



Table of Contents

[**1. Introduction: 2**](#_heading=h.ogvepf8fza3j)

[**2. Overview of the Software: 3**](#_heading=h.f1vd54kwa09d)

[**3. Purpose of the User Guide: 3**](#_heading=h.qaiqsp773k25)

[**4. Target Audience: 4**](#_heading=h.huxp72qxzpf6)

[**5. Getting Started: 5**](#_heading=h.aqx590mz5m2f)

[**6. Installation Instructions: 6**](#_heading=h.m7a7vsnbn6m4)

[**7. System Requirements: 7**](#_heading=h.r5d6r84llz2i)

[**8. User Interface Overview: 8**](#_heading=h.w19yhpm0req)

[**9. Using the Software: 10**](#_heading=h.w27ikravuy3u)

[**10. Step-by-step instructions for using the software's main features: 12**](#_heading=h.rms55ybn9gte)

[**11. Troubleshooting Tips: 14**](#_heading=h.ojxionrawrbl)

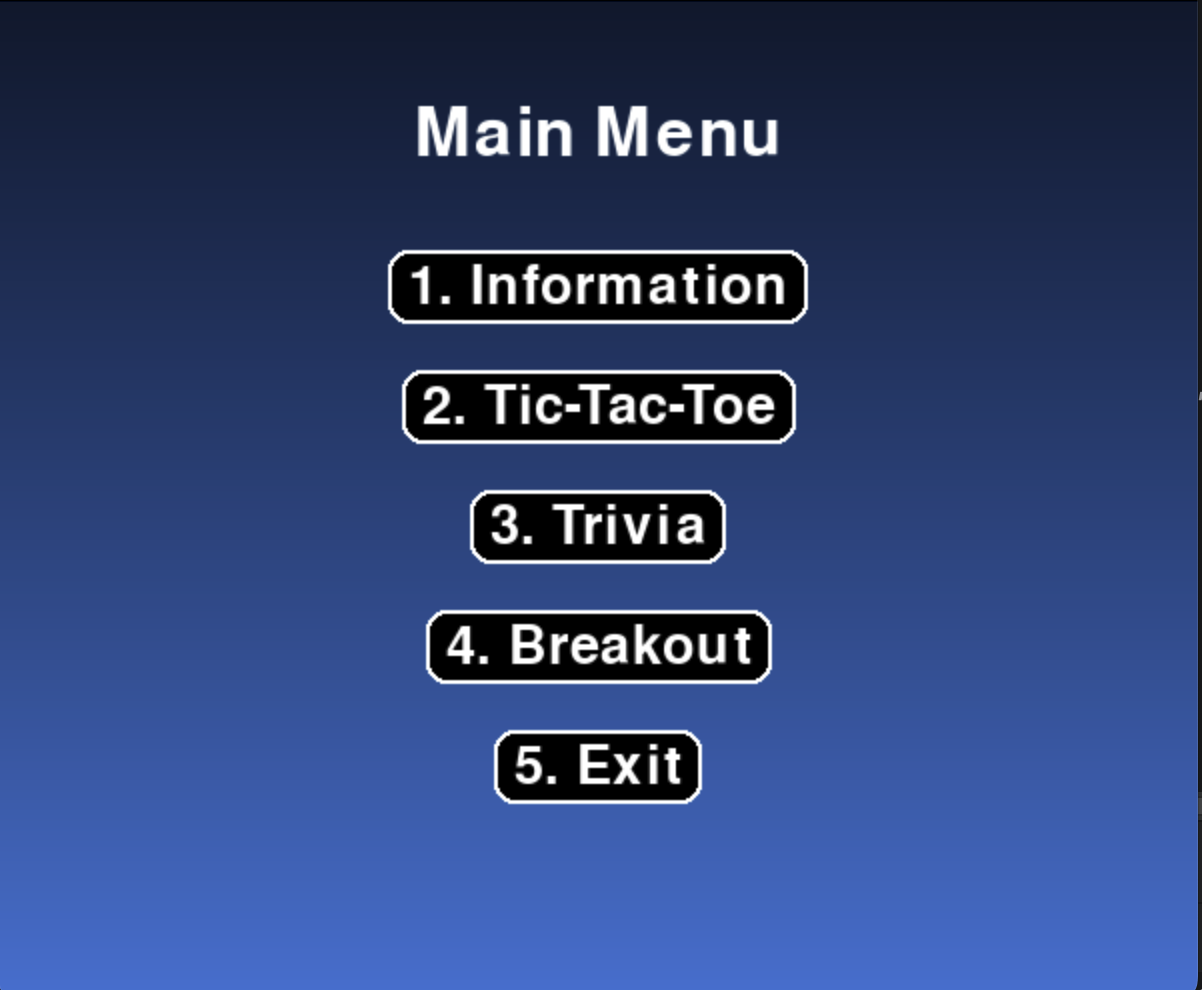
[**12. Take Action: 16**](#_heading=h.vf0xh24ff7z1)

[**13. References: 17**](#_heading=h.4dupjksce5xa)

[**14. Glossary of terms: 19**](#_heading=h.ejwepbol1csc)

[**15. Index 22**](#_heading=h.mf7jgxmn57ub)

# **Introduction:**

Welcome to the **Python Game Hub**, an interactive desktop gaming platform developed using Python and the Pygame library. This project is designed to provide a seamless user experience by integrating multiple mini-games under a single main menu. Whether you’re testing your wits with Trivia, strategizing in Tic-Tac-Toe, or enjoying the reflex-based challenge of Breakout, this hub offers engaging gameplay with intuitive navigation and vibrant visual design.

**Key Features:**

* **Centralized Main Menu**: A visually dynamic menu interface lets you quickly launch any game or exit the hub.
* **Information (online):** An information button that requires the internet to be redirected to a hyperlink of the GitHub Repo and README.md.
* **Mini-Games Collection (offline)**:
  + **Tic-Tac-Toe** – Classic strategy game with a smart AI opponent.
  + **Trivia** – Timed quiz with multiple-choice questions and score tracking.
  + **Breakout** – Paddle-and-ball arcade game with sound effects and block-breaking mechanics.
* **Sound Integration**: Background music and game-specific sound effects enhance immersion.
* **Mouse & Keyboard Support**: Menu navigation and gameplay support both input modes.
* **Visual Enhancements**: Gradient backgrounds, hover effects, and animations improve aesthetics and usability.



This guide will help users get started, navigate the hub, and understand gameplay mechanics for each game module. 🧐

Whether you’re a casual gamer or testing functionality for development, the Python Game Hub offers a polished and fun experience. 🎉

# **Overview of the Software:**

🎮 The **Python Game Hub** is a desktop-based, multi-game application developed as a capstone project for CMSC 495. Built using Python and the Pygame library, the software provides users with a seamless interface to play three mini-games: Tic Tac Toe, Trivia, and Breakout. The other option is the Information button, which requires internet for the README.md to get access to history, details, and testing. These games are integrated into a centralized Main Menu, allowing players to navigate easily between them with mouse input. ⌨️🖱

The purpose of the Python Game Hub is to deliver an engaging, easy-to-use application that demonstrates modular design, teamwork, and core principles of software development, such as clean architecture, test-driven development, and multi-platform compatibility. ⚙️ 🛠️

Each game was designed to be independent yet stylistically consistent, providing a unified player experience. The games were coded using object-oriented principles and feature a mix of real-time logic, artificial intelligence, UI design, and user input handling. 💻

# **Purpose of the User Guide:**

The purpose of this user guide is to help players of all experience levels easily navigate and enjoy the **Python Game Hub**. This guide provides clear, step-by-step instructions on how to access each game, understand the rules, and use the controls to play effectively. ✍️

💡 With this guide, users will:

* Learn how to start the Game Hub and interact with the main menu.
* Understand how to play each mini-game: **Tic-Tac-Toe**, **Trivia**, and **Breakout**.
* Get familiar with game controls using the keyboard and mouse.
* Discover how scoring works and how to restart or exit games.
* Troubleshoot basic issues like freezing, sound not playing, or window resizing.

**📙**Whether you’re a casual gamer looking for fun or just trying out each mini-game, this guide ensures you get the most out of your experience—without needing any technical knowledge.

# **Target Audience:**

**Python Game Hub**

🤗 *Built-In Gaming for Shared Spaces*

The Game Hub is a simple, app-based gaming portal designed to be pre-installed by default on shared devices in schools, libraries, and hotels. This application features a collection of nostalgic, easy-to-play games such as Tic-Tac-Toe, Breakout, and Trivia. It is built with inclusivity in mind, ensuring that users of all age groups can enjoy quick, engaging games without the need to log in or connect to the internet. The Game Hub provides a fast, fun, and frictionless experience perfect for public settings.

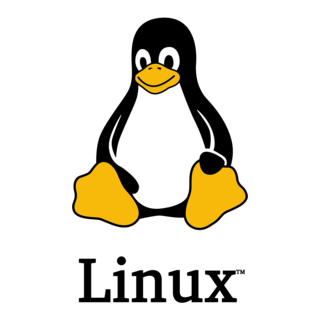
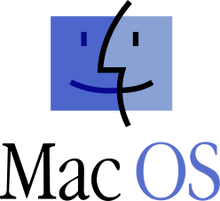
This app addresses the need for hassle-free entertainment in shared spaces. By eliminating the need for setup, downloads, or accounts, it simplifies access to digital entertainment. Whether it's a short mental break for students, a quick distraction for hotel guests, or light recreation for library visitors, The Game Hub enhances user experience without demanding administrative maintenance or supervision. 

The Game Hub targets institutions that serve diverse user bases—primarily educational environments, hospitality venues, and public information centers. Decision-makers such as IT administrators, library coordinators, and guest experience managers are the primary audience for this app. These stakeholders prioritize simplicity, reliability, and universal appeal in the software they provide to their communities.

By offering offline functionality, a clean and intuitive interface, and games that promote safe, distraction-free fun, The Game Hub stands out as a ready-to-use solution. It includes a two-player Trivia mode to encourage social engagement, making it ideal for short bursts of activity during downtime or waiting periods. Its pre-installed nature ensures that it's always available, regardless of internet access or user login status. ​​Ultimately, The Game Hub is more than just a collection of games—it’s a digital amenity that brings effortless enjoyment to public and semi-public environments. Its low-maintenance design and high accessibility make it a valuable addition to any shared-use device, fostering positive engagement through nostalgia and simplicity.

# **Getting Started:**

**😊 To get started, this is a very quick *summary* of the User Guide to skip the detailed steps!**

**🛠️ *System Requirements***

Operating System: Window, MacOS, or Linux

Python Version: 3.9 or higher

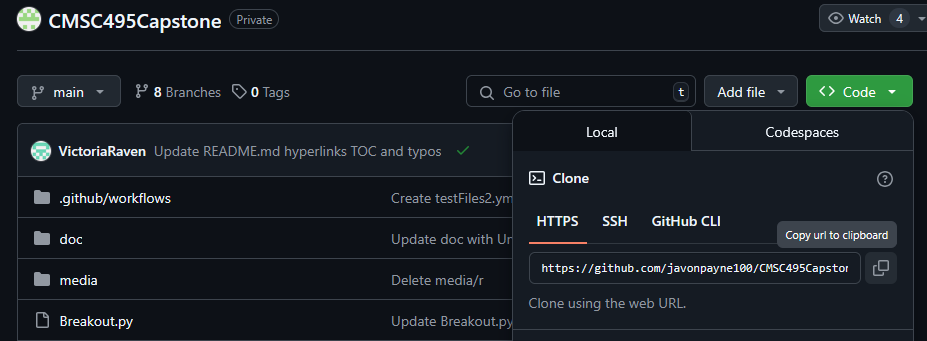
Required libraries: pygame, numpy, pyautogui

**🛠️ *Installation Instruction***

1. **Clone the Repository**

Enter Github link provided: <https://github.com/javonpayne100/CMSC495Capstone/tree/main>

Click on ‘Code’ and click the copy button to copy the project.



1. **Set Up the Environment 🌐**

Install all dependencies through the following command in the terminal/command prompt inside the project directory once you have the project installed/downloaded:  


1. **Launch Game Hub ▶️**

Upon setting up your environment, you can enter `python MainMenu.py` in the terminal, inside your project directory to launch the Game Hub.



### **🎮 4. Quick Controls and Navigation**

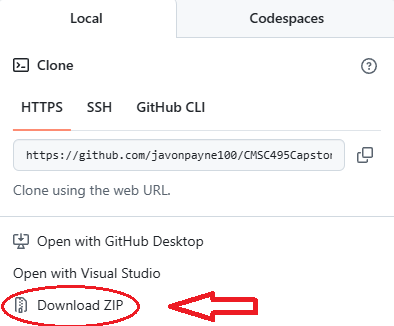
* 1. Once on the Main Menu
  2. Use your mouse to navigate the main menu.
  3. Click on:
     1. **"Tic-Tac-Toe"** to play vs AI
     2. **"Trivia"** to answer questions against a timer
     3. **"Breakout"** to destroy bricks with a bouncing ball
     4. **"Information"** to open the GitHub project page
     5. **"Exit"** to close the game hub

**⚠️Note: Refer to the** [**Installation**](#_heading=h.m7a7vsnbn6m4) **and** [**System**](#_heading=h.r5d6r84llz2i) **/** [**README.md**](https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1) **for more steps.**

# **Installation Instructions:**

**❗This should be more detailed instructions if the** [**Getting Started**](#_heading=h.aqx590mz5m2f) **Section is not enough!**

**🛠️Option 1: Download as Zip (No GitHub Account Required)**

1. Navigate to the GitHub repository:  
   <https://github.com/javonpayne100/CMSC495Capstone>
2. Click the green **** button → ****
3. Extract the ZIP contents into a folder. 
4. Open the project in your Python IDE like PyCharm
5. Run the Application in Python IDE.

**🛠️Option 2: Clone Repository (GitHub Account Required)**

1. Ensure you are logged into GitHub.
2. Fork the repository:
   * Click **** → **** → Select **main** branch only.
3. Open a terminal  and run:
4. git clone https://github.com/YOUR\_USERNAME/CMSC495Capstone.git
5. Open the cloned folder in your IDE or terminal.

**▶️Run the Application**

1. Make sure your terminal/IDE is pointing to the root directory of the project.
2. Install dependencies in the terminal:
3. *python -m pip install --upgrade pip*
4. *pip install -r requirements.txt*
5. Launch the Game Hub:
6. *python MainMenu.py*
7. Or use the PyCharm current file, and run the current file with the green button. 

**Note: Refer to the** [**README.md**](https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1) **file for more details and steps!**

**⚠️README.md:** <https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1>

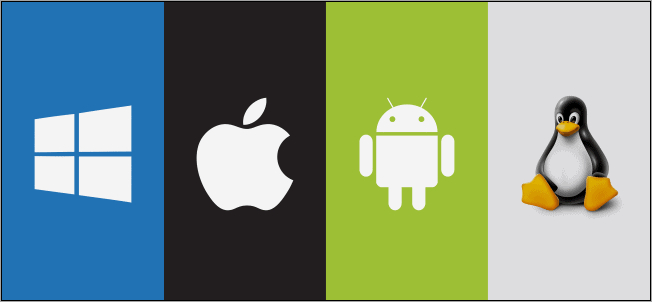
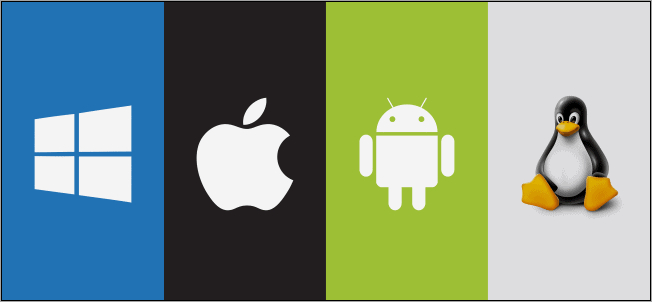
**⚠️ Any other issues/concerns? Please Refer to the** [**Take Action**](#_heading=h.vf0xh24ff7z1) **Section!**

# **System Requirements:**

**❗This should be more detailed instructions if the** [**Getting Started**](#_heading=h.aqx590mz5m2f) **Section is not enough!**

The **Python Game Hub** is a desktop-based application built using Python and the Pygame library. To run the software smoothly, users must have Python 3.9 or higher installed on their system, along with required libraries such as pygame, numpy, and pyautogui. It is compatible with Windows, macOS, and Linux operating systems. The application can be launched on any desktop or laptop with at least a 600x400 screen resolution and standard audio/video drivers enabled. After accessing the GitHub repository, users can either clone or download the project, install the necessary dependencies, and start the Game Hub using a Python IDE or terminal.

| **Component** | **Minimum Requirement** |
| --- | --- |

**Operating System:** Windows, macOS, or Linux 

**Python Version:** Python 3.9+

**IDE (Optional):** PyCharm, or any Python IDE

**Libraries/ Dependencies:** Pygame, json, random, math, time, 

sys, unittest, numpy, pyautogui

**Screen Resolution:** 600x400 or higher

**Drivers:** Audio, System audio, video, and display

must be enabled 

**Terminal Tools:** Command Prompt / Terminal / IDE Terminal 

/ Git Bash

**Computer Type:** Laptop or Desktop (Model Year: 2017-2025+)

**⚠️README.md:** <https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1>

**⚠️ Any other issues/concerns? Please Refer to the** [**Take Action**](#_heading=h.vf0xh24ff7z1) **Section!**

# **User Interface Overview:**

**UI Flow Chart**

*Screen Navigation*

**❗Purpose/Objectives:** This section should help you understand how to navigate through an application, what screens or pages you will interact with, and how each element or button leads to another part of the system. It's like a visual map of the user journey through the software.

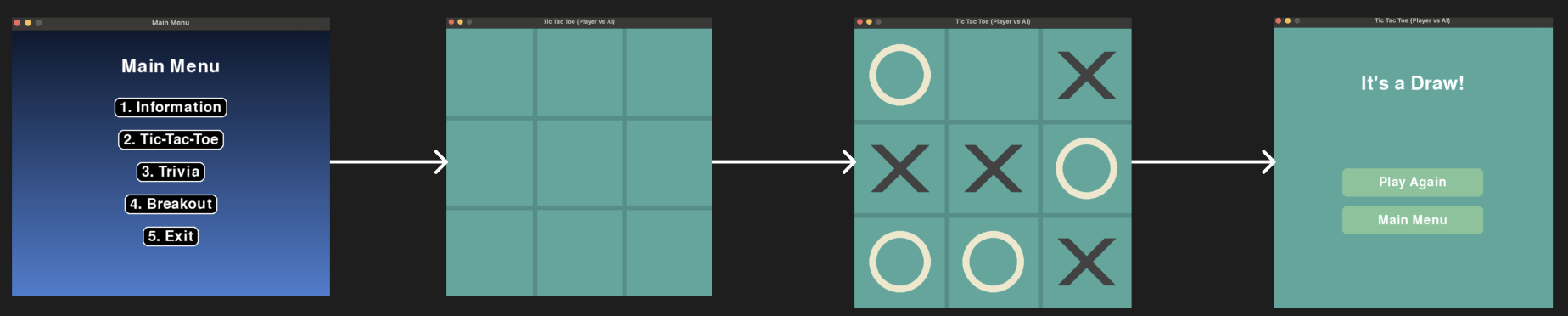
**⚠️Note:** To view UI flow more interactively, navigate the following link and open with Draw.IO [GameHubFlow.drawio](https://drive.google.com/file/d/1volF0kf8yUY_mpPorQwHTohgThDG2Etc/view?usp=sharing)



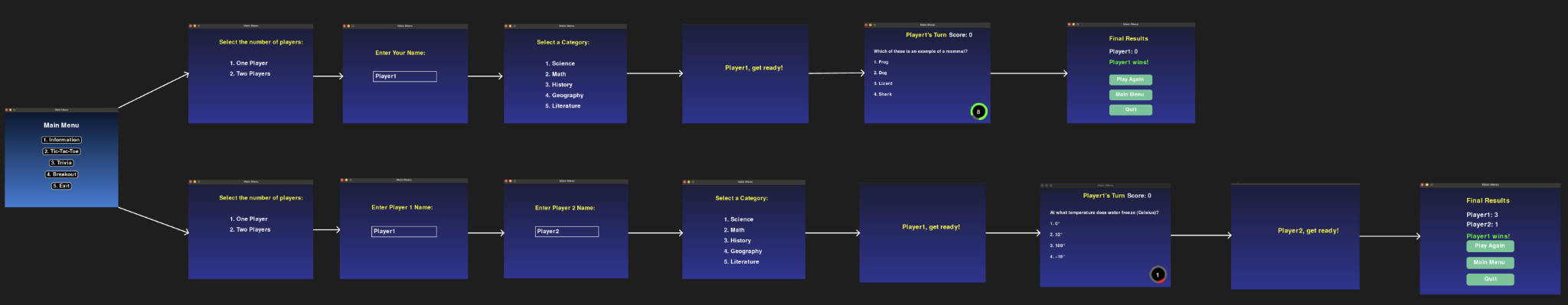
**UI Screen Flow**

*Game Visual*

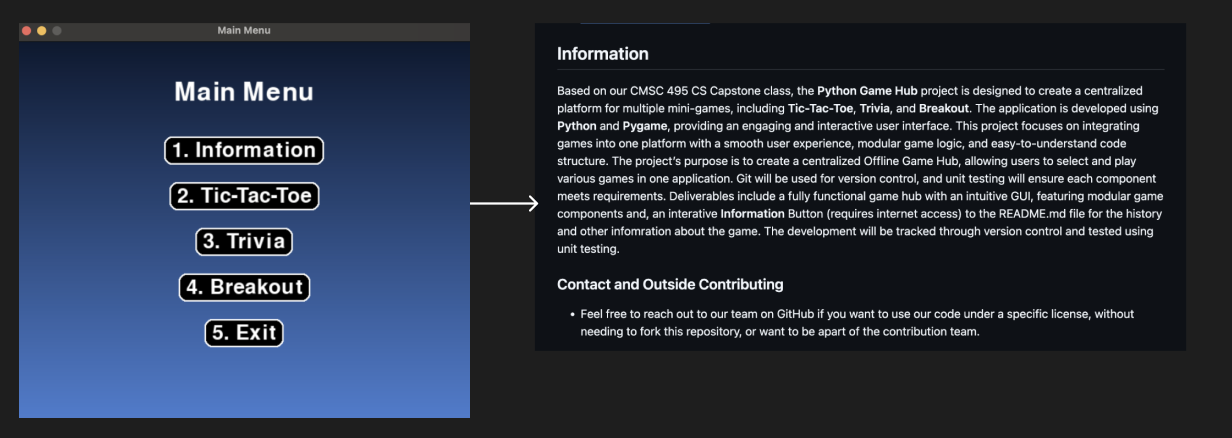
1️⃣Tic-Tac-Toe Flow: Click link for full view [Tic-Tac-Toe Flow](https://www.figma.com/design/gigetgeab53Ot7ywj5cEaP/Tic-Tac-Toe-Flow?t=XmUkM8XdVs0UKcQI-1)



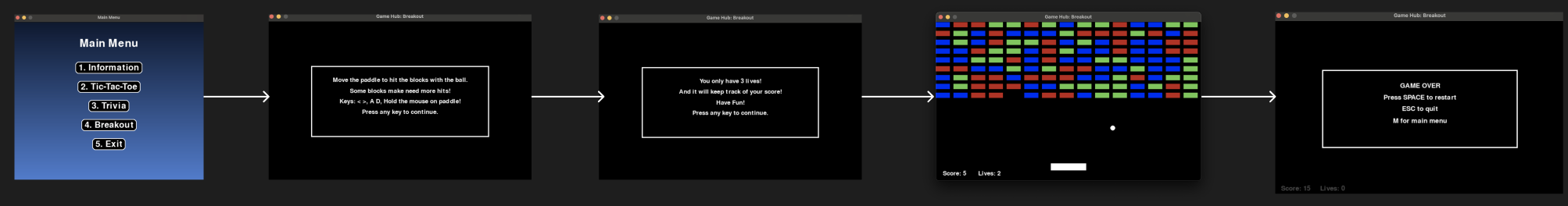
2️⃣Trivia Flow: Click link for full view [Trivia Flow](https://www.figma.com/design/VItUZcalpGeJPNbmExQhZD/Untitled?node-id=0-1&t=XmUkM8XdVs0UKcQI-1)



3️⃣Information Flow: Click link for full view [Information](https://www.figma.com/design/7jgNXyHOpBQacpmPb03PG1/Untitled?t=3SdIWTyzHHxQn1St-1)



4️⃣Breakout Flow: Click link for full view [Breakout Flow](https://www.figma.com/design/h5ld6kwRN8taRVs5edgNmm/Untitled?node-id=0-1&t=3SdIWTyzHHxQn1St-1)

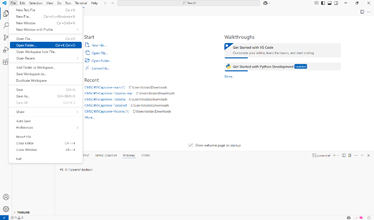


# **Using the Software:**

**❗Note: A *quicker* version of this is in the** [**next section (Step-by-Step)**](#_heading=h.rms55ybn9gte) **for main features!**

**😊This is a Rule(s)/Information(s) overview of Using the Software!**

The **Python Game Hub** is a convenient and user-friendly application that brings together three classic games—**Tic Tac Toe**, **Breakout**, and **Trivia**—into one easy-to-access platform. After installing the software and necessary dependencies, users can launch the program by opening their IDE or terminal, navigating to the Game Hub folder, and running the command **python MainMenu.py**. This brings up a central Main Menu where users can select the game they want to play using their mouse.

Each game offers a distinct experience. *Tic Tac Toe* allows for either two-player matches or player-versus-AI mode, with the AI powered by the Minimax algorithm for intelligent gameplay. Players place Xs or Os by clicking on the grid, and a message displays the outcome at the end of the game. *Breakout* is an arcade-style game where players use the left and right arrow keys to move a paddle, keeping the ball in play to break blocks. Lives and scores are shown throughout the game. *Trivia* offers both single and two-player options, challenging players with multiple-choice questions under a countdown timer that changes color to indicate time pressure. Final scores are displayed after all questions are answered. 

**After Installing all the Requirements, system, and IDE (Python), check to make sure these are correct:** Open your IDE or open a terminal and navigate to the folder where the Game Hub files are saved and open the folder. And **check** if you have the **dependencies installed**! **Refer to the image above! 😊**

Then check if you are on the **Main Menu** with *options*:

**(1) Information; (2) Tic Tac Toe; (3) Trivia; (4) Breakout; (5) Exit**

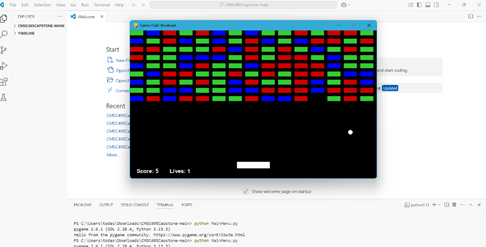
### **📜Information Details/Rules (Requires Internet)**

* Open GitHub link to the Python Game Hub repository.
* View README.md for gameplay instructions, history, and development notes.

### **📜Tic Tac Toe Details/Rules (Offline)**

* AI = X, Player = O; AI moves first, then player; AI uses Minimax algorithm for smart responses
* Use mouse clicks to place your move.
* Win with three X's or O's in a row; game checks win/loss/draw after each move.
* Option to replay or return to menu after the game ends.

### **📜Trivia Details/Rules (Offline)**

* Choose 1 or 2 player mode; Enter your name and pick a trivia subject.
* Answer multiple-choice questions before the timer runs out (green, yellow, red).
* Score increases with each correct answer.
* In 2-player mode, scores and winners are shown.
* Replay or return to the menu after the quiz ends.

### **📜Breakout Details/Rules (Offline)**

* Move the paddle with left/right keys.
* Bounce ball to break blocks
* Three lives total; and scores shown
* option to restart or exit the menu.

### **📜Exit Details/Rules (Offline)**

* Closes the Python Game Hub and ends the program.

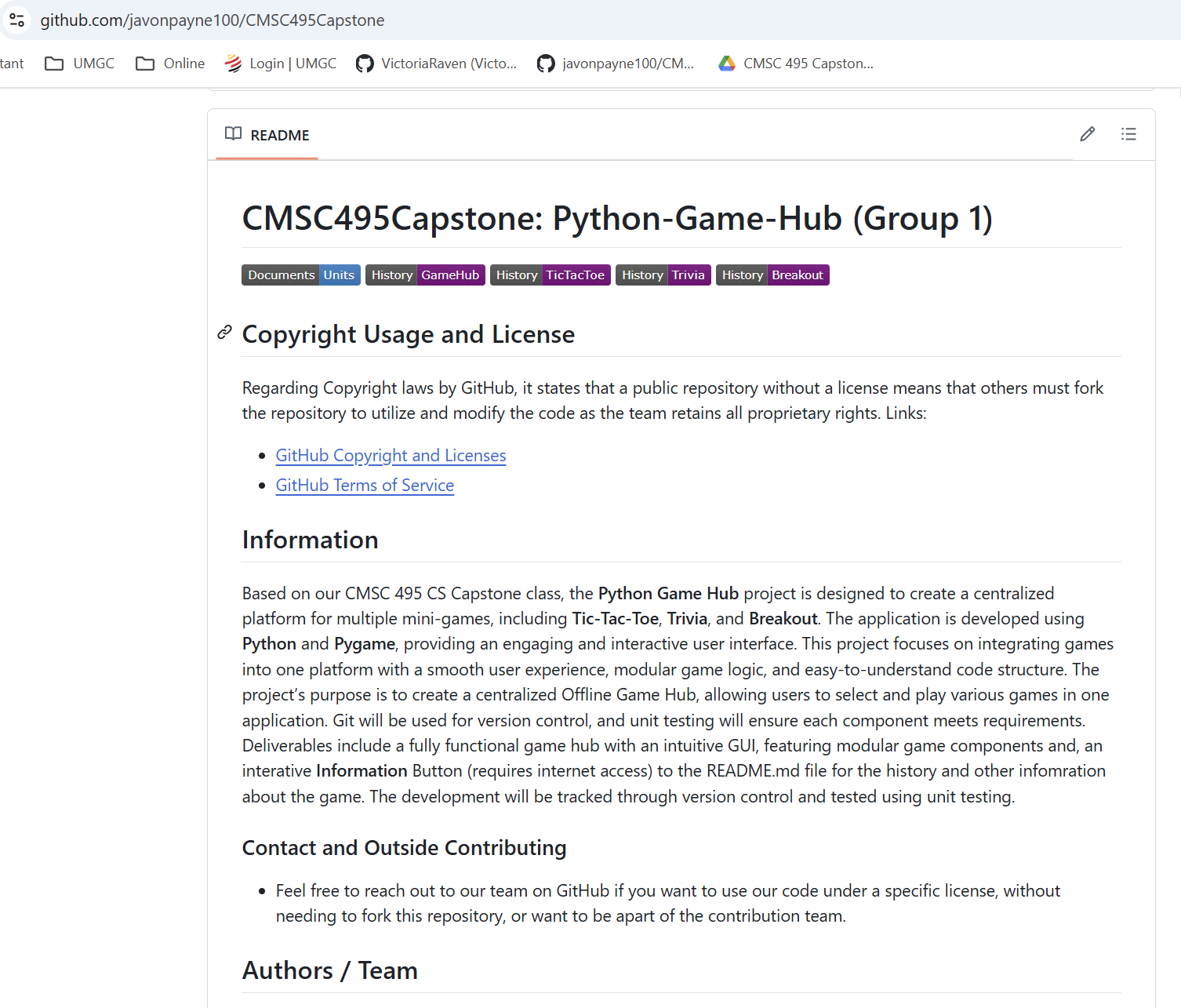
Whether you're in the mood for a quick puzzle, a fast-paced arcade challenge, or a brain-teasing quiz, the Python Game Hub offers fun and variety for players of all skill levels and caters to a wide range of interests.

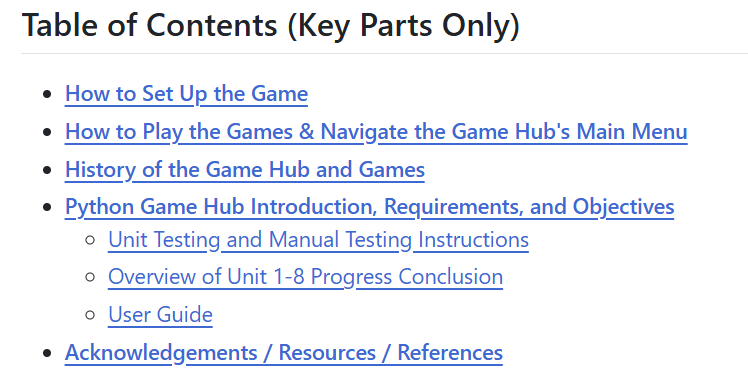
**❗**For a quicker or detailed usage guidance of the main features in this software, users can consult the[**step-by-step instructions in the next section**](#_heading=h.rms55ybn9gte) or **README.md:** <https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1>

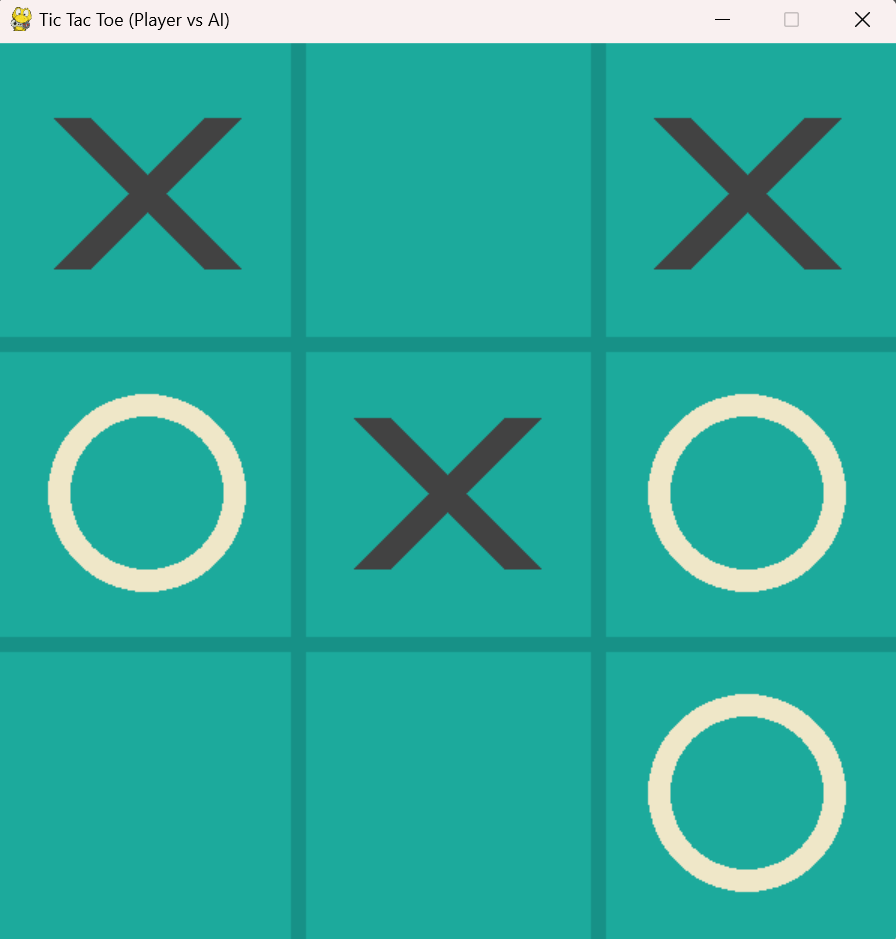
# **Step-by-step instructions for using the software's main features:**

❗This is a **quicker breakdown** of [**Using the Software**](#_heading=h.w27ikravuy3u) *without* the *Rules/Information* (**Main Menu)**.

* Once in the **Main Menu**, you may select the numbered options #2-4 to play a game.
  + Option 1 requires internet access and redirects users to the GitHub Repository page for the README.md file (including history links).
  + Options 2-4 are 100% offline and do not require internet to play the games
* If **Information** is selected, the menu will redirect the user to the GitHub Repository page where the user can scan the README.md file for other information (requires internet)
  + This includes a History of the Games
  + This includes Installation/Playing them (detailed instructions)
  + Scrolling down to the Table of Contents, you can use this for more information:



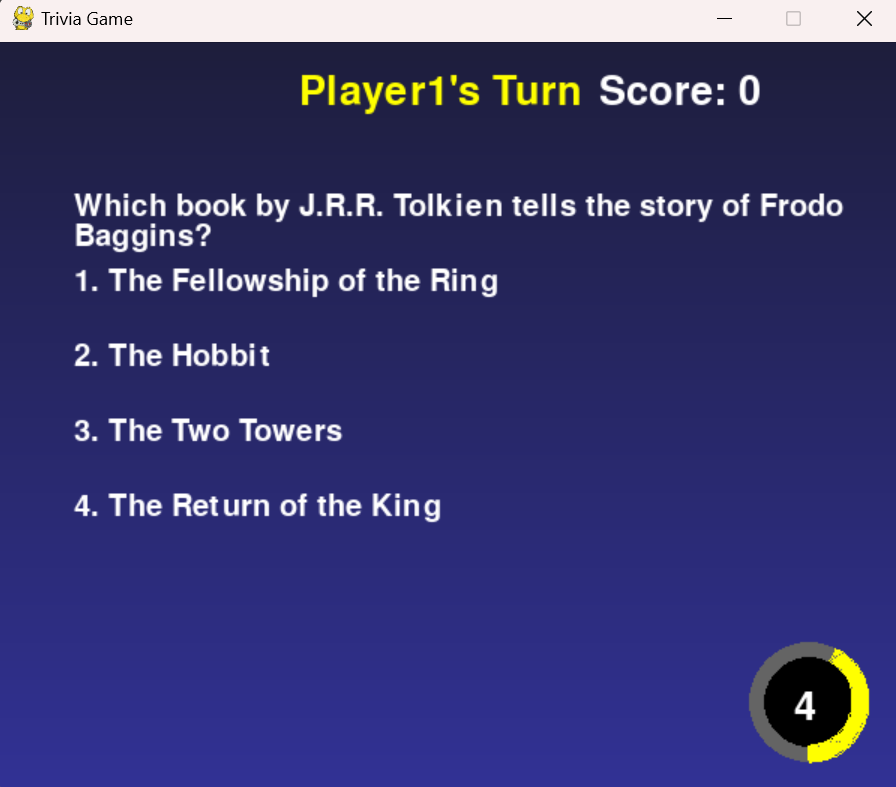
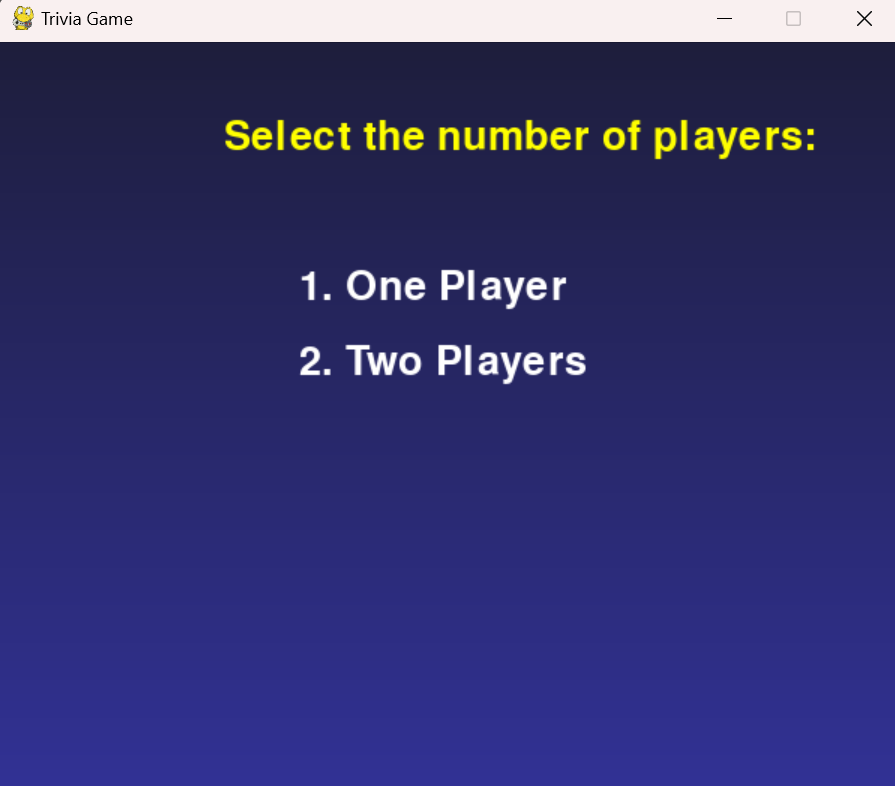


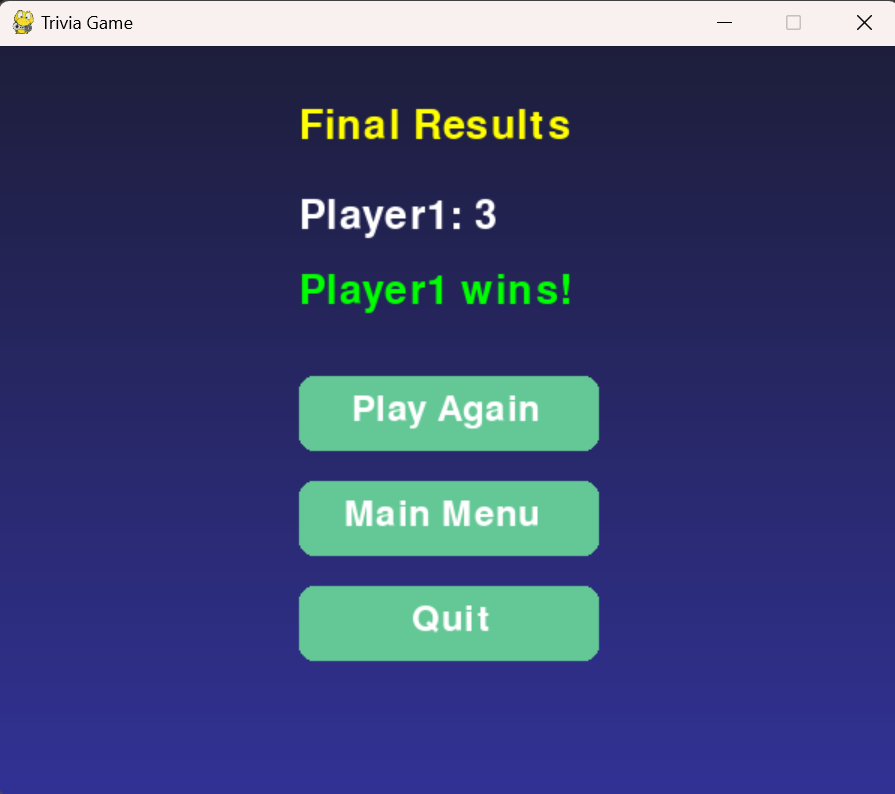
* If **Tic-Tac-Toe** is selected, the game will load up, and you will play against an AI opponent with the objective of getting a straight line with your Os.

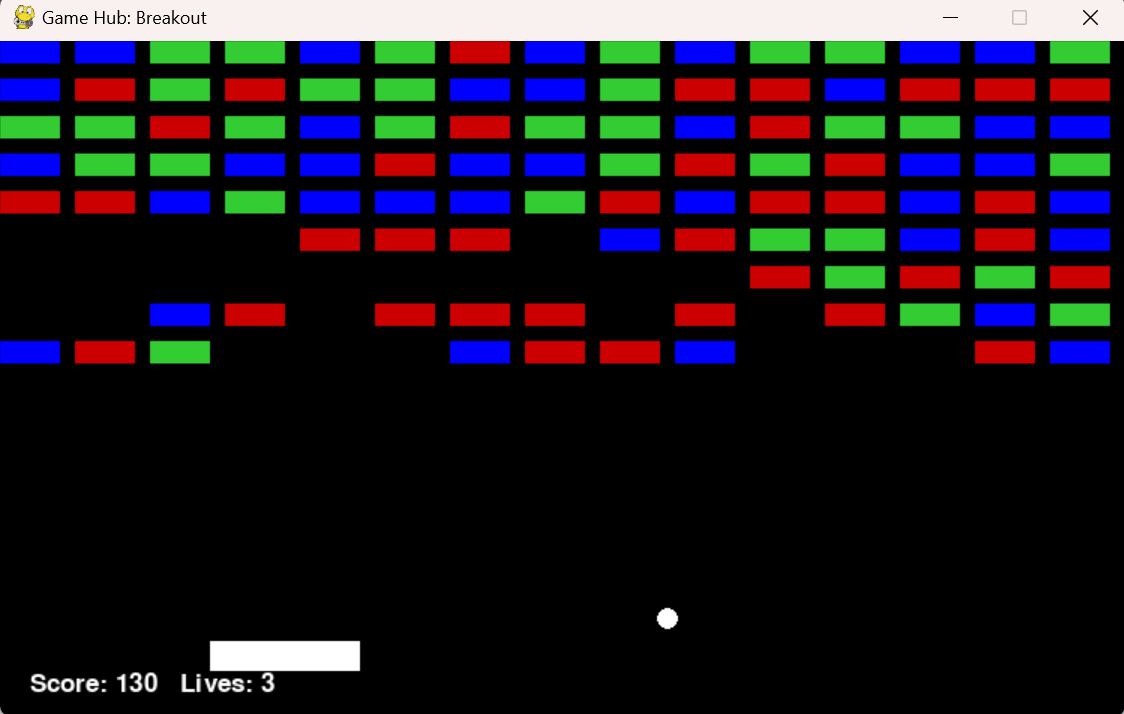
**Refer the image to the left!**

* If **Trivia** is selected, you may select and enter names for one player or two players. Choose the category to test yourself on, and the questions and countdown begins. Answer all the questions before time runs out and the score will be displayed in the end.

**Refer to the images below (left to right to down)!**







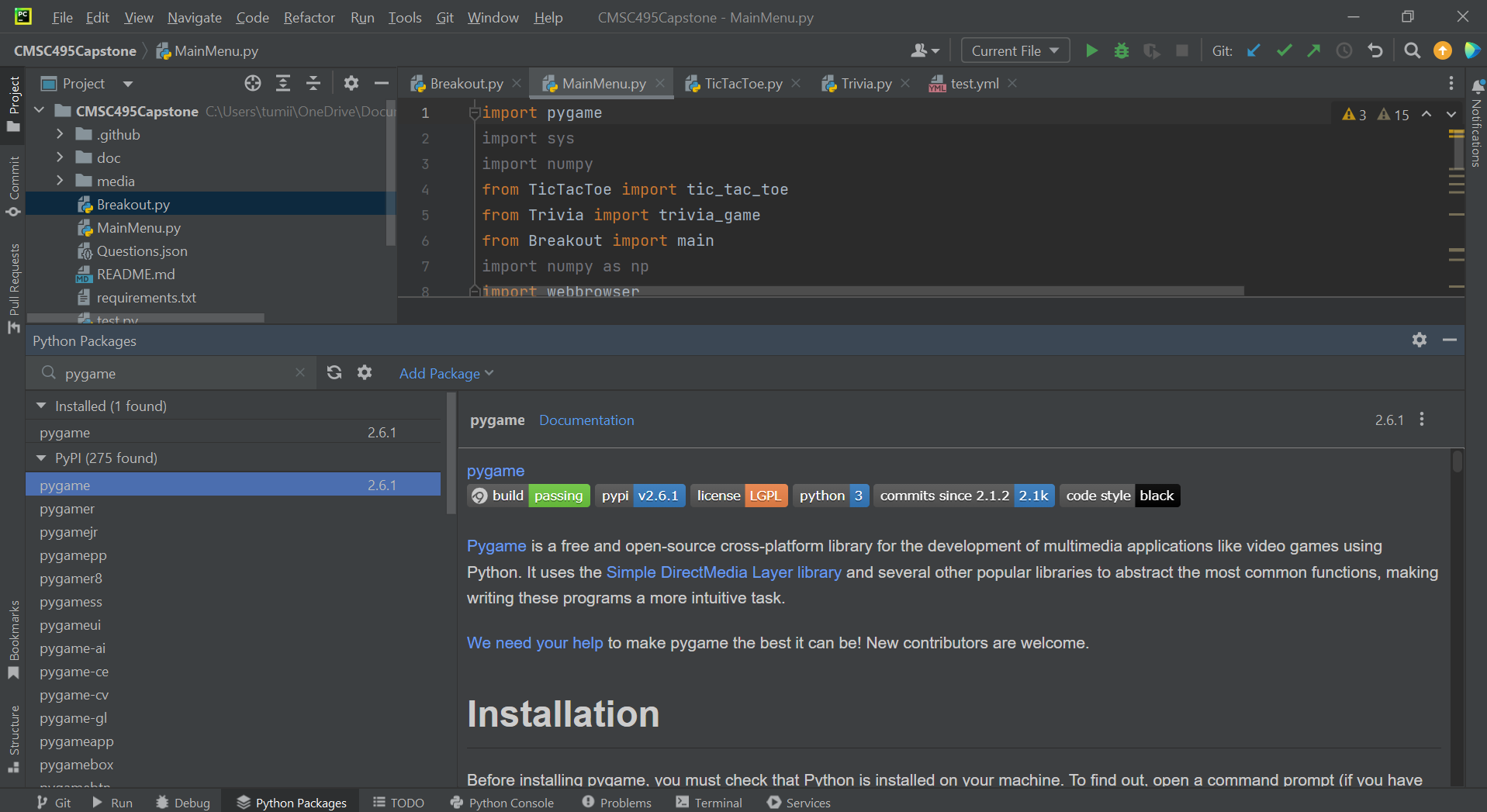
* If **Breakout** is selected, you may control the paddle with your mouse or the left & right buttons on your keyboard. The goal is to clear all the bricks with the ball without losing all your lives. Some blocks may need to be hit more than once to clear them.

**Refer to the left image!**

* At the end of any game/round, you will have the option to return to the **main menu** to **play a different game/play again** or to **exit to desktop**.

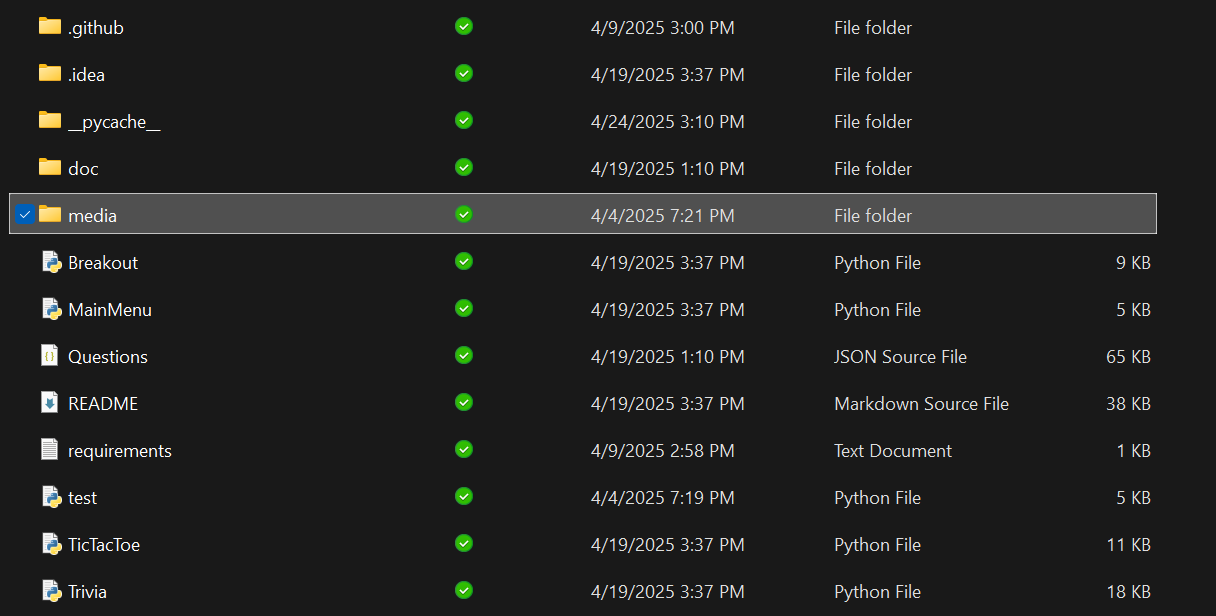
# **Troubleshooting Tips:**

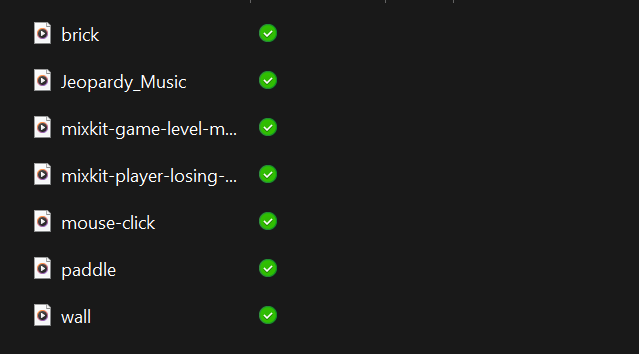
* **🛠️MainMenu.py does not launch:** Ensure you have the following packages installed in your IDE: pygame, sys, numpy, webbrowser, random, math, time, and json. These can be found in the Python Packages tab of your IDE. Below is an example for PyCharm.



***Troubleshooting Tips (Continued):***

* **🛠️No music or sound played:** In order to have music and sounds playing while using the Game Hub, the ‘Media’ folder needs to be in the same directory as the rest of the files. Ensure the folder is not empty. If it is empty, you have to redownload the entire Game Hub zip file.

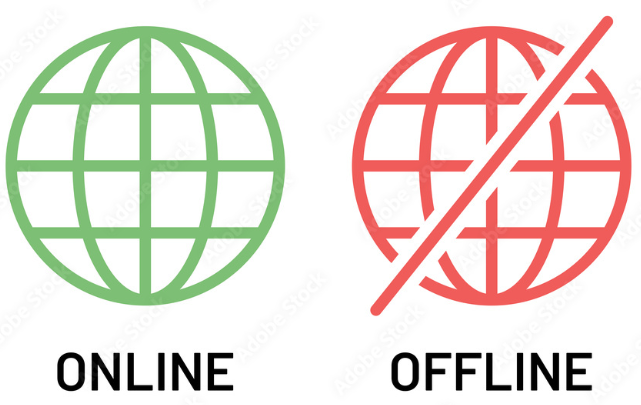


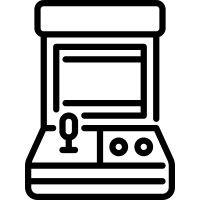
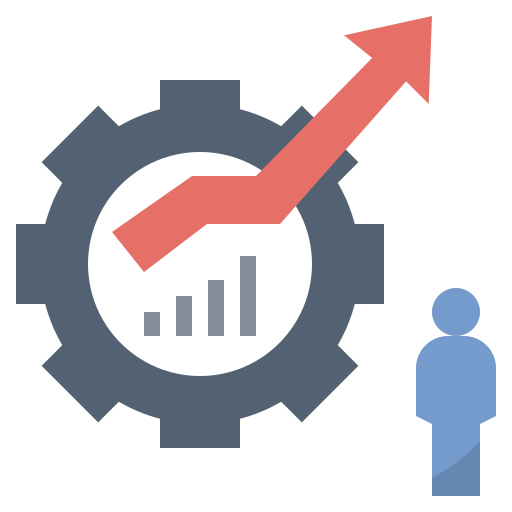


* **🛠️Mouse does not move to select or play games:** Outdated or incompatible drivers may cause the mouse to not function as required. Update all mouse drivers to the latest version, then restart the game. If using an external mouse, ensure that it is plugged into your computer correctly.
* **🛠️Keyboard does not work:** Just as for the mouse, outdated drivers may cause the keyboard to not function properly. Updating the drivers may fix this issue as well.
* **Any other issues? Please Refer to the** [**Take Action**](#_heading=h.vf0xh24ff7z1) **Section (next one or below)!**

# **Take Action:**

👨‍🔧Hello there, here’s what you can do after reading this **Python Game Hub User Guide**!

**▶️Play the Games**

* Launch **MainMenu.py**
* **Select a game/option** from the interactive menu
* **Information** requires the internet (online), while the games are offline (no internet)
* Enjoy playing **Tic Tac Toe**, **Trivia**, or **Breakout!**

**⚙️Contribute to Development**

* Fork the repository and make enhancements or bug fixes
* Submit a **pull request** on GitHub
* Reach out to the team to join as a contributor

**🚀Help Rerun-Tests**

* **Go to the README.md file for the steps to run the Tests!**
* **README.md:** <https://github.com/javonpayne100/CMSC495Capstone#cmsc495capstone-python-game-hub-group-1>
* Re-run the Testings:
  + **main** branch has 13 unit tests (old test.py file)
  + **docs in unit 5** has Excel (manual tests)
  + **james** branch has 24 unit tests (new test.py file)
* **Run unit tests on IDE or GitHub Actions in the README.md:**
  + **IDE:** *python test.py*
  + **GitHubActions:** Click on Actions→Workflow (any)→ workflow\_dispatch event trigger of “Run Workflow” → Run in main.
* Report bugs or suggestions via [Issues](https://github.com/javonpayne100/CMSC495Capstone/issues)

**Contact the Team**

🧐Open to contributions, collaborations, or licensing inquiries? Connect with us via GitHub:

* README.md file: <https://github.com/javonpayne100/CMSC495Capstone#authors--team>
* Thank you for reading our User Guide! - Python Game Hub Authors 😎

# **References:**

Amazon Web Services. (n.d.). *What is unit testing?* Amazon Web Services, Inc.<https://aws.amazon.com/what-is/unit-testing/>

Devaraj, K. (2024, December 19). 6 key phases of testing in software testing. *Testsigma*.<https://testsigma.com/blog/phases-of-testing/>

Doe, J., & Lee, A. (2020). Creating effective contribution reports in software projects: Guidelines and strategies. *IEEE Software Engineering Journal*, 45(6), 102–115.<https://doi.org/10.1109/SEJ.2020.3082548>

Erickson, J. (n.d.). *Algorithms*. University of Illinois at Urbana-Champaign. Online textbook available.

GeeksforGeeks. (n.d.). *Software Development Life Cycle (SDLC)*.<https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/>

GeeksforGeeks. (2024). *Control Flow Software Testing*.<https://www.geeksforgeeks.org/control-flow-software-testing/>

GeeksforGeeks. (2024). *Software engineering: Differences between coupling and cohesion*.<https://www.geeksforgeeks.org/software-engineering-differences-between-coupling-and-cohesion/>

GeeksforGeeks. (2024). *Difference between Function-Oriented Design and Object-Oriented Design*.<https://www.geeksforgeeks.org/difference-between-function-oriented-design-and-object-oriented-design/>

GeeksforGeeks. (2024). *Object-Oriented Design (OOP)*.<https://www.geeksforgeeks.org/oops-object-oriented-design/>

GeeksforGeeks. (2024). *Functions decomposition in software engineering*.<https://www.geeksforgeeks.org/functions-decomposition-in-software-engineering/>

GeeksforGeeks. (2024). *What is a Project Plan*.<https://www.geeksforgeeks.org/project-plan/>

Gehry, F. (Producer). (2019, April 10). *How to make your design stand out* [Video]. YouTube.<https://www.youtube.com/watch?v=Va64slVDpI4>

Git. (n.d.). *Git user manual*.<https://git-scm.com/docs/user-manual>

Jones, M., & Davis, L. (2019). Effective project design planning in software engineering: Best practices and strategies. *IEEE Software Engineering Conference Proceedings*, 31(2), 95–110.<https://doi.org/10.1109/SEConf.2019.2787093>

Khan, S., & Patel, A. (2018). Designing and implementing a centralized game hub for multiplayer platforms. *IEEE Transactions on Game Development*, 39(1), 77–89.<https://doi.org/10.1109/TGD.2018.2876062>

Kleinberg, J., & Tardos, É. (2006). *Algorithm design*. Pearson Education.

LeetCode. (n.d.). *LeetCode - The world's leading online programming learning platform*. LeetCode. Retrieved April 12, 2025, from<https://leetcode.com/>

Lehman, E., Leighton, F. T., & Meyer, A. R. (2021). *Mathematics for computer science*. LibreTexts, MIT, & UC Davis.

Liang, Y. D. (2023, May 23). *Introduction to Java programming and data structures* (13th ed.). Pearson. Online textbook available. (n.d.)

Mount, D. (n.d.). *Algorithms [Lectures CMSC 251]*. University of Maryland Global Campus.

Nievergelt, J. (n.d.). *Algorithms and data structures*. University of Maryland Global Campus.

Smith, J., & Taylor, R. (2021). Using Git for version control in Python-based game development. *IEEE Transactions on Software Engineering*, 47(3), 215–230.<https://doi.org/10.1109/TSE.2021.3056789>

Solomon, J. (2015). *Numerical algorithms*. MIT.

Stellman, A., & Greene, J. (2005). Software project planning. In *Applied software project management* (Part I, Chapters 5, 6, 8, 9, 10, 11). O'Reilly Media. [https://go.oreilly.com/umgc/https://learning.oreilly.com/library/view/applied-software-project/0596009488/ch08.html](https://go.oreilly.com/umgc/https:/learning.oreilly.com/library/view/applied-software-project/0596009488/ch08.html)

Sweigart, A. (n.d.). *Making games with Python & Pygame*.<https://inventwithpython.com/makinggames.pdf>

Tsui, F., Karam, O., & Bernal, B. (2014). *Essentials of software engineering* (3rd ed.). Jones and Bartlett Learning.<https://library-books24x7-com.ezproxy.umgc.edu/toc.aspx?site=VGX8U&bookID=51648>

University of Maryland Global Campus. (n.d.). *Sample User Guide*.https://leocontent.umgc.edu/content/dam/course-content/tus/cmsc/cmsc-495/document/CMSC%20495%20Sample%20Project%20User%20Guide%20(May%202020).pdf?ou=1248245

Unity Technologies. (2020, September 3). *Testing and quality assurance tips for Unity projects*.<https://unity.com/how-to/testing-and-quality-assurance-tips-unity-projects>

Winters, T., Manshreck, T., & Wright, H. (2020). Software engineering at Google (Chs. 18–19). O'Reilly Media.<https://learning.oreilly.com/library/view/software-engineering-at/9781492082781/ch18.html>

# **Glossary of terms:**

| **Term** | **Definition** |
| --- | --- |
| **AI (Artificial Intelligence)** | Simulates intelligent decision-making in games like Tic-Tac-Toe using algorithms such as Minimax. |
| **Breakout** | Arcade-style mini-game in which players break blocks using a ball and paddle. |
| **Command Prompt / Terminal** | Interface where users enter commands to install dependencies or launch the application. |
| **Dependencies** | Required Python libraries (e.g., pygame, numpy, pyautogui) for running the Game Hub. |
| **Draw.IO** | External tool for viewing interactive UI flowcharts included in the project. |
| **Exit Option** | Menu feature that closes the Python Game Hub application. |
| **Fork (GitHub)** | GitHub action used to create a personal copy of a repository for edits or contributions. |
| **Git Bash** | Terminal alternative used for running Git commands on Windows. |
| **GitHub** | Hosting platform for the Game Hub source code and collaboration. |
| **IDE (Integrated Development Environment)** | Software (e.g., PyCharm) for writing, editing, and executing Python code. |
| **Information Button** | Main Menu option that opens the online README.md via browser. |
| **Installation** | Process of downloading, extracting, or cloning the Game Hub to run it locally. |
| **Main Menu** | The application’s starting screen with buttons to launch games or access the GitHub link. |
| **Media Folder** | Directory containing sound files required for audio in games. |
| **Minimax Algorithm** | AI algorithm used to make optimal decisions in Tic-Tac-Toe. |
| **Module** | Self-contained feature or game (Tic-Tac-Toe, Trivia, Breakout). |
| **Mouse Navigation** | Interaction method for selecting Main Menu options or placing moves. |
| **Pygame** | Python library used to create the graphical interface and run the games. |
| **PyCharm** | A popular IDE for Python development. |
| **README.md** | A GitHub markdown file containing project documentation and test instructions. |
| **Screen Resolution** | Required display size for optimal viewing (600x400 or higher). |
| **System Requirements** | Hardware and software criteria needed to run the Game Hub. |
| **Test.py** | File used for running unit tests to validate functionality. |
| **Timer (Trivia)** | Countdown clock that changes color based on urgency during Trivia questions. |
| **Trivia** | A mini-game that tests players with timed, multiple-choice questions. |
| **Two-Player Mode** | Trivia game feature allowing competition between two users. |
| **UI (User Interface)** | Graphical layout including menus, buttons, and in-game visuals. |
| **Unittest** | Python module used for writing and running automated tests. |
| **Unit Tests** | Scripts that verify the behavior of the software components. |
| **Workflow Dispatch** | GitHub Actions trigger used to run test workflows manually. |
| **Zip File** | Compressed archive that can be downloaded and extracted without needing Git. |

# **Index**

| **Key Terms** | **Page Number(s)** |
| --- | --- |
| AI / Minimax Algorithm | 10, 12 |
| Audio Troubleshooting | 14 |
| Breakout Game | 10, 12 |
| Cloning from GitHub | 6 |
| Command Prompt / Terminal | 6, 7 |
| Controls Overview | 5, 10 |
| Dependencies | 6, 7 |
| Draw.IO / UI Flow | 8 |
| Exit Option | 10, 12 |
| Forking Repository | 16 |
| Game Hub Overview | 3 |
| Getting Started | 5 |
| GitHub Links / README.md | 5, 6, 10, 16 |
| IDE Setup (PyCharm) | 6, 7 |
| Information Button | 5, 10 |
| Installation Instructions | 6 |
| Main Menu | 5, 8, 10 |
| Media Folder | 14 |
| Mouse Navigation | 5, 8, 10 |
| Playing Games | 10, 12 |
| PyCharm | 6 |
| Pygame | 3, 6, 7 |
| Purpose of User Guide | 3 |
| Quick Navigation Tips | 5 |
| Scoring Systems | 10, 12 |
| Screen Resolution | 7 |
| System Requirements | 7 |
| Target Audience | 4 |
| Test.py File | 17 |
| Timer Function (Trivia) | 10, 12 |
| Troubleshooting | 14 |
| Trivia Game | 10, 12 |
| Two-Player Mode | 10, 12 |
| Unit Testing | 17 |
| UI Overview | 8 |
| Unittest Library | 17 |
| Workflow Dispatch | 17 |
| Zip File Download | 6 |