Python Game Hub Presentation

- ➢ Victoria Lee
- ➢ James Mutry
- ➢ Todasha Foster
- ➢ Dajin Chung
- ➢ Javon Payne
- ➢ Oluwatumitumu Ipayne

CMSC 495 Capstone Project:

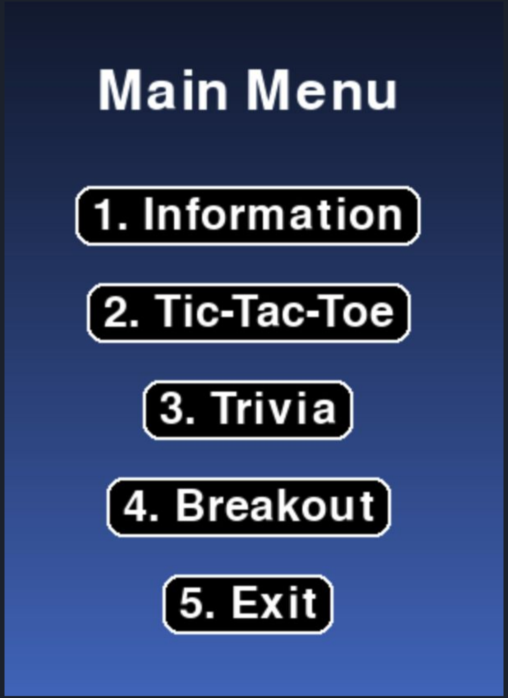
- ❖ (Group 1)
- ❖ 5/3/2025

Introduction

Welcome to the Python Game Hub – a multi-game application built in Python using Pygame

Centralized interface featuring:

- Tic-Tac-Toe (with AI)
- Trivia (with timed responses)
- Breakout (arcade-style)
- Focused on modular design, replayability, and user-friendly navigation
- Developed through collaboration, GitHub version control, and iterative planning



Process Overview

For our Python Game Hub, we built a multi-game application using Python and Pygame, featuring Tic Tac Toe, Trivia, and Breakout, all connected through a central Main Menu.
We used modular coding, GitHub for version control, and GitHub Actions for automated testing.

Our main tools were Python 3.9+, PyCharm, VSCode, GitHub, and Excel for manual testing.

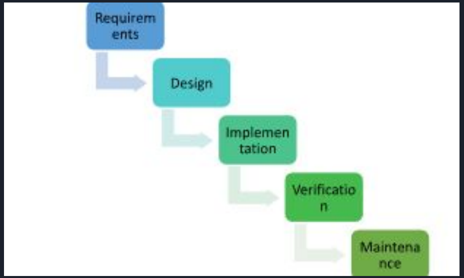
Each teammate contributed:

- **James Mutry** developed unit tests for all modules and manual testing.
- **Victoria Lee** handled documentation, UI improvements, GitHub Actions, and debugging support.
- **Todasha Foster** built the Main Menu, worked on Breakout and Trivia improvements.
- **Javon Payne** helped create the Main Menu and developed the Trivia game.
- **Oluwatumininu Ipaye** built the Breakout game with full physics and scoring.
- **Dajin Chung** developed the Tic Tac Toe game, including the AI logic, and assisted debugging for Trivia and Breakout.

Strong collaboration and clear task ownership helped us create a polished, fully tested application.

Project Plans

Figure: This image below represents the plans for our team using the SDLC waterfall method



- Project Plans:
  - Unit 2: Project Plan
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
  - Unit 3:
    Project Design Plan
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]

- Other Plans:
  - Unit 4: Phase 1 Source Report Plan
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
  - Unit 5: Testing Plan
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
  - Unit 6: Phase 2 Source Report Plan
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
  - Unit 7: User Guide
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
  - Unit 8: Deployment Report
    - [Google Drive (Docx) Link]

Requirements and System Specification

Functional Requirements

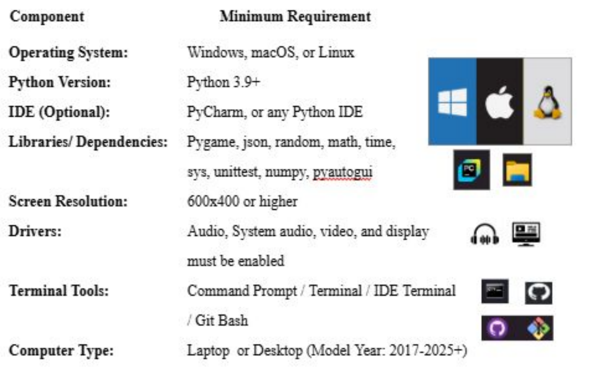
1. Main Menu Navigation:
   Users will be able to easily navigate the main menu and select any game to play.
2. Session Control:
   Users will be able to restart the game, return to the main menu, or exit after each session.
3. Outcome Tracking:
   The system must track and display game outcomes, including winners, losers, ties, scores, and remaining lives.

Non-Functional Requirements

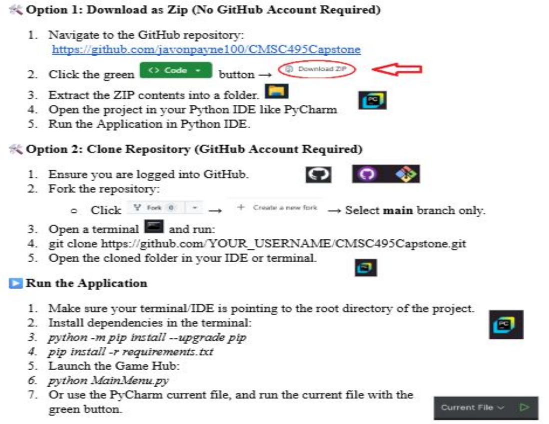
1. Performance:
   Games must load within 3 seconds and run smoothly without noticeable lag.
2. Usability:
   The interface should be intuitive and user-friendly, accessible to users of all ages.
3. Reliability:
   The system must handle all expected inputs and edge cases reliably, with no crashes.

[Functional/Non-Functional Requirements Link Here]

- **User Guide:** [Google Drive (Docx) Link] [GitHub (doc) Link]
- **Unit 6:** [Google Drive (Docx) Link] [GitHub (doc) Link]
- Note: Please, Refer to User Guide / Unit 6 to help set up System Requirements and System! (Images to the Right!)



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+)



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

1. Navigate to the GitHub repository:
   https://github.com/javonpayne100/CMSC495Capstone
2. Click the green <> Code button → Download ZIP
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 Fork 0 → + Create a new fork → 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.

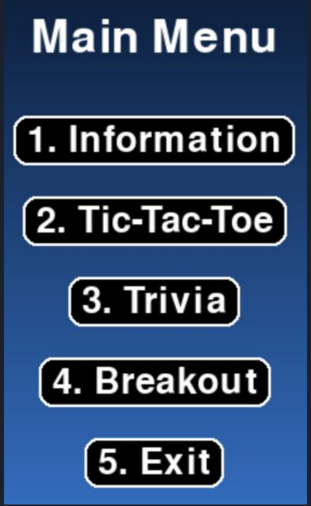
Run the Application

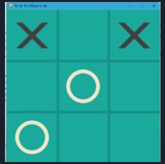
1. Make sure your terminal/IDE is pointing to the root directory of the project.
2. Install dependencies in the terminal:
3. *python -m pip install --upgrade pip*
4. *pip install -r requirements.txt*
5. Launch the Game Hub:
6. *python MainMenu.py*
7. Or use the PyCharm current file, and run the current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

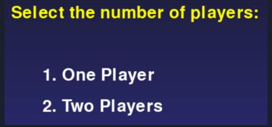
Project Design and User Guide

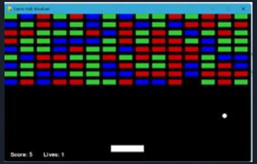
The Python Game Hub utilizes the Pygame library and is a desktop-based application.

- It features three simple games and an information:
  - Tic-Tac-Toe, Trivia, and Breakout (offline)
  - Information (online, readme.md on GitHub Repo.)
- The major idea behind the project is its design that requires no installation or account setup, allowing it to be used in shared spaces such as schools and libraries.
- We aimed for a diverse audience during the project development.
- The low maintenance design allows for bug fixes and error checking without interrupting the user experience.
- The link below is the user guide for further information:
  - [User Guide CMSC 495 Capstone Link Here]





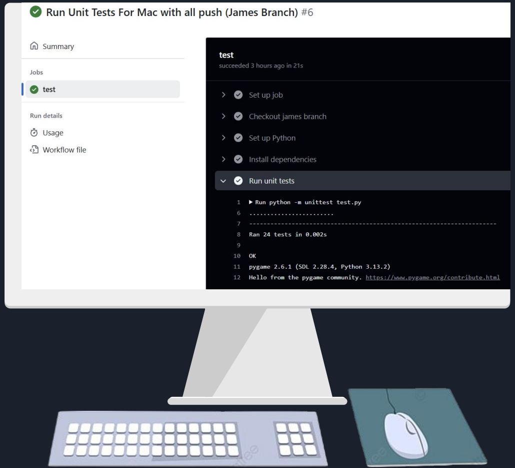




Project Evaluation, Test Plan, and Results:

Finalized manual and automated test plans (includes GitHub Actions),
we were able to identify and fix all major issues before deployment and passed the tests.

- **Unit 4: Phase 1 Source Report Plan**
- **Tests:** 13 unit tests (old test.py, main branch) – no GitHub Actions.
- **Overview:** Covers early development and core game implementations (Mainmenu, TicTacToe, Breakout, Trivia).
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
- **Unit 5: Testing Plan**
- **Tests:** 20 manual test cases (Excel), 24 unit tests (new test.py, James branch), 6 GitHub Actions.
- **Overview:** Full validation of features and behavior, including Trivia.py and Questions.json accuracy.
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
- **Unit 6: Phase 2 Source Report Plan**
- **Tests:** Same as Unit 5, plus 2 GitHub Actions to verify all core files exist for deployment.
- **Overview:** Final features, bug fixes, test reruns, and full documentation (README, comments).
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
- **Unit 7: User Guide**
- **Overview:** End-user instructions for setup and gameplay. Includes game rules, navigation, and GitHub links.
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link]
- **Unit 8: Deployment Report**
- **Overview:** Final packaging of all files, documentation, and submission materials. Includes peer review prep.
- **Presentation:** Google Slides video/ppt summarizing all phases and key outcomes.
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link] (Check it at the GitHub repository doc folder!)
    - [Google Slides Link] (Check it at the GitHub repository doc folder or the Unit 8 discussion!)
      [PythonGameHub GitHub Repo Link]

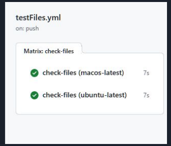

Run Unit Tests For Mac with all push (James Branch) #6

test
succeeded 3 hours ago in 21s

- Set up job
- Checkout james branch
- Set up Python
- Install dependencies
- Run unit tests

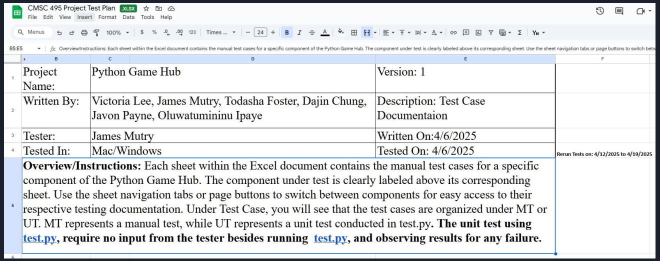
```
▶ Run python -m unittest test.py
........................
----------------------------------------------------------------------
Ran 24 tests in 0.002s

OK
pygame 2.6.1 (SDL 2.28.4, Python 3.13.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
```


testFiles.yml
on: push

Matrix: check-files

- check-files (macos-latest) 7s
- check-files (ubuntu-latest) 7s


CMSC 495 Project Test Plan .XLSX

File Edit View Insert Format Data Tools Help
Menus 100% $ % .0 .00 123 Times ... 24

B5:E5 fx Overview/Instructions: Each sheet within the Excel document contains the manual test cases for a specific component of the Python Game Hub. The component under test is clearly labeled above its corresponding sheet. Use the sheet navigation tabs or page buttons to switch betw

Project Name:	Python Game Hub	Version: 1	
Written By:	Victoria Lee, James Mutry, Todasha Foster, Dajin Chung, Javon Payne, Oluwatumininu Ipaye	Description: Test Case Documentation	
Tester:	James Mutry	Written On:4/6/2025	
Tested In:	Mac/Windows	Tested On: 4/6/2025	Rerun Tests on: 4/12/2025 to 4/19/2025

Overview/Instructions: Each sheet within the Excel document contains the manual test cases for a specific component of the Python Game Hub. The component under test is clearly labeled above its corresponding sheet. Use the sheet navigation tabs or page buttons to switch between components for easy access to their respective testing documentation. Under Test Case, you will see that the test cases are organized under MT or UT. MT represents a manual test, while UT represents a unit test conducted in test.py. **The unit test using test.py, require no input from the tester besides running test.py, and observing results for any failure.**

Design and Alternate Designs:

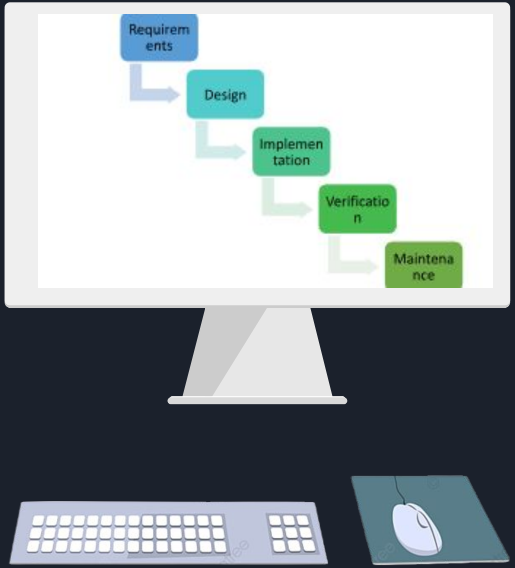
Our Python Game Hub followed a modular, object-oriented design using the Waterfall model.

Phase 1 established the core architecture—Main Menu, Tic-Tac-Toe, Trivia, and Breakout—along with early unit tests and GitHub version control.

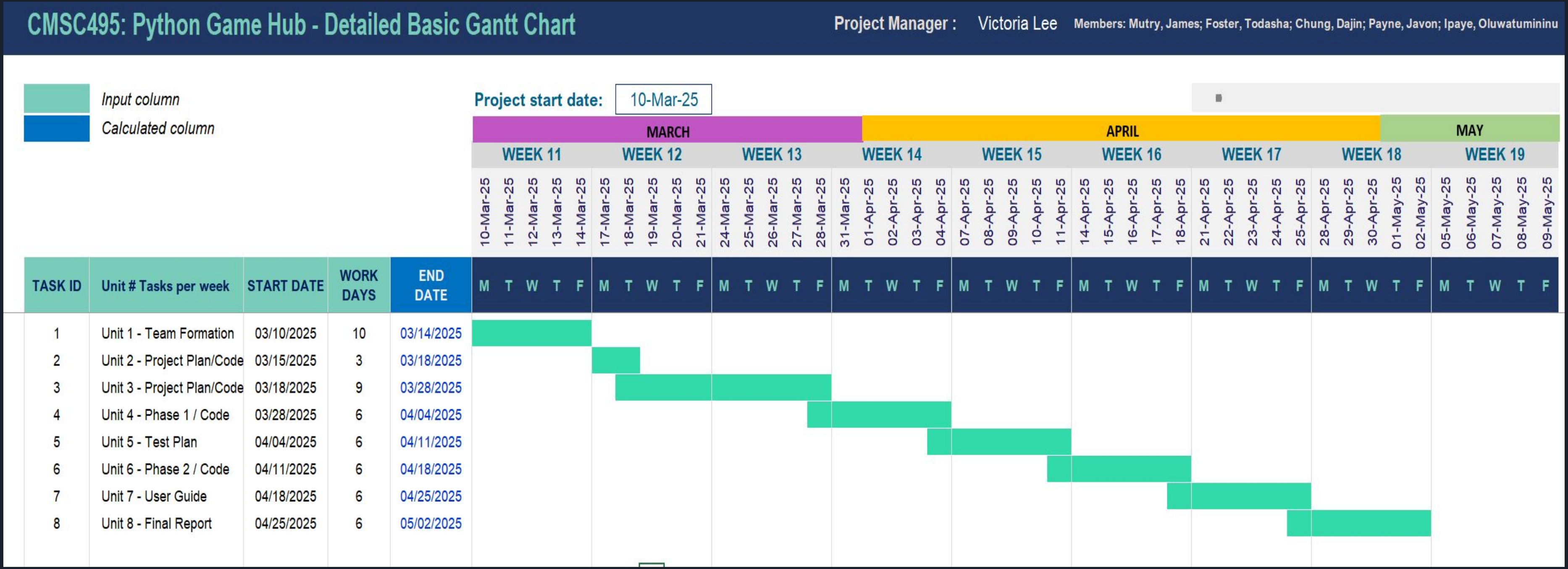
Phase 2 finalized the application with 24+ unit tests, GitHub Actions CI/CD, full manual testing, and optimized gameplay and UI.

All features were validated for performance, usability, and reliability.

Phase 1 Report
Phase 2 Report



Project timeline Image 1



CMSC495: Python Game Hub - Detailed Basic Gantt Chart

Project Manager : Victoria Lee

Members: Mutry, James; Foster, Todasha; Chung, Dajin; Payne, Javon; Ipaye, Oluwatumininu

🟩 *Input column*

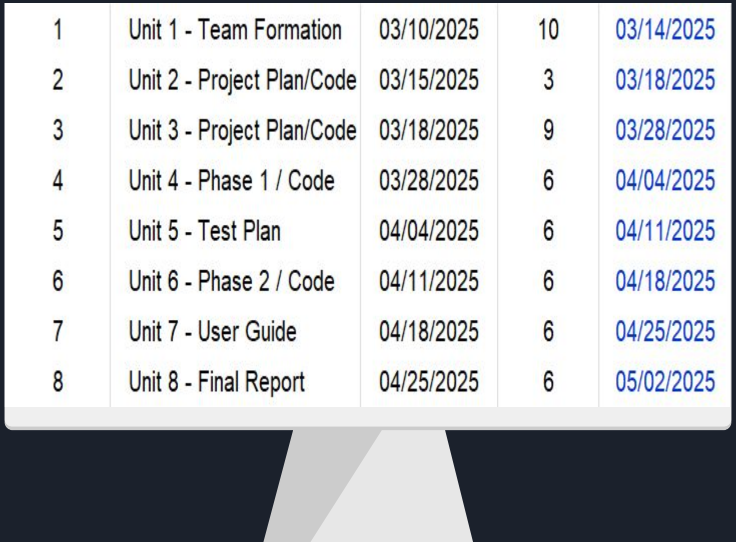
🟦 *Calculated column*

Project start date: 10-Mar-25

TASK ID	Unit # Tasks per week	START DATE	WORK DAYS	END DATE
1	Unit 1 - Team Formation	03/10/2025	10	03/14/2025
2	Unit 2 - Project Plan/Code	03/15/2025	3	03/18/2025
3	Unit 3 - Project Plan/Code	03/18/2025	9	03/28/2025
4	Unit 4 - Phase 1 / Code	03/28/2025	6	04/04/2025
5	Unit 5 - Test Plan	04/04/2025	6	04/11/2025
6	Unit 6 - Phase 2 / Code	04/11/2025	6	04/18/2025
7	Unit 7 - User Guide	04/18/2025	6	04/25/2025
8	Unit 8 - Final Report	04/25/2025	6	05/02/2025

Development History (Project Timeline) and Deployment:

Timeline for Project:



1	Unit 1 - Team Formation	03/10/2025	10	03/14/2025
2	Unit 2 - Project Plan/Code	03/15/2025	3	03/18/2025
3	Unit 3 - Project Plan/Code	03/18/2025	9	03/28/2025
4	Unit 4 - Phase 1 / Code	03/28/2025	6	04/04/2025
5	Unit 5 - Test Plan	04/04/2025	6	04/11/2025
6	Unit 6 - Phase 2 / Code	04/11/2025	6	04/18/2025
7	Unit 7 - User Guide	04/18/2025	6	04/25/2025
8	Unit 8 - Final Report	04/25/2025	6	05/02/2025

⚠️Python Game Hub must be Publicly Accessed on GitHub:

- **GitHub Repository:**
- [CMSC495Capstone-Python Game Hub]
  - Read the README.md OR User Guide

Project Timeline (based on updates[Phases 1 & 2]):

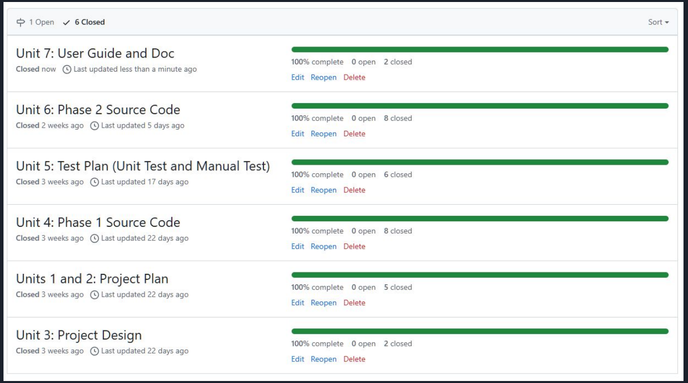
- **Unit 2: Project Plan**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 3: Project Design Plan**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 4: Phase 1 Source Report Plan**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 5: Testing Plan**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 6: Phase 2 Source Report Plan**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 7: User Guide**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- **Unit 8: Deployment Report**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link][Google Slides Link]
  - (Refer to GitHub Repo for those links)
  - [PythonGameHub GitHub Repo Link]

Project timeline Expansion Explanation

Refer to Unit 4, Unit 6, and Unit 8 to compare the project timelines, job contributions, Git (Version Control), Testing, and Changes to the Project Management sections.

- Unit 6: Phase 2 Source Report Plan
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- Unit 7: User Guide
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link]
- Unit 8: Deployment Report
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link and Slides]
  - (At GitHub repository “doc”)

  - [PythonGameHub GitHub Repo Link]



1 Open ✓ 6 Closed Sort

Unit 7: User Guide and Doc
Closed now Last updated less than a minute ago
100% complete 0 open 2 closed
Edit Reopen Delete

Unit 6: Phase 2 Source Code
Closed 2 weeks ago Last updated 5 days ago
100% complete 0 open 8 closed
Edit Reopen Delete

Unit 5: Test Plan (Unit Test and Manual Test)
Closed 3 weeks ago Last updated 17 days ago
100% complete 0 open 6 closed
Edit Reopen Delete

Unit 4: Phase 1 Source Code
Closed 3 weeks ago Last updated 22 days ago
100% complete 0 open 8 closed
Edit Reopen Delete

Units 1 and 2: Project Plan
Closed 3 weeks ago Last updated 22 days ago
100% complete 0 open 5 closed
Edit Reopen Delete

Unit 3: Project Design
Closed 3 weeks ago Last updated 22 days ago
100% complete 0 open 2 closed
Edit Reopen Delete

It is crucial to view the APPENDIX sections:

- The testing and Git version control processes are further detailed in APPENDIX D and APPENDIX E. The Milestones and Gantt chart updates with the Contribution Report are in APPENDIX A, APPENDIX B, and APPENDIX E.
- A shows the overall timeline due dates and how our progress compares with units 1-8. B shows the breakdown of each task/milestones completed and costs.
- APPENDIX C is referring to the delays and emergencies so far to push back the code.
- APPENDIX E shows the breakdown of the Git issues/milestones completed for that week (screenshots).

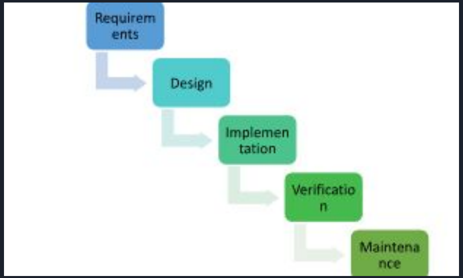


Figure: This image below represents the project timeline plans for our team using the SDLC waterfall method

Discussion

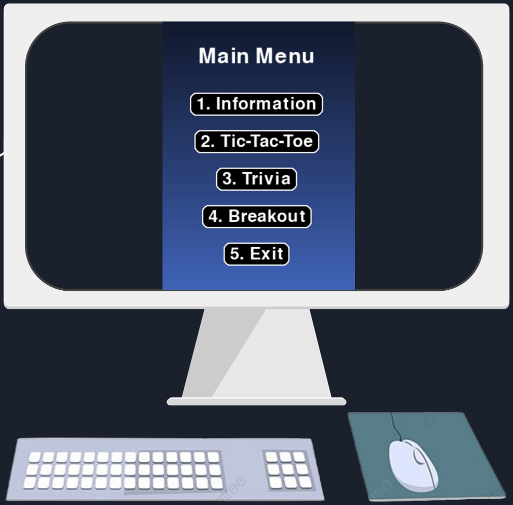
Throughout this project, we learned the value of strong modular design, clear documentation, and regular testing. One of our strengths was creating a smooth and unified player experience across different games.

However, some limitations included challenges with cross-platform testing and limited mobile compatibility.

For the future, we suggest adding database support, expanding automated testing, and improving UI responsiveness. Overall, this project taught us that solid planning, communication, and teamwork are essential to delivering a polished and maintainable software product.

Conclusion

- Successfully delivered a modular Python game hub with three mini-games
- Followed the SDLC Waterfall Model across eight project units
- Combined automated and manual testing to ensure a bug-free deployment
- Used GitHub for version control and implemented CI/CD via GitHu Actions
- Learned collaboration, version control, task ownership, and technical writing
- Ready for public GitHub release and future scalability



Acknowledgements

Victoria Lee (GitHub: VictoriaRaven): As a Project Manager / Technical Writer / Tester (GitHub Actions / Repo), I want to express my deepest gratitude to my amazing team for their unwavering dedication and hard work throughout this project. The way we came together, blending our skills, strengths, commitment,and teamwork like in the *Avengers*, Marvel movie, made this experience truly special (Whedon, 2012). I also want to extend a heartfelt thank you to my peers and family, whose suggestions helped shape the final product. Your support, both on and off the project, was invaluable and truly appreciated. Thank you and as always, “Not all treasure is silver and gold, mate.” -Captain Jack Sparrow (Pirates of the Caribbean) (Verbinski, 2003).

James Mutry (GitHub: jamesmutry): As a Technical Writer / Tester (Unit Tests & Manual Tests), I would like to take this opportunity to sincerely thank my entire team for their dedication, hard work, and collaboration throughout the Python Game Hub project. Each member brought their own unique skills and strengths, contributing to the success of this challenging and rewarding experience. From initial planning to final deployment, our team consistently supported one another, tackled obstacles together, and maintained a strong commitment to producing a polished and functional final product. I am especially grateful for the positive attitude, adaptability, and professionalism that everyone demonstrated during each phase of the project. Working alongside such a talented and motivated group made this capstone experience not only educational but truly enjoyable. I am proud of what we accomplished together and thankful to have had the opportunity to collaborate with such an outstanding team.

Javon Payne (GitHub: javonpayne100): As a Git Lead (.py files) / Developer (MainMenu & Trivia) / Sub Tester), I want to give a huge thanks to my team for bringing me on board with open arms. It’s a blessing to have a team with diverse skills, restless drive, and great character. This experience has definitely taught me a lot and this type of camaraderie should be the standard of many development teams. I want to especially thank Victoria; her organization skills and leadership is unmatched. Congratulations everyone, graduation is here! “May the force be with you.” - Yoda

Acknowledgements (Continued)

Todasha Foster (GitHub: DayFoster123): As a Developer (Breakout) / Sub Tester, I would like to express my sincere appreciation to my team members for their energy, creativity, and collaboration throughout this project. Contributing as a developer was a rewarding experience that allowed me to strengthen my skills and grow professionally. I am proud of the challenges we overcame together and the final products we created. This achievement reflects the passion, effort, and dedication of every member of the team. "I never dreamed about success. I worked for it." – Estée Lauder

Dajin Chung (GitHub: dajinchung): As a Developer (TicTacToe) / Sub Tester, I'd like to thank my entire team for their hard work and collaboration throughout this project. Each member contributed their strengths, from development and testing to design and documentation, to bring this project to life. A special thanks to Victoria, who truly stepped up as project manager. Her organization, communication, and consistent support helped keep our team focused and on track. We would not have reached this level of polish and completion without her leadership. Everybody, thank you for all your hard work. As a wise ogre once said, "That'll do Donkey. That'll do." - Shrek

Oluwatumininu Ipaye (GitHub: tumiwiththewave): As a Developer (Breakout) / Sub Tester, I want to express my deepest gratitude to the entire team for their hard work and resilience to get this project done. I have been able to learn a lot from each single member and effectively learnt how to work in a group setting such as this. My skills have been boosted and challenged at the same time, which shows that being a coder/programmer is a never ending lesson. I would like to specially thank Victoria Lee, as she was the backbone of this entire project and ensured everything went smoothly and without any hitches. I am proud of what we were able to accomplish in a short period, and wish you all the best ahead. *The sky's the limit.*

References (Key parts only)

Amazon Web Services. (n.d.). *What is unit testing?* Amazon Web Services, Inc. https://aws.amazon.com/what-is/unit-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). *Object-Oriented Design (OOP)*. https://www.geeksforgeeks.org/oops-object-oriented-design/

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

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

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.

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

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

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

- **Unit 8: Deployment Report**
    - [Google Drive (Docx) Link]
    - [GitHub (doc) Link] (Check it at the GitHub repository doc folder!)
    - [Google Slides Link] (Check it at the GitHub repository doc folder or the Unit 8 discussion!)
    - [PythonGameHub GitHub Repo Link]

Appendix

- **Unit 8: Deployment Report**
  - [Google Drive (Docx) Link]
  - [GitHub (doc) Link] (Check it at the GitHub repository doc folder!)
  - [Google Slides Link] (Check it at the GitHub repository doc folder or the Unit 8 discussion!)
  - [PythonGameHub GitHub Repo Link]

-Thank you!