

Team Project One: Battleship

Language: Python

Platform: Terminal

Estimated Man Hours: 25 Hours

Justification:

We broke down the project into a few main goals:

Board Creation (2 hours)

Objective: Implement the foundational game board, including grid setup and initialization.

Reasoning: Creating a basic grid structure involves setting up a 2D list and ensuring correct indexing for game actions. Given the straightforward nature of this task and similar implementations from past projects, we estimate 2 hours for this component.

Ship Creation (2 hours)

Objective: Define and create the various ship types and sizes.

Reasoning: This involves coding the attributes and behaviors for different ship types. We anticipate this task to take 2 hours based on our experience with object-oriented programming and previous projects that required similar object definitions.

Placing Ships (2 hours)

Objective: Implement the logic for placing ships on the board, ensuring they adhere to game rules.

Reasoning: Ensuring that ships are placed correctly and do not overlap involves validating input and implementing placement rules. This task is estimated at 2 hours based on the complexity of placement logic and our familiarity with similar tasks in past projects.

Destroying Ships (2 hours)

Objective: Develop the functionality to handle ship destruction when hit.

Reasoning: This requires implementing logic to track and update the state of ships when they are hit. Given that this involves state management and updating game status, we estimate 2 hours based on previous experience with state-dependent logic in games.

Hit/Miss Detection (2 hours)

Objective: Implement the logic to determine whether a shot hits or misses a ship.

Reasoning: Accurate hit/miss detection is critical and involves checking coordinates against ship positions. Our estimate of 2 hours reflects the need to implement and test this functionality thoroughly to ensure accuracy.

Turns/Game Logic (15 hours)

Objective: Develop the turn-taking mechanism, game flow, and overall game logic.

Reasoning: This is the most complex part of the project, requiring integration of various components, management of game state, and ensuring smooth gameplay. Given the complexity of managing player turns, implementing game rules, and ensuring robust error handling, we have allocated 15 hours. This estimate is based on our understanding of the interactions between components and the experience from similar projects, where game logic and flow were time-consuming due to the need for thorough testing and refinement.

Overall Time Justification:

The total estimated man-hours of 25 hours have been derived by analyzing the complexity of each task and considering our previous experiences with similar projects. We have allocated more time to the game logic and turn management due to their complexity and critical role in ensuring a seamless player experience. Each component has been carefully assessed to ensure that the estimated time is realistic and achievable, allowing us to deliver a functional and engaging Battleship game.

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|--|-----------------------------|---|------------------------------|
| Hamza Jalil | 09/02/2024 | 1. Do background research and prepare for meeting. 2. Update meeting notes. | 1. 1 hour 2. 10 Minutes. | 1. Dld research on Py game methods, web application method (java script, socket.io, html & css, and some c). 2. Updated notes. | 1. 2 Hours. 2. 5 Minutes. |
| Timo Aranjó | | | | | |
| Lingfeng Li | | Write the logic for add_ship and add_shoot function with Bob. | 1 hour | Add the BOb、add_ship and add_shoot part in project,add a function called check_sunk, get the __init__ set the board size. | 1 hour |
| Isaac Mohabbat | | | | | |
| Harry Wang | | Researched on battleship implementation | 30 minutes | Researched on battleship implementation | 42 minutes |

Team Work Section:

Discord Meeting (6:10 pm - 6:44 pm)

- No Questions regarding the project objectives.
- Possible web application UI but using pygame will be the initial position.
- Need to do independent research before splitting responsibilities.
- Concerns on how to adequately present the board to both players without cheating.
- The goal is to complete the project by 09/13/2024 for 2 days of debug if needed.

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/03/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/04/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/05/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|--------------------------|----------------|-------------------|---------------|
| Hamza Jalil | 09/06/2024 | 1. Update Meeting notes. | 1. 10 Minutes. | 1. Updated Notes. | 1. 5 Minutes. |
| Timo Aranjó | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Meeting (4:20 pm - 5:16 pm)

- We only need 1 device (not a 2 device implementation)
- Did an estimation of work time.
- Use 2D array for hit (neg.), miss (-6), sunk (all # are neg.), and ship places (pos.)..
- Drop ship head and switch from vertical (down) to horizontal (right).
- 3 Main classes
 - Interface - bridges comms. Through player and board.
 - Player - Does the actions on the board for the player.
 - Board - functions fo the board.

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|--|----------------|--|--------------|
| Hamza Jalil | 09/07/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | Researched battleship and python list manipulation | 2 hour | Researched battleship and python list manipulation | 1:35 hour |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/08/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/09/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|--|----------------|---|---------------|
| Hamza Jalil | 09/10/2024 | 1. Update the project log with meeting notes. | 1. 30 Min | 1. Updated all notes for the day. | 1. 48 Minutes |
| Timo Arango | | Create code skeleton and implementation notes | 15 Minutes | Create code skeleton and implementation notes | 30 Minutes |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | Git downloaded the program skeleton. Implemented some of the basic functions | 1. 1 hour | Git downloaded the program skeleton. Implemented some of the basic functions such as get_point, add_point, coord_to_xy functions. | 1. 45 minutes |

Team Work Section:

Meeting (3:00 pm - 3:30 pm)

- Board Logic for out-of-bounds placement using a try-and-catch.
- Using a temp. to store a board's value, checking each position, and if it fails, going back to the temp.
- 2 Classes:
 - Players
 - Interface

Meeting (3:45 pm - 5:10 pm)

- We discussed at length the different possible implementations of HOW to do certain functions.
- Decided to revert back to the basics and understand the constraints of the program (focus on the WHAT) and from there understand the required functions.
 - B.O.B
 - Add_Ship
 1. Check if the head coordinate is out of bounds with B.O.B
 2. Check the Body/Tail is not out of bounds.
 3. Check we do not put the ship on an existing ship
 4. Add ship.
 - Shoot
 1. Check if the coordinate is out of bounds with B.O.B
 2. Check the coordinate has not already been used.
 3. Check if the shot was a hit or miss.
 - IF HIT
 - Mark the coordinate as a hit.
 - Check if there was a Sunk.
 - Check if there is a win/loss with the shot.
 - IF MISS
 - Mark the coordinate as a miss.
 4. Update the Boards

- Get_Point
- Add_Point
- Status
- Start
- Timo will organize the project classes, variables, and functions and describe the following:
 - Classes
 - Respective Variables.
 - Respective Functions
 1. Function Name.
 2. Function Description.
 3. Function Inputs.
 4. Function Calls to other Functions.
 5. Function Outputs.
 6. Pseudo Code.
- Once we understand the program organization and set good coding standards we will split the responsibilities and code for a completion date goal of 09/13/2024

GTA Meeting Section:

Meeting (3:30 pm - 3:45 pm)

- Met with GTA.
- Clarified project objectives and details.
- Need to send an email to GTA and the Professor regarding the programming language. Completed as of 9/13/24
- Spoke about the previous week's accomplishments and the plans moving forward.

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|--|---|--|--|
| Hamza Jalil | 09/11/2024 | 1. Update Player Class. 2. Update Interface Class. 3. Work On logic/Debug. | 1. 4 Hours 2. 4 Hours 3. 2 Hours. | 1. Updated Player Class. 2. Updated Interface Class. 3. 60% working but edge cases and bad input not protected. Also board is not working as expected. | 1. 2 Hours 2. 2 Hours 3. 2 Hours |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/12/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---|-------------------------------------|---|-------------------------------------|
| Hamza Jalil | 09/13/2024 | 1. Debug edge cases and user input. | 1. 1 Hour. | 1. Fixed bad user validation, error handling, and case issues. | 1. 1.5 hours |
| Timo Arango | | Bug Fixing Related to misses showing up incorrectly | 1 Hour | Fixed Bug | 2 Hours and 30 Minutes |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | 1. Clean up the interface and fix the logic of the amount of ships. 2. Fix the logic of check_win 3. Fix duplicate hits issue | 1. 30 min 2. 30 min 3. 15 min | 1. Changed so that board is displayed after every ship placement, as well as to assign player 1 to choose amount of ships so that it is consistent 2. Had a comparison of the hits to the number of ships to | 1. 33 min 2. 21 min 3. 16 min |

| | | | | | |
|------------|--|--|--------|--|------------|
| | | | | determine the win 3. fixed the logic of receive_shot & take_shot so that if the user hits an already hit or sunk spot it warns them and lets them try again | |
| Harry Wang | | Cleaned Up the Interface and made it more readable | 1 hour | Cleaned Up the Interface and made it more readable | 1:04 hours |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---------------|----------------|----------------|--------------|
| Hamza Jalil | 09/14/2024 | | | | |
| Timo Arango | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

Individual Work Section:

| Name | Date | Work Expected | Hours Expected | Work Completed | Actual Hours |
|----------------|------------|---|---|---|--|
| Hamza Jalil | 09/15/2024 | 1. Final Testing 2. Program Documentat ion 3. Upload Project Log on Github | 1. 30 minutes 2. 30 minutes 3. 10 minutes | 1. Completed Final Testing (see chart below). 2. Added Relevant Documentat ion. 3. Uploaded Project Log. | 1. 1 hour 2. 1 hour. 3. 33 minutes |
| Timo Aranjó | | | | | |
| Lingfeng Li | | | | | |
| Isaac Mohabbat | | | | | |
| Harry Wang | | | | | |

Team Work Section:

| | |
|-----------------------------------|-----------------------------|
| Total Time Spent | |
| Name | Hours |
| Hamza Jalil | 13 hours 58 minutes |
| Timo Arango | 3 hours 0 minutes |
| Lingfeng Li | 3 hours 48 minutes |
| Isaac Mohabbat | 6 hours 48 minutes |
| Harry Wang | 1 hour 30 minutes |
| Total | 29 hours 4 minutes. |
| | |
| Team Meetings/GTA Meetings | 3 hours 40 minutes. |
| | |
| Total | 32 hours 44 minutes. |

| Category | Test Description | Pass/Fail | Comments |
|-----------------------------|---|-----------|----------|
| Board Initialization | Verify the board is a 10x10 grid | P | |
| | Ensure columns are labeled A-J and rows 1-10 | P | |
| Number of Ships | Check if the user is prompted for ship quantity (1-5) | P | |
| | Validate input allows a minimum of 1 and a maximum of 5 ships | P | |
| | Invalid input (e.g., 6, -1, non-numeric characters) prompts the user to try again | P | |
| Ship Types and Sizes | If 1 ship: Check if the player receives a single 1x1 ship | P | |
| | If 2 ships: Check if the player receives a 1x1 and a 1x2 ship | P | |
| | If 3 ships: Check if the player receives a 1x1, 1x2, and 1x3 ship | P | |
| | If 4 ships: Check if the player receives a 1x1, 1x2, 1x3, and 1x4 ship | P | |
| | If 5 ships: Check if the player receives a 1x1, 1x2, 1x3, 1x4, and 1x5 ship | P | |
| Ship Placement | Check if players can place ships horizontally and vertically | P | |
| | Validate that ships cannot overlap on the board | P | |
| | Ensure ships are placed within the boundaries of the board | P | |
| | Ensure players can place ships secretly without revealing locations to the other player | P | |
| Taking Turns | Verify that after each shot, the turn switches to the other player | P | |
| | Ensure players can select a grid space to fire at (A1-J10) | P | |
| | Validate the game informs the player whether the shot was a hit or miss | P | |
| | Invalid input (e.g., K1, Z3, 11) should prompt the user to try again | P | |
| Player's View | Ensure the player can see their own board, including where their ships are placed | P | |
| | Verify that the player's board updates with hit or miss results | P | |
| | Check that a player has a board to track shots they've fired, marking hits and misses | P | |
| Destroying a Ship | Ensure that when all cells of a ship have been hit, the ship is marked as destroyed | ? | |

| | | | |
|---------------------------|---|---|--|
| | Verify the same spot cannot be hit twice by the same player | P | |
| | Validate the player is informed when a ship is sunk | P | |
| Game End Condition | Verify the game ends immediately once all ships of one player are sunk | P | |
| | Check if the game announces the winner when all opponent's ships are sunk | P | |
| User Interface | Confirm the game's interface is intuitive and easy to use (no manual required) | P | |
| System Stability | Stress test: Ensure no crashes during extended gameplay | P | |
| | Check for any memory leaks or performance issues during prolonged play | P | |
| Modularity of Code | Verify that the code is logically divided into modules (e.g., separate classes for board, ship, player, etc.) | P | |
| Documentation | Ensure there is a detailed estimate of person-hours in the project repository | P | |
| | Ensure there is a day-by-day accounting of actual person-hours for each team member | P | |
| | Validate the presence of detailed system documentation in the repository folder | P | |
| | Check if the code is well-commented, with prologue comments and line-by-line explanations | P | |
| | Verify that the comments clearly explain the source of any external code (e.g., ChatGPT, StackOverflow) and are written in the team's own words | P | |