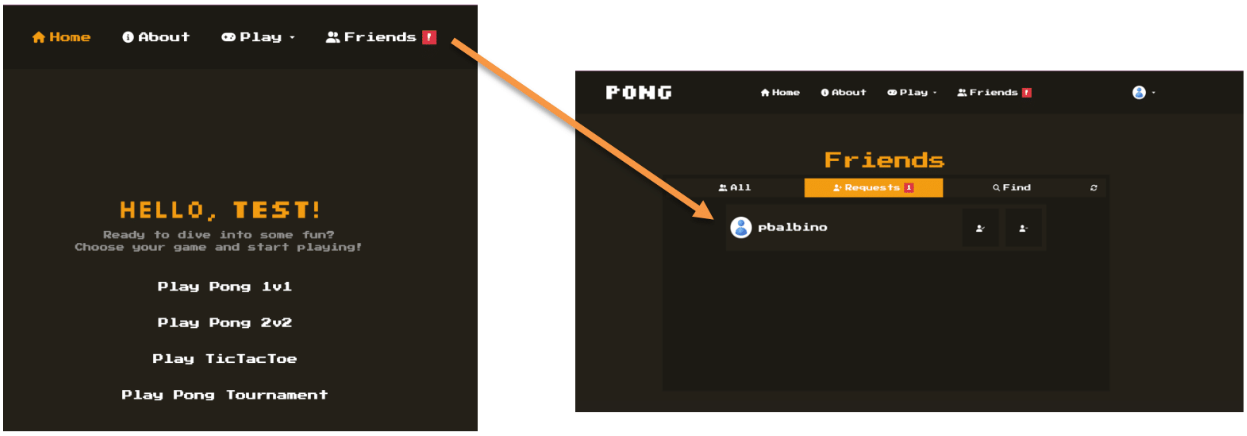


HELLO, PBALBINO!
Ready to dive into some fun?
Choose your game and start playing!

Search For Player: test

FIGURE 3.11: Friends Page

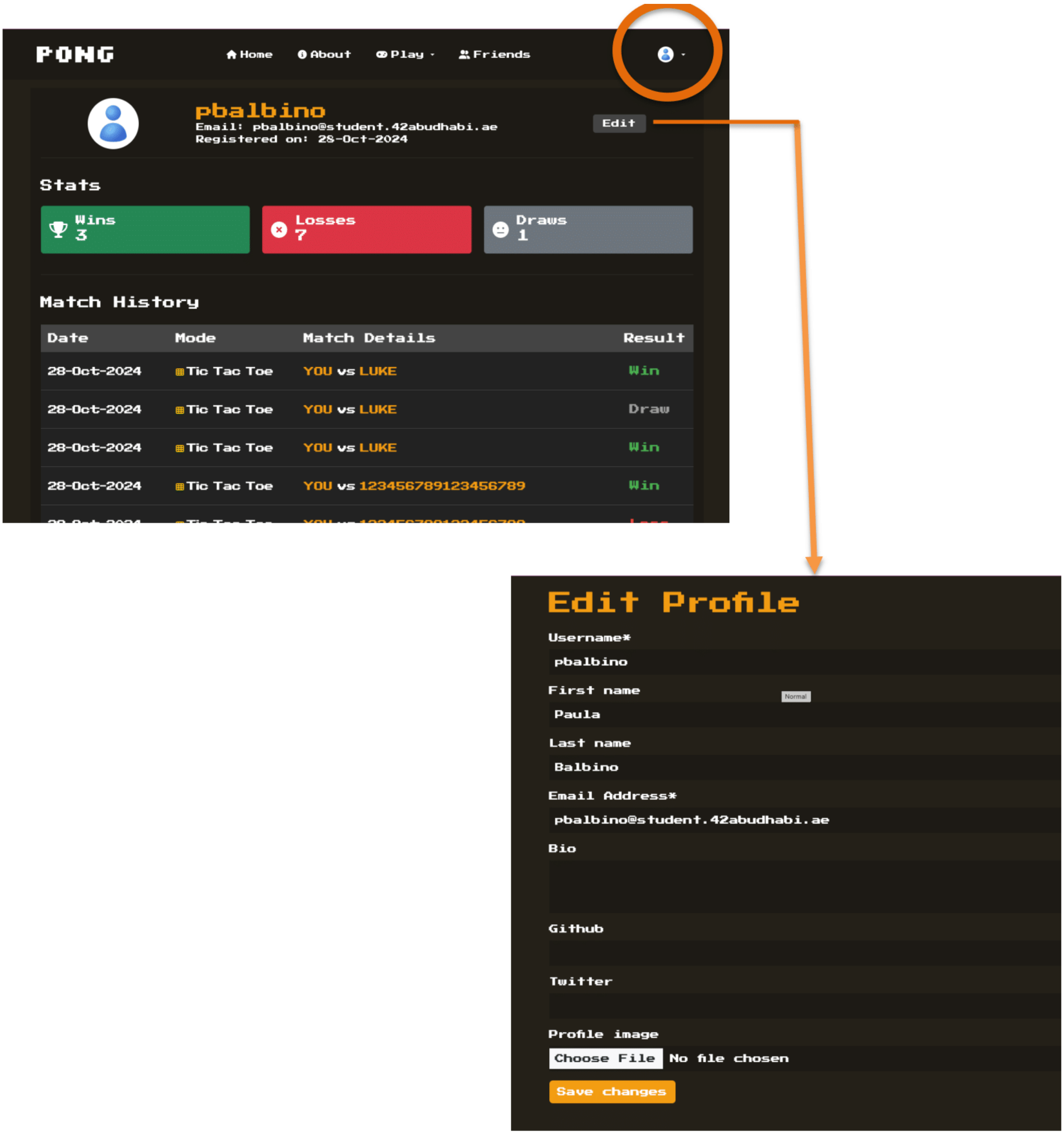
When the user "test" logs in, there will be a new friend request on his home page:



HELLO, TEST!
Ready to dive into some fun?
Choose your game and start playing!

FIGURE 3.12: Friends request and approval

Another functionality this application has is the match history, by clicking on the profile icon (top right), the user can check the wins, losses and draws, match dates and results, moreover, by clicking on edit the user can change the profile information:



pbalbino
Email: pbalbino@student.42abudhabi.ae
Registered on: 28-Oct-2024

Stats

Wins 3 | Losses 7 | Draws 1

Match History

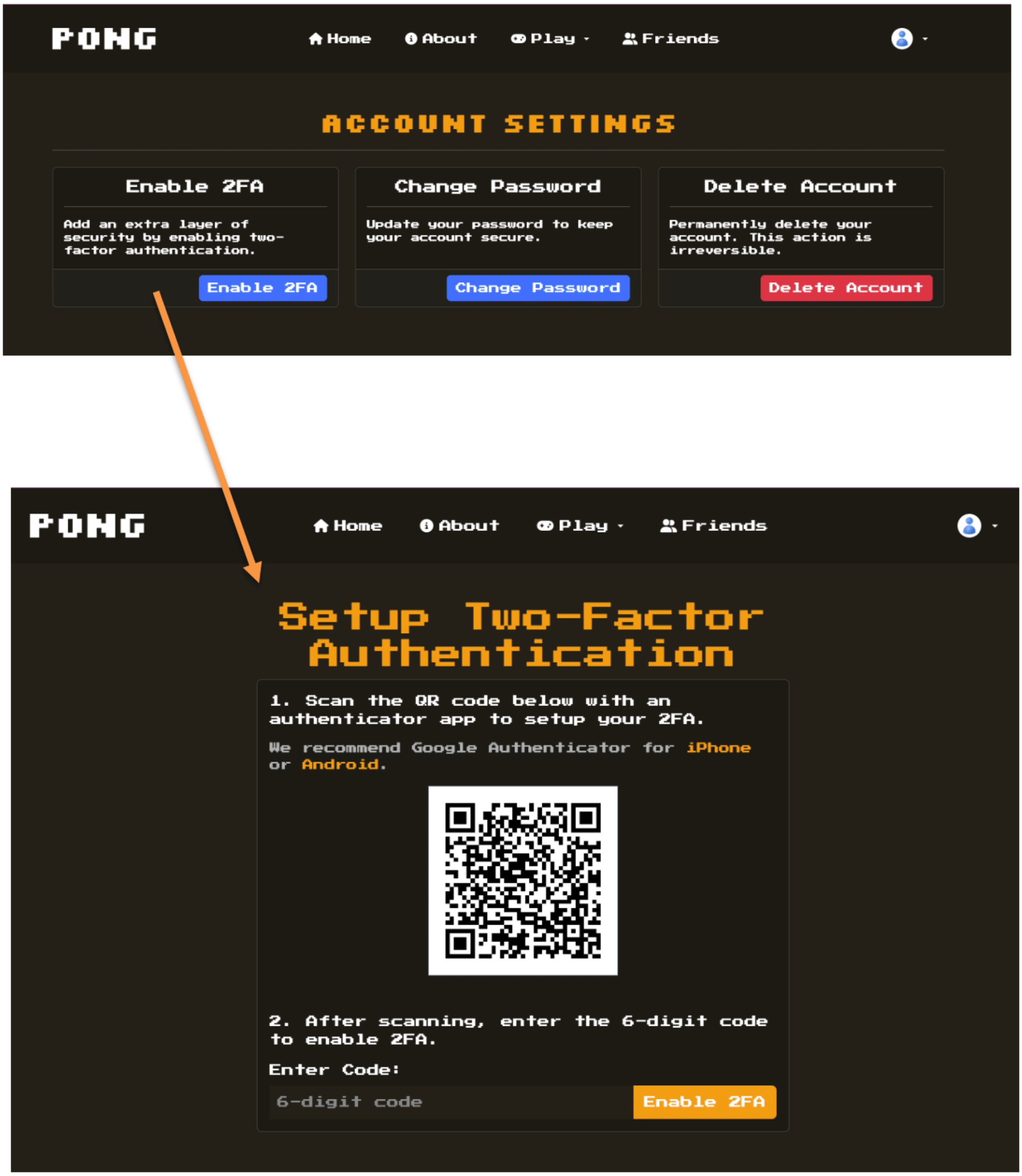
Date	Mode	Match Details	Result
28-Oct-2024	Tic Tac Toe	YOU vs LUKE	Win
28-Oct-2024	Tic Tac Toe	YOU vs LUKE	Draw
28-Oct-2024	Tic Tac Toe	YOU vs LUKE	Win
28-Oct-2024	Tic Tac Toe	YOU vs 123456789123456789	Win

Edit Profile

Username*: pbalbino
First name: Paula
Last name: Balbino
Email Address*: pbalbino@student.42abudhabi.ae
Bio:
Github:
Twitter:
Profile image: Choose File No file chosen

FIGURE 3.13: Profile screen and Edit

Regarding account settings, users can enable/disable the 2FA, change the password or delete the account. To set up the 2FA its necessary to scan the QR code using Google Authenticator for example:



ACCOUNT SETTINGS

Enable 2FA

Add an extra layer of security by enabling two-factor authentication.

Change Password

Update your password to keep your account secure.

Delete Account

Permanently delete your account. This action is irreversible.

Setup Two-Factor Authentication

1. Scan the QR code below with an authenticator app to setup your 2FA.

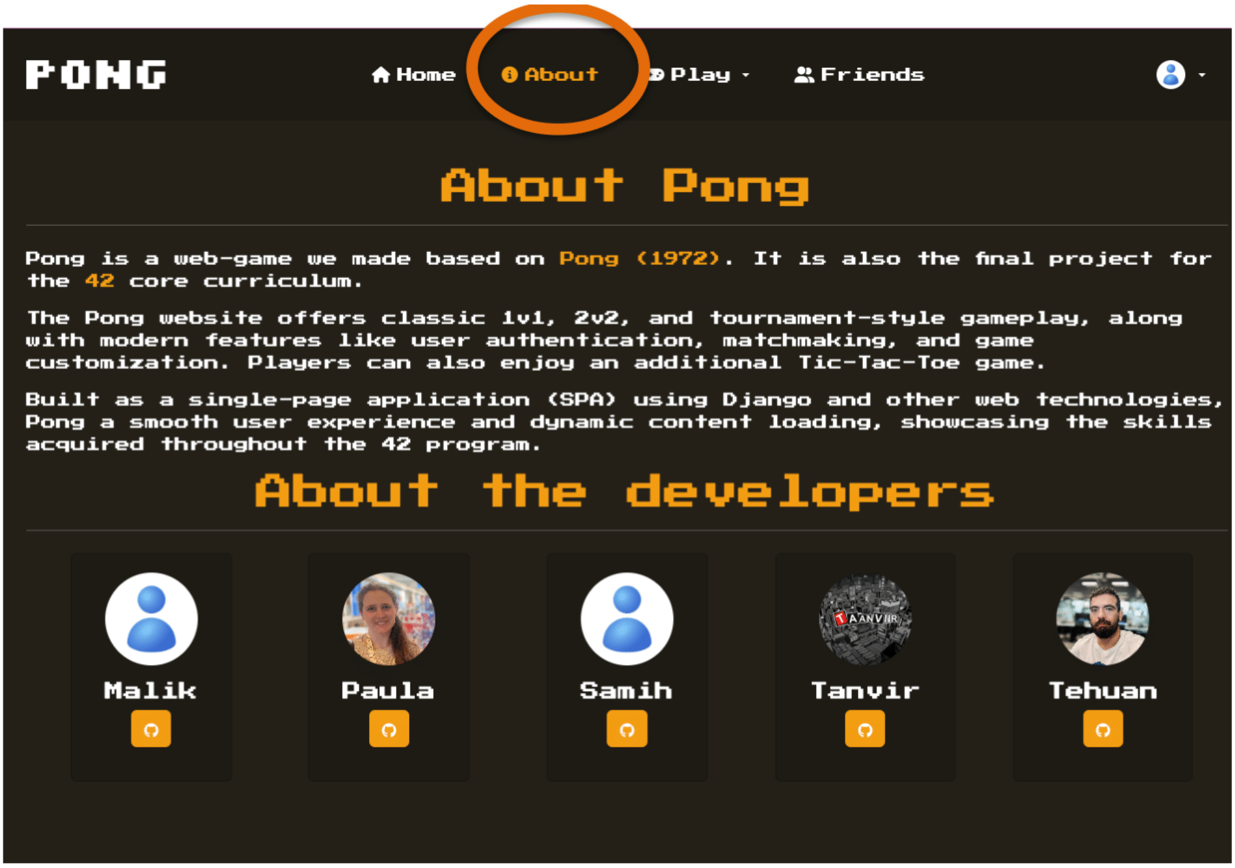
We recommend Google Authenticator for iPhone or Android.

2. After scanning, enter the 6-digit code to enable 2FA.

Enter Code: 6-digit code

FIGURE 3.14: Account Settings and 2FA Authentication

Lastly, the “About” page shows information about the game Pong and the developers in charge of the application, with the corresponding GitHub account:



About Pong

Pong is a web-game we made based on Pong (1972). It is also the final project for the 42 core curriculum.

The Pong website offers classic 1v1, 2v2, and tournament-style gameplay, along with modern features like user authentication, matchmaking, and game customization. Players can also enjoy an additional Tic-Tac-Toe game.

Built as a single-page application (SPA) using Django and other web technologies, Pong a smooth user experience and dynamic content loading, showcasing the skills acquired throughout the 42 program.

About the developers

FIGURE 3.15: About Page

Conclusion

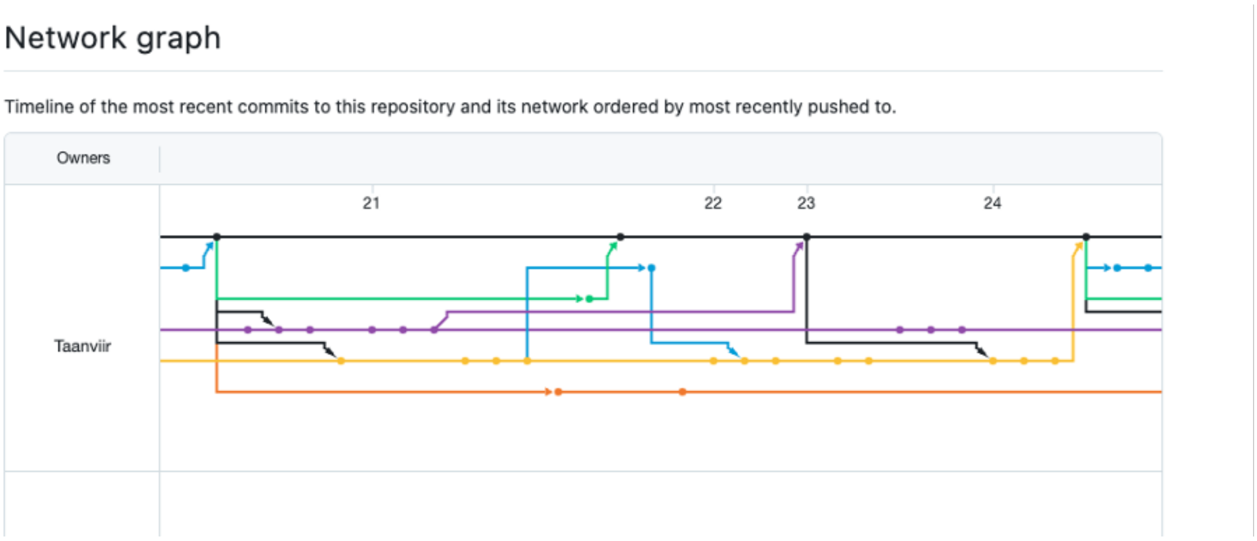
In conclusion, this project achieved its goal of creating an engaging and accessible multiplayer gaming platform by building a web application that hosts classic games, including Pong and Tic-Tac-Toe. Through modern web development techniques, the platform offers real-time interactivity, robust user management, and a secure environment, enhancing user engagement through detailed gameplay statistics and customized matchmaking features. Integrating responsive design and accessibility options further broadens inclusivity, catering to users with varying abilities.

By following the Software Development Life Cycle (SDLC) methodology, the project team systematically analyzed requirements, designed scalable architecture, implemented secure code, and conducted rigorous testing. Each stage addressed specific technical challenges, such as optimizing performance for real-time gameplay, securing sensitive data with JWT and 2FA, and adapting the user interface for multiple devices. Additionally, collaboration through GitHub branches and CI/CD practices ensured streamlined development and quality assurance, allowing for efficient task management and error reduction.

Future work may focus on expanding the game portfolio, adding more accessibility features, and developing the matchmaking system with AI-driven algorithms to improve player experience and even an AI opponent. This project not only underscores the technical capabilities required to build a secure and responsive web-based gaming platform but also demonstrates how thoughtful design can elevate user experience in a competitive digital environment.

Chapter 4

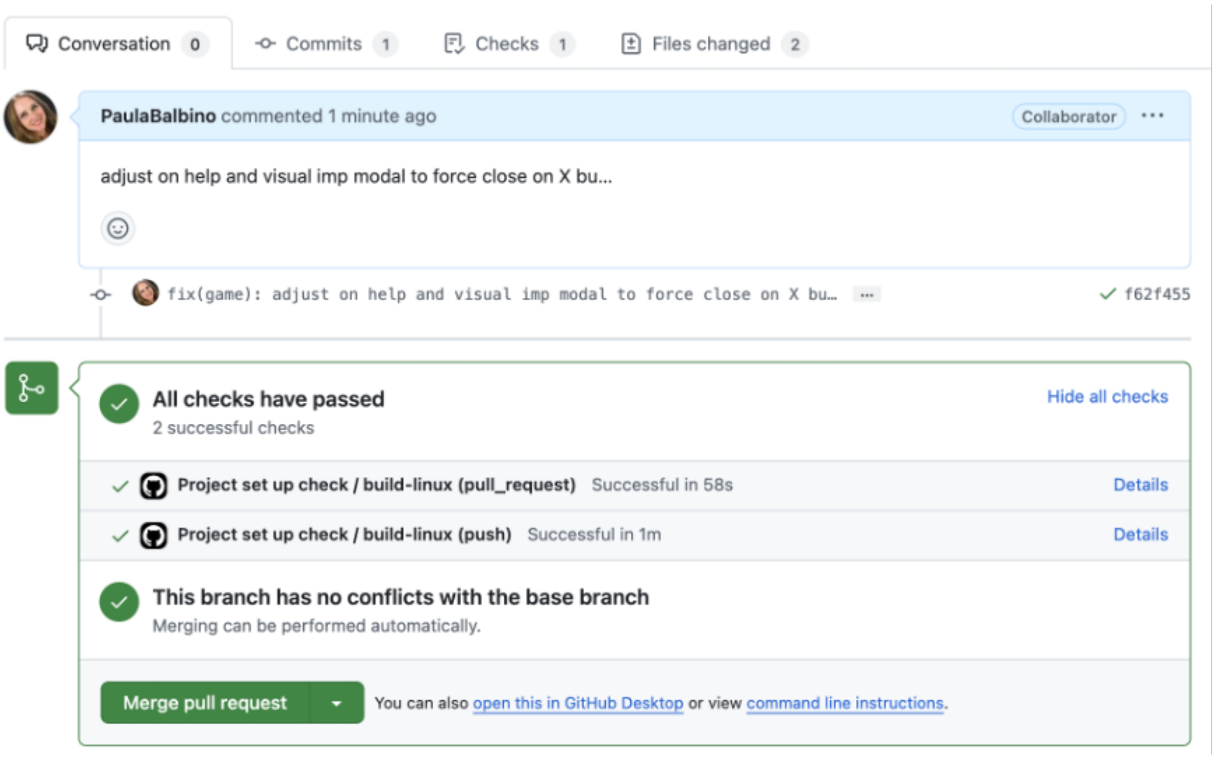
Appendix



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

Network graph of GitHub, representing the main branch (black), fix on the game (green), improvement of design of the pages (purple), features of user management (yellow and blue) and development on the additional game (orange).



PaulaBalbino commented 1 minute ago

adjust on help and visual imp modal to force close on X bu...

```
fix(game): adjust on help and visual imp modal to force close on X bu… ✓ f62f455
```

All checks have passed

2 successful checks

✓ Project set up check / build-linux (pull_request) Successful in 58s Details

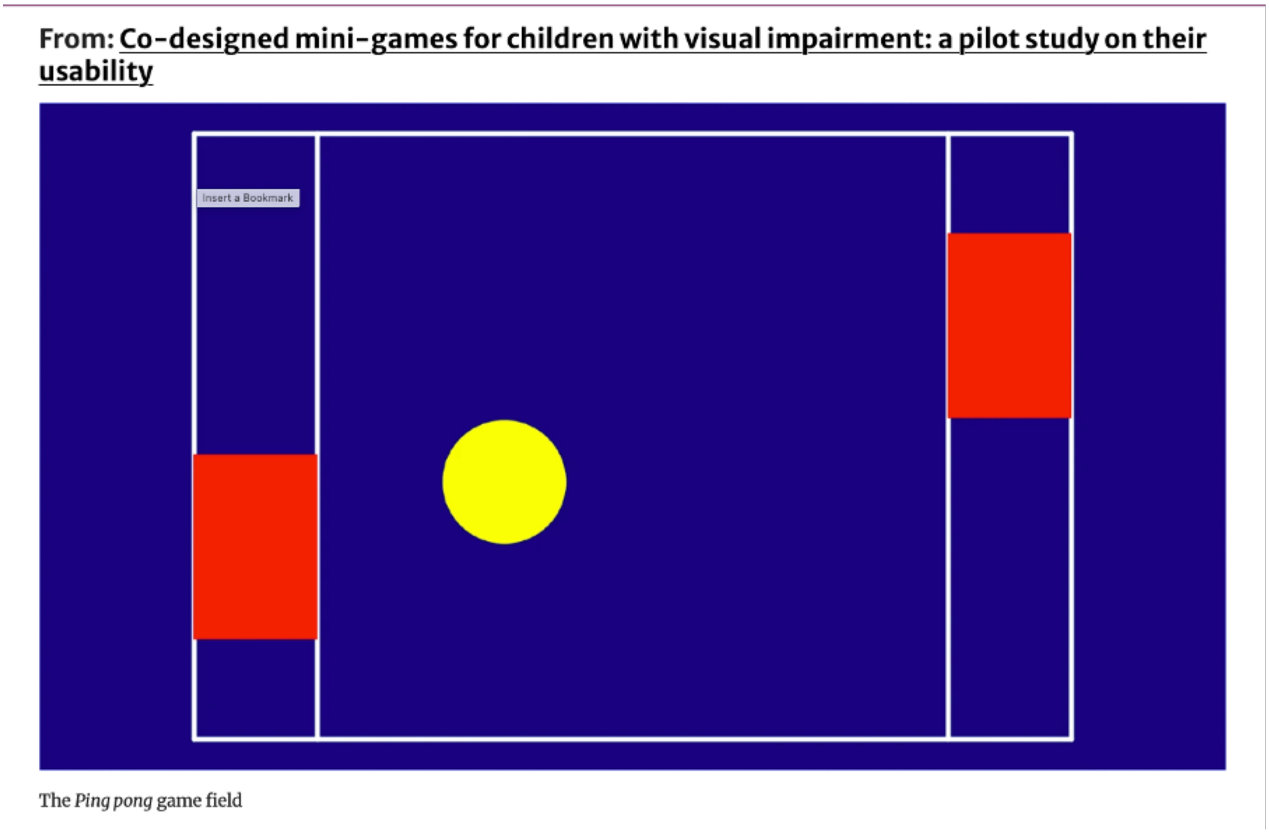
✓ Project set up check / build-linux (push) Successful in 1m Details

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Example of CI/CD integration on GitHub.



From: Co-designed mini-games for children with visual impairment: a pilot study on their usability

The *Ping pong* game field

The visual impairment mode of the project was inspired by the study Co-designed mini-games for children with visual impairment: A pilot study on their usability by Battistin et al. (2022), which explored accessible game design principles tailored for visually impaired children to enhance usability and engagement.

References

[1] 42 (n.d.). Intra API documentation. Retrieved October 20, 2024, from https://api.intra.42.fr/apidoc/guides/

[2] Battistin, T., Dalla Pozza, N., Trentin, S., Volpin, G., Franceschini, A., & Rodà, A. (2022). Co-designed mini-games for children with visual impairment: A pilot study on their usability. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-022-13665-7

[3] Bootstrap. (n.d.). Bootstrap documentation. Retrieved September 15, 2024, from https://getbootstrap.com/docs

[4] Django Software Foundation. (n.d.). Django documentation (version 5.0). Retrieved September 15, 2024, from https://docs.djangoproject.com/en/5.0/

[5] Ecma International. (n.d.). ECMAScript language specification. Retrieved September 25, 2024, from https://www.ecma-international.org/publications-and-standards/

[6] GeeksforGeeks. (2024). What is single page application? Available at: https://www.geeksforgeeks.org/what-is-single-page-application/

[7] Guanabara, G. (2019). Curso de JavaScript [Video]. YouTube. https://www.youtube.com/watch?v=1-w1RfGIov4

[8] Guanabara, G. (2020). Curso de HTML e CSS [Video]. YouTube. https://www.youtube.com/watch?v=1-w1RfGIov4

[9] Programming with Mosh. (2021). Python Django tutorial for beginners [Video]. YouTube. https://www.youtube.com/watch?v=rHux0gMZ3Eg

[10] The PostgreSQL Global Development Group. (n.d.). PostgreSQL documentation, from https://www.postgresql.org/docs/current/

[11] Stakeholdermap.com. (n.d.). Risk assessment matrix - Simple 3x3. Retrieved October 26, 2024, from https://www.stakeholdermap.com/risk/risk-assessment-matrix-simple-3x3.html

[12] Simple JWT. (n.d.). Simple JWT documentation. Retrieved October 18, 2024, from https://django-rest-framework-simplejwt.readthedocs.io/en/latest/

[13] Wikipedia contributors. (n.d.). Single-page application. Wikipedia, The Free Encyclopedia. Retrieved September 8, 2024, from https://en.wikipedia.org/wiki/Single-page_application