Report - Warbound

Rodrigo Severo, Arthur Rabello

December 4, 2024

1 Introduction

Warbound is a turn-based strategy game where players move units on a board to defeat their opponents. The game was developed using Python and the Pygame library.

1.1 Goal/Objective

Warbound is a turn-based strategy game that involves battlefield tactics and army management. The main objective of the game is to defeat the enemy units. Victory is achieved when a player kills the enemy general or annihilates their entire army. This goal requires a combination of strategic planning, tactical movement, and the effective use of available resources and units.

1.2 How to play

This section provides a basic introduction to how to play *Warbound*. For a more detailed tutorial, including the keys to use during the game, refer to the tutorial available in the game.

Initially, each player starts with a set of units on the board, which can be moved each turn. The units vary between infantry, cavalry, and archers, each with its own strengths and weaknesses. Players can adopt different strategies to gain an advantage over their opponents:

- **General Selection:** Before starting a match, each player chooses a general to lead their forces, each with unique abilities that can significantly change the course of the battle. For example, choosing ”Alexander the Great” may strengthen infantry units, while ”Julius Caesar” may improve the movement and attack capabilities of legions. This initial choice is crucial, as the general defines the playstyle and the initial formation of their army.
- **Formation System:** Units can assume various formations, such as ”Phalanx”, ”Shield Wall”, and ”Spread”, which alter their defense and attack attributes. For instance, the ”Phalanx” formation increases attack but reduces defense, ideal for heavy offensives.
- **Unit Orientations:** The orientation of a unit on the battlefield determines its effectiveness in attack and defense. Units facing the enemy directly are more effective in combat, while attacks from the sides or rear can deal more devastating blows.
- **Positioning and Movement:** Utilize the terrain to your advantage by positioning units in strategic locations, such as mountains, which increase defense attributes and can protect more vulnerable units.

2 How to run

Follow the steps below to set up the project in your local environment:

1. Clone the repository:

```
git clone https://github.com/arthurabello/trabalho_lp_A2.git
cd trabalho_lp_A2
```

2. Create and activate a virtual environment (recommended):

```
python3 -m venv venv
source venv/bin/activate
```

3. Install the dependencies:

```
pip install -r requirements.txt
```

4. Run the main script to play Warbound:

```
cd src
python3 main.py
```

3 Project Development

The development of the game focused on three main components: the game board, the units, and the user interface.

3.1 Main Game Logic

- **The Board and the Graph:**

  The game board is the foundation where all interactions take place. It was implemented as a two-dimensional grid, with the size configurable through variables representing the number of columns and rows. Each cell on the board can contain a unit and has specific terrain characteristics that affect unit movement. To manage the movement logic based on terrain costs, a graph was associated with the board. This graph is dynamically updated to reflect the positions of units and the characteristics of the terrain, using Dijkstra's algorithm to calculate movement costs, thus allowing units to move according to their capabilities and the terrain conditions.

- **Units and Their Characteristics:**

  1. Class Hierarchy and Inheritance: The units in the game were implemented following a class hierarchy that derives from a common base class, located in the `base_unit.py` file. This base class manages universal attributes such as position on the board, direction, and player identification. From this class, specific subclasses like `Hoplite` and `Archer` are derived and configured with specific attributes such as attack type, range, etc.

  2. Some Important Unit Attributes:

  – **General_id and has_general**: There is only one general per player in the game. Some units can be commanded by a general, which alters their abilities. If a unit with a general is eliminated, the game ends for that player.

  – **Movement and Attack Range (movement_range and attack_range)**: These attributes define how far a unit can move or attack in a turn, directly influencing positioning tactics on the board.

  – **Terrain**: The type of terrain a unit is positioned on affects its defense and movement. For example, units on mountains receive a defense bonus against melee attacks.

  – **Formation**: A unit's formation affects its attack and defense stats. Changing a unit's formation during the game is conditioned by the unit's remaining movement points; if the movement points are zero, the unit cannot change formation during the current turn. This adds a strategic layer to movement and positioning on the battlefield.

3. General's Movement: The general can only move between adjacent units, reflecting the need for proximity for effective command on the battlefield. This rule adds another strategic layer to the game, as players must position their units carefully to maintain protection for the general.

4. Attack Mechanic: Attacking in *Warbound* is characterized by a distinctive approach, involving a counterattack mechanism. When a unit attacks, it also takes damage from the target, creating a situation where the attack can result in mutual damage. The direction of the attack relative to the orientation of the defending unit (north, south, east, west) also influences the result, with flank or rear attacks being more effective.

- **User Interface:**

  1. Main Menu: The main menu of "Warbound" is the first interface with which the player interacts. This menu includes four main buttons: Play, Tutorial, Options, and Exit. Each button is designed with an attractive visual style and intuitive functionality, making it easy to access the different sections of the game.

  - Play: Starts the general and map selection sequence, leading to the beginning of the match.
  - Tutorial: Opens a series of interactive "pages" that explain the fundamentals of the game, such as moving units, attacking, and using formations, among other aspects.
  - Options: Allows players to adjust settings such as sound volume and other game preferences.
  - Exit: Closes the game.

  2. Tutorial: The *Warbound* tutorial is designed to guide new players through a detailed introduction to the game. It is presented in a book format, where players can navigate between pages that contain information on how to play, move units, attack, and use various strategies. The tutorial is accessible from the main menu and is essential for new players to understand the complex mechanics of *Warbound.*

  3. Options: On the options screen, players can adjust sound and other game settings. This includes turning sound on or off, adjusting the volume, and modifying other preferences to help personalize the gaming experience. These settings are adjusted through toggle buttons and sliders that are easy to use and visually integrated into the game's style.

  4. Game Flow After Selecting 'Play': After clicking "Play", the player is taken to the general selection screen, where they can choose from various historical generals, each with specific bonuses for different types of units. After selecting a general, the game begins with the units positioned according to the chosen general.

  5. In-Game Status Screen: During the game, a status screen provides continuous information about the selected unit and the current turn. This screen shows details such as the unit's health, attack and defense points, attack and movement range. Additionally, information about the terrain, the current formation, and the orientation of the unit are displayed, helping players make smart tactical decisions during the game.

3.2 Challenges Encountered

3.2.1 Game Architecture

Challenge: Modularizing the code and ensuring good separation of responsibilities between the various game components.

Solution: A modular architecture was created with specialized managers. The `GameManager` controls the game loop and initialization, the `CombatManager` is responsible for combat calculations, the `TurnManager` handles the turns, and the `InputHandler` processes user inputs. Each module operates independently but communicates through clearly defined interfaces, making the code easier to maintain and expand.

3.2.2 Core Game Mechanics

Unit Movement

Challenge: Implementing an efficient and realistic system for unit movement, considering terrain and other units.
Solution: In the `BoardGraph` module, the `dijkstras_algorithm` function calculates the shortest paths considering terrain and other units. This allows units to move efficiently, accounting for variable movement costs.

Combat System

Challenge: Creating a balanced and flexible combat system.
Solution: In the `UnitCombatMixin` module, functions like `attack` and `calculate_modifiers` allow for damage calculation considering factors like attack direction, terrain, and leadership bonuses. This keeps combat fair and engaging.

3.2.3 Formation System

Challenge: Allowing easy modification and addition of combat formations.
Solution: The system uses a dictionary in each unit class to define attack and defense modifications, making formation management easier.

3.2.4 Visual Challenges

Challenge: Adapting the game to different screen resolutions.
Solution: The `GameRenderer` class manages adaptive rendering, automatically adjusting graphics to fit different screen sizes, ensuring a consistent visual experience.

3.2.5 Command Input

Challenge: Managing player input efficiently and intuitively.
Solution: The `CommandHandler` class maps keyboard and mouse actions to specific game commands, simplifying user control and interaction.

3.2.6 Menu System

Challenge: Creating intuitive menus and managing smooth transitions between different game states.
Solution: The menu system is managed by `MenuState`, which handles state transitions and user interactions, offering smooth and clear navigation.

3.2.7 Game Balancing

Challenge: Balancing the abilities of units to ensure fair gameplay.
Solution: A modifier system was implemented that adjusts unit stats, allowing dynamic tweaks to maintain game balance.

3.2.8 Asset Management

Challenge: Efficiently managing multimedia assets.
Solution: Centralized asset management through path constants, making updates and maintenance easier.

3.3 Project Organization

The game is organized as follows:

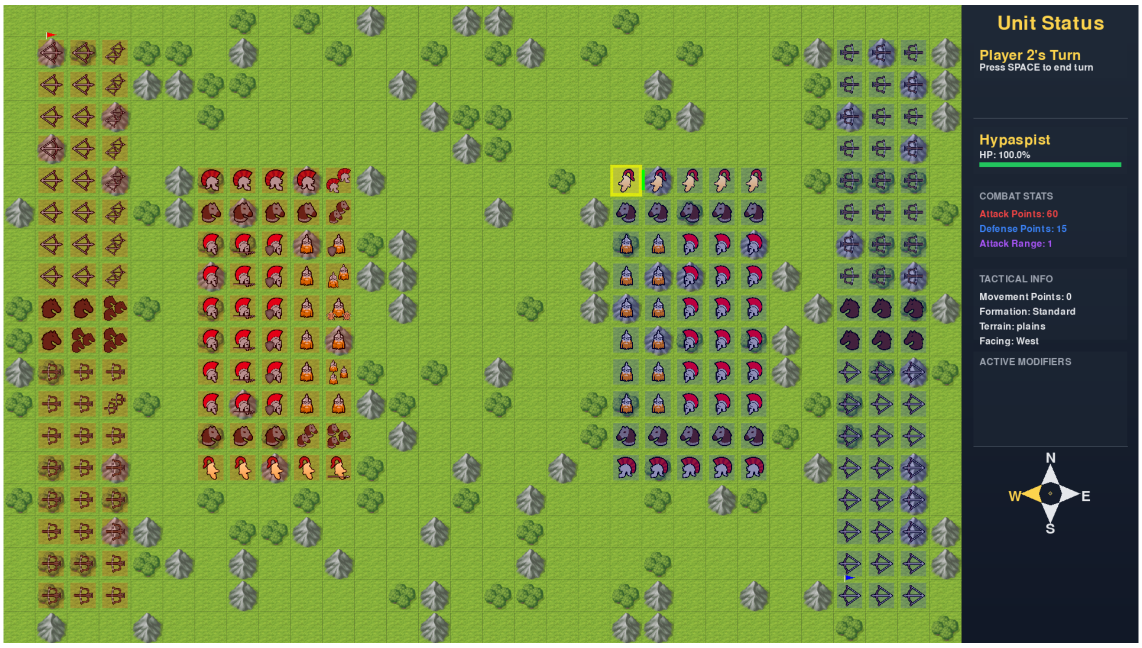
```
trabalho_lp_A2/

  assets/                    # Contains all game assets (images, sounds, etc.)
     sprites/               # Sprites of terrains and units
     sounds/                # Sound effects and background music

  src/                       # Source code folder
     classes/               # Entity classes, such as characters or items
        game/              # Contains the game loop and initialization logic
        menu/              # Contains the game menu logic
        units/             # Contains the game unit logic
        board.py           # Game board logic
        graph.py           # Logic for the graph associated with the game board

     main.py                # Main game file

  tests/                     # Game tests folder
  docs/                      # Game documentation folder
     pdf/                   # pdf documentation, including Report of the game
     tex/                   # .tex files
  .gitignore                 # File for ignoring files in git
  LICENSE                    # Project license
  README.md                  # Project info in markdown format
  requirements.txt           # Python package dependencies
```

4 Design Decisions

The design decisions were made with the players in mind and how they should devise their strategies, focusing on an arcade-style approach, where units are represented by icons that symbolize them, ensuring the game is visually appealing.

4.1 The Game

Warbound was designed to be an elegant game, combining colorful graphical elements with a clear and easy-to-navigate interface. The layout of the game is intuitive, allowing new players to quickly understand the mechanics present in the game, thanks in part to the status screen, which plays a significant role in this.


Unit Status

Player 2's Turn
Press SPACE to end turn

Hypaspist
HP: 100.0%

COMBAT STATS
Attack Points: 60
Defense Points: 15
Attack Range: 1

TACTICAL INFO
Movement Points: 0
Formation: Standard
Terrain: plains
Facing: West

ACTIVE MODIFIERS

Figure 1: Warbound

4.2 Units

In the initial development of the game, the idea of representing the units using detailed full-body sprites was considered. However, this approach had many challenges, particularly due to the density of units that a single piece could represent on the board. Representing multiple units with a single icon would complicate visualization. Additionally, in one of the versions of the game, it was necessary to increase the size of the board, which would result in excessively small icons, negatively affecting the visual experience of the game.

To overcome these design challenges, a simplified and functional visual style was adopted, with sprites specially adjusted for the game, aligning them with the theme and design of *Warbound*. Additionally, new sprites were created, specifically configured to reflect the different tactical formations available in *Warbound*, allowing players to distinguish between formations during the game.

The units in *Warbound* are categorized into two main types: Melee and Ranged. Each unit is represented by distinct icons that clearly illustrate its role and combat style, making it easy for players to quickly identify the units in play.

The possible units in the game are shown in the figure set below.

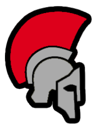


Figure 2: Hoplite

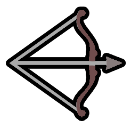


Figure 3: Archer

Figure 4: Cavalry

Figure 5: Viking

Figure 6: Hypaspist

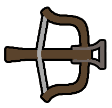


Figure 7: Crossbowman

Figure 8: Heavy Cavalry

Figure 9: Legionary

Figure 10: Men-at-arms

4.3 Formations

Formations are an important aspect in *Warbound*, offering players the ability to adapt their tactics based on the battle situation. Each formation was designed in a standardized manner to reflect its strategic purpose, influencing the defense, attack, and movement of units. The available formations are shown in the figure set below:

Figure 11: Standard

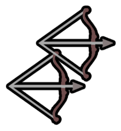


Figure 12: Spread

Figure 13: V

Figure 14: Shield Wall

Figure 15: Phalanx

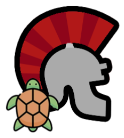


Figure 16: Turtle

4.4 Terrains

The terrain in *Warbound* directly affects the strategy and movement of units. Different types of terrain, such as plains, forests, and mountains, are visually distinct and provide strategic advantages and disadvantages. Below are the designs for each of the available terrains.

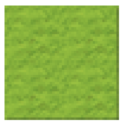


Figure 17: Plains

Figure 18: Forest

Figure 19: Mountain

5 Collaboration of Team Members

This section details the main contributions of the team members to the project. It is important to note that, although a member was assigned responsibility for certain tasks, this does not imply that they carried out all activities alone. The indicated responsibility primarily reflects leadership and coordination of the respective items, with occasional support from other team members as needed.

Arthur:

- Responsible for the main game logic, such as constructing the board and the graph
- Responsible for creating the base unit of the game
- Responsible for unit movement within the game
- Responsible for unit attacks within the game
- Responsible for designing the menu and tutorial
- Responsible for implementing unit directions in the game

Rodrigo:

- Responsible for developing and organizing the sprites
- Responsible for implementing directions in the existing sprites
- Responsible for the terrain logic in the game
- Responsible for creating new units
- Responsible for the logic of formations in the game

- Responsible for creating maps in the game
- Responsible for writing the report