Imprimir

PSI Group C Wiki

Group members

- Bedriñana Cárdenas, Renato
- Blanco Demelo, Hugo
- Blanco Pumar, Pablo
- Riveiro Vilar, Aarón

Project explanation

Our project will consist on a **turn based game** for two players, where each player will be represented on a grid as a tank, and will have the objective of **hitting the enemy tank**, until destroyed.

On every turn, the player will first be asked to **shoot** a cell. Then, it will be asked to **move** its tank to a cell, and the turn will end. Same for the other player, and repeat until one player is defeated, or until both players can no longer shoot.

On the first screen of the app, the user will be prompted to choose the gamemode, either User vs AI, AI vs AI or Training. Then it will be prompted to select a difficulty level for the AI(s), and the game will begin.

Features and characteristics

- The grid will consist on a 6x6 square.
- Players will have 5 health points.
- Players will not know the exact enemy position on the grid. Each time a player shoots, it's position will be revealed, but since it has to move before the turn gets to the other player, you can only have an approximate guess.
- When a cell gets hit, it becomes useless (can't move to that cell, or use it to move to another cell).
- There will be various types of ammo, with a limited number of shots for each type:
    - Standard: the normal ammo, hits the cell where it lands. Deals 2 HP. Available: 6
    - Precision shot: ignores wind. Deals 1HP. Available: 3
    - Nuke: hits a 3x3 area. Deals 3HP. Available: 1
- On each turn, there may be wind. It will deviate the landing point of the shot, depending on the strength and the direction. Players will only know the wind characteristics of the previous turn. Wind will not make abrupt changes, so that players can approximately guess the wind for their turn.
- Players will have 20 units of fuel. For every move (going from a cell to an adjacent one) they will consume one unit of fuel.
- Players will choose their desired cell to move to, and the game will find the most optimal path.
- Players can move up/down, right/left.
- Players can opt to not moving (by selecting its own cell).
- If a player can't move, either because of no fuel, or because there is no available path, it will be able to select its own cell.
- After the enemy's turn (in case of Human vs AI), the enemy's last known location will be visible on the grid.

Not implemented ideas

- Some cells may be destroyed from the start, simulating rivers, rocks, and other obstacles.
- Other ammo:
    - Strong shot: wind effect is duplicated. Deals 2HP. Available: ?
    - Fire bomb: makes the row it hits unusable for the next turn. Deals no damage. Available: ?

- Add special abilities that players may use on their turn:
    - Smoke bomb: the player's position will not be revealed after shooting.
    - Shield: only for the next enemy's turn, obtain a shield that will ignore the enemy's shoot.
    - Spikes: for the next enemy's turn, if the player gets hit, it will replicate the damage to the enemy
    - Bridge: can be used on a destroyed cell in order to make it usable again

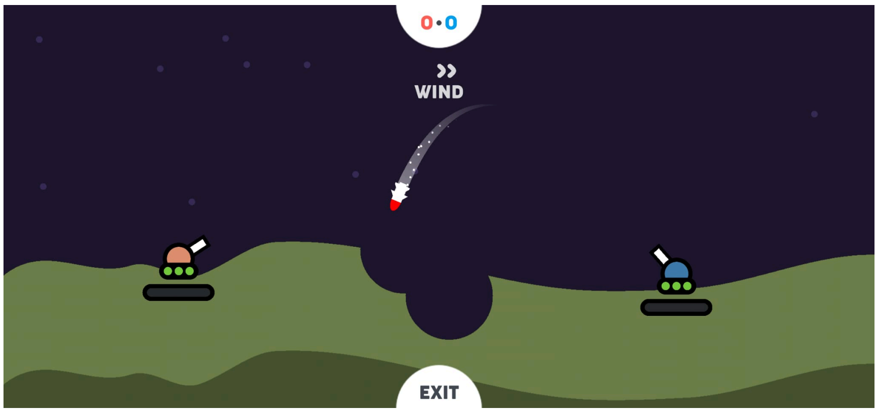
AI Implementation

- Random AI: this AI will take all decisions based on randomness. Useful for testing the application, or for training the normal AI.
- Normal AI: this AI will take decisions based on Q-Learning, with two tables, one for shooting and one for moving.

More detailed information and examples can be found below in the mocks section.

State of the art

Our first inspiration came from a game called *Cannon Duel*, available in an Android app known as *Two Player Games*. From this game, we took the idea of the core for our project. Each player is represented with a tank, and the objective is destroying the enemy. We also took some idea like different ammo types, wind, destroyed terrain...

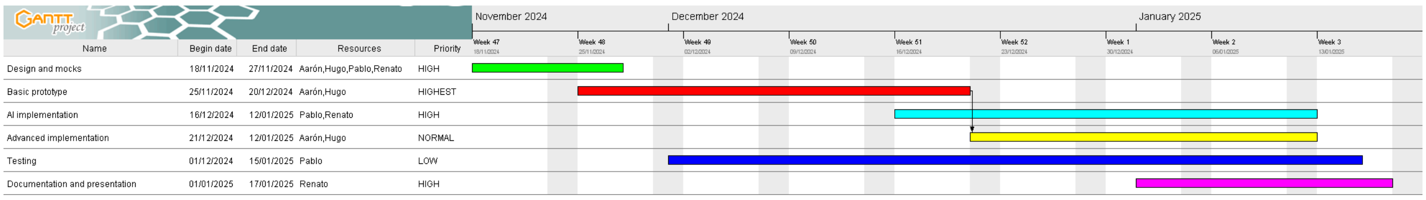


We then combined the ideas from this game, with the classic *Battleship* game, from where we took the idea of making the game turn based, as well as using a coordinate based grid. We also took inspiration of other features from this games, such as the unknown enemy position.

As a result, with the combination of both games, and our own ideas, we got the design for our project.

Project planning

This is the planning for the development of our project. We will try to follow it as closely as possible, but changes may be made as needed, such as adjusting task dates or reassigning participants.



Name	Begin date	End date	Resources	Priority
Design and mocks	18/11/2024	27/11/2024	Aarón,Hugo,Pablo,Renato	HIGH
Basic prototype	25/11/2024	20/12/2024	Aarón,Hugo	HIGHEST
AI implementation	16/12/2024	12/01/2025	Pablo,Renato	HIGH
Advanced implementation	21/12/2024	12/01/2025	Aarón,Hugo	NORMAL
Testing	01/12/2024	15/01/2025	Pablo	LOW
Documentation and presentation	01/01/2025	17/01/2025	Renato	HIGH

A document with more detail, such as tasks descriptions, can be found here.

While we didn’t stick to the plan perfectly, it turned out to be really helpful for organizing the team.

Resources used

Some online resources and courses:

- Android Developers http://developer.android.com/index.html
- Android Developer NanoDegree https://www.udacity.com/course/android-developer-nanodegree--nd801
- Programming Mobile Applications for Android Handheld Systems: Part 1 https://www.coursera.org/learn/android-programming
- Programming Mobile Applications for Android Handheld Systems: Part 2 https://www.coursera.org/learn/android-programming-2
- Android programming course: learn how to create your own applications http://www.sgoliver.net/blog/curso-de-programacion-android/
- Kotlin programming language: https://kotlinlang.org/docs/getting-started.html
- Jetpack Compose UI Toolkit: https://developer.android.com/compose
- Chaquopy documentation: https://chaquo.com/chaquopy/doc/current
- ChatGPT
- Many other online forums, blogs, and discussions

Class concepts

- **Reinforcement Learning**:
  The AI player leverages reinforcement learning techniques to improve its decision-making process. Using a Q-learning approach, the AI updates its Q-table after each shot and move, associating actions with rewards to maximize its performance over time.

- **State-Action Mapping**:
  Each game state is represented as a combination of variables, such as player position, fuel levels, and available ammunition. The AI determines the optimal action (e.g., which cell to move, which ammo to use...) by evaluating the Q-values associated with each state-action pair.

- **Reward System**:
  The AI uses a reward-based system to learn from its actions. For instance:

  - Hitting the opponent grants a positive reward.
  - Missing a shot or moving inefficiently results in a negative reward.
  - Strategic movement is rewarded to encourage intelligent behavior.

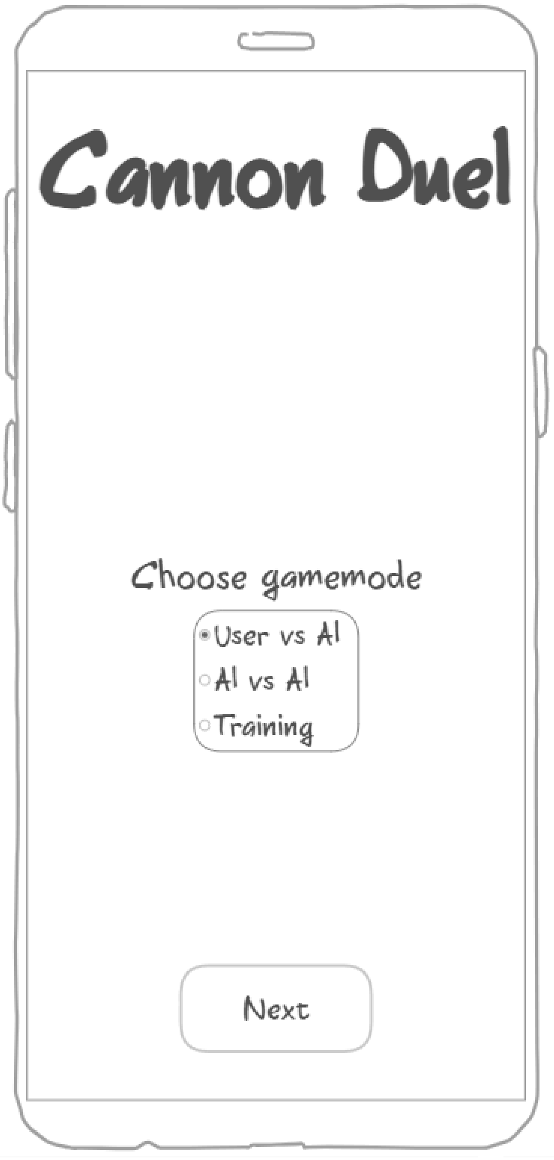
- **Exploration vs Exploitation**:
  The AI employs an epsilon-greedy strategy, balancing exploration (trying new actions) and exploitation (choosing actions it knows are effective). This ensures the AI can discover better strategies while still utilizing known optimal moves.

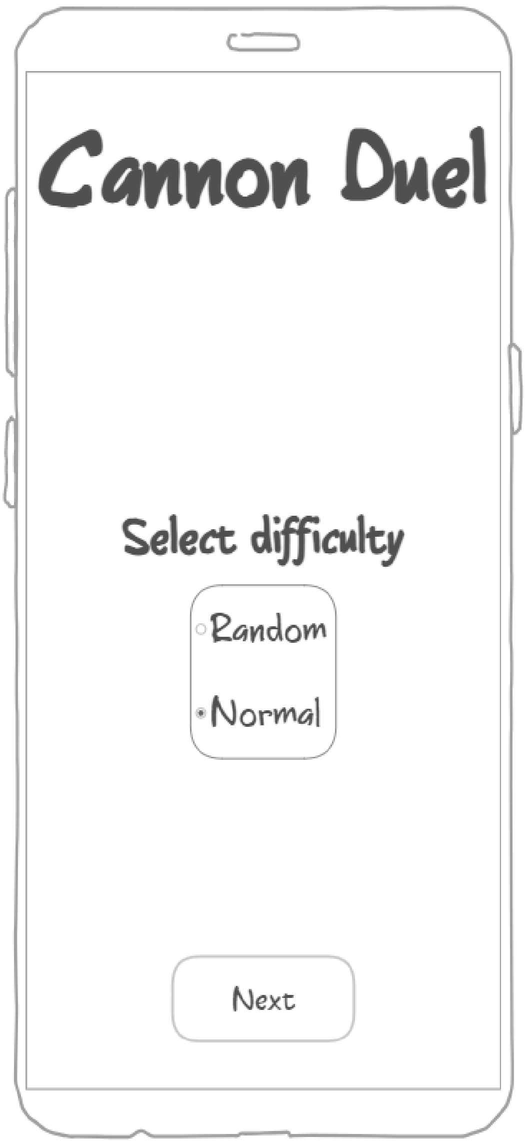
- **Environment Simulation**:
  The AI operates within a simulated game environment where it interacts with dynamic variables such as wind direction and strength, which impact the trajectory of its shots. This adds complexity and realism to its decision-making process.

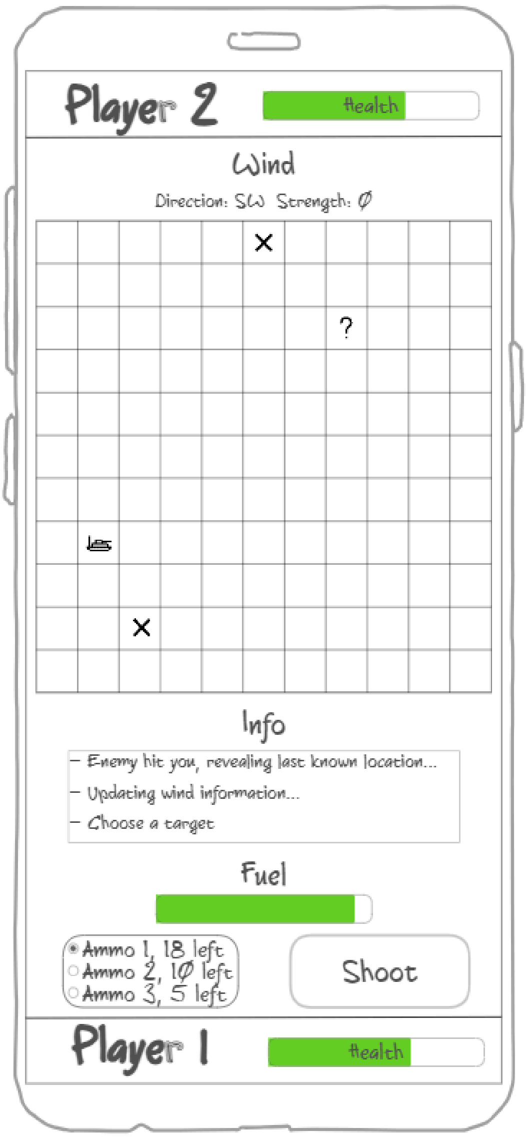
These concepts allowed us to develop an AI that adapts to the game dynamics and improves its performance over time, creating a challenging and engaging opponent for players.

Due to the nature of the game, the AI training heavily relies on the specific playing style of the human opponent. After extensive training against the same user, the AI adapts and begins to recognize patterns unique to that player.

Mocks

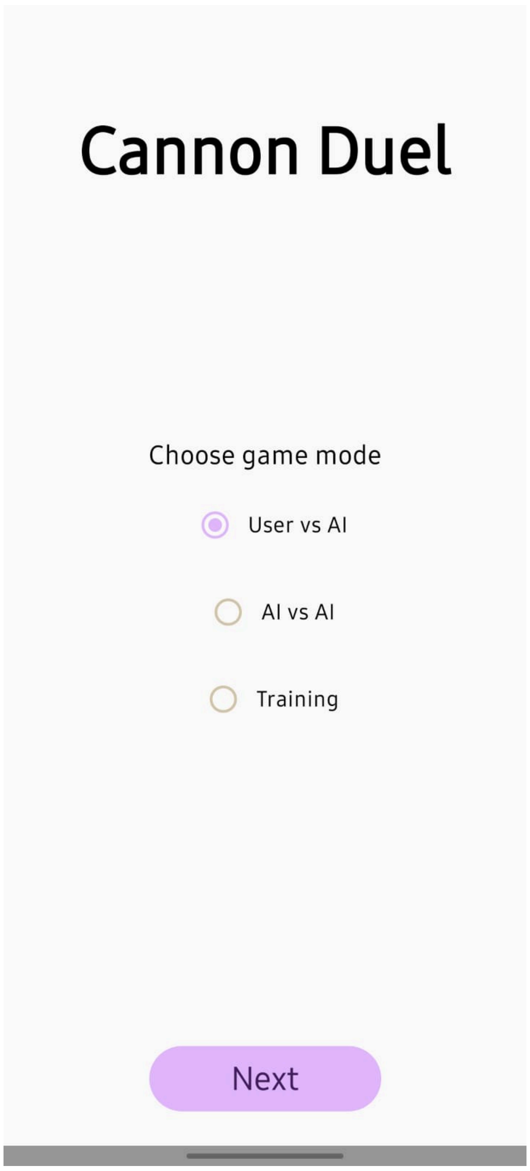

Cannon Duel
Choose gamemode
[✓ User vs AI]
AI vs AI
Training
Next

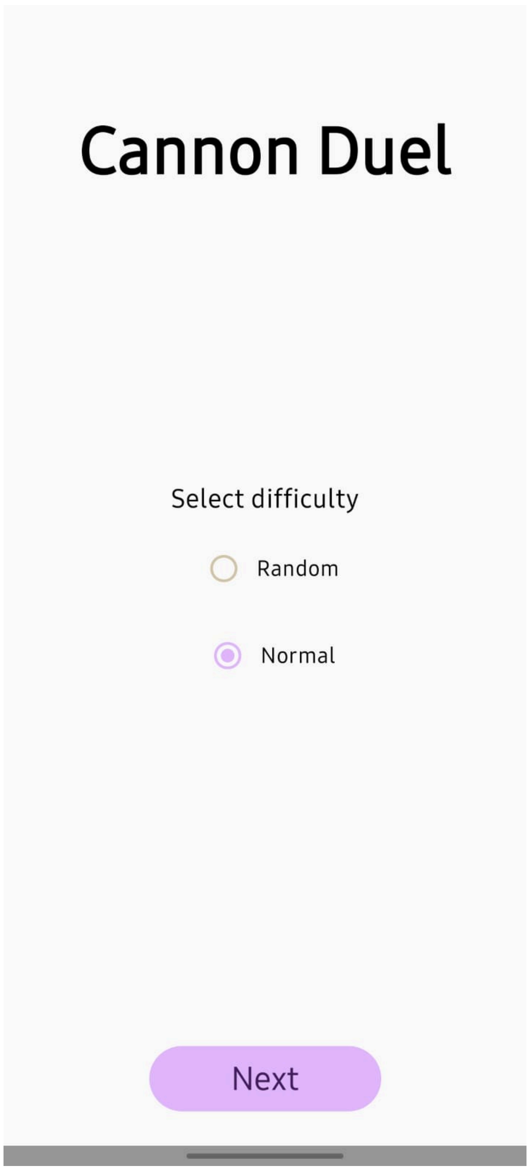

Cannon Duel
Select difficulty
Random
[✓ Normal]
Next

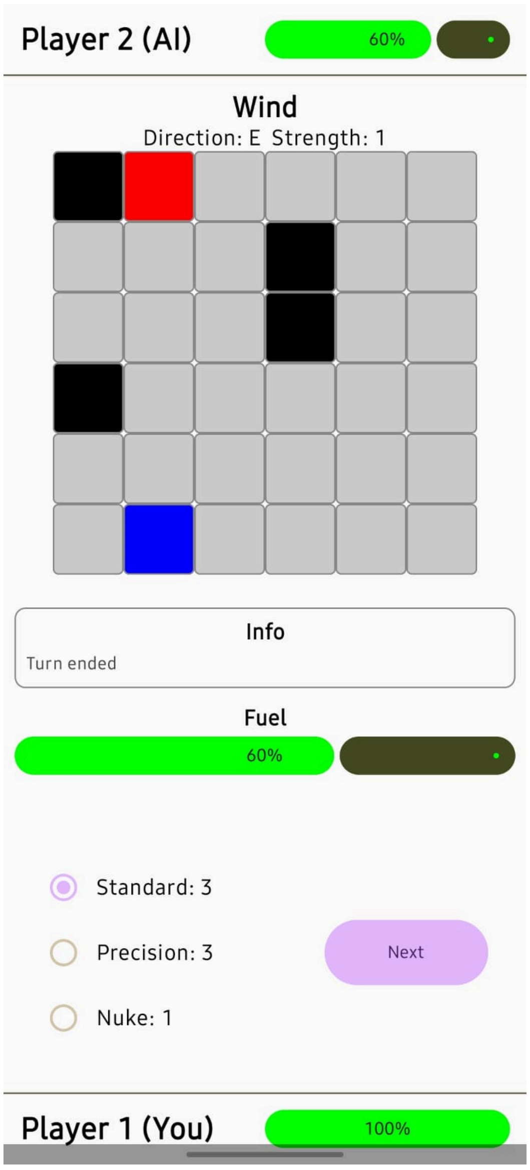


- Enemy hit you, revealing last known location...
- Updating wind information...
- Choose a target

Final Interface







Player 2 (AI) 60%

Wind

Direction: E Strength: 1

Info

Turn ended

Fuel

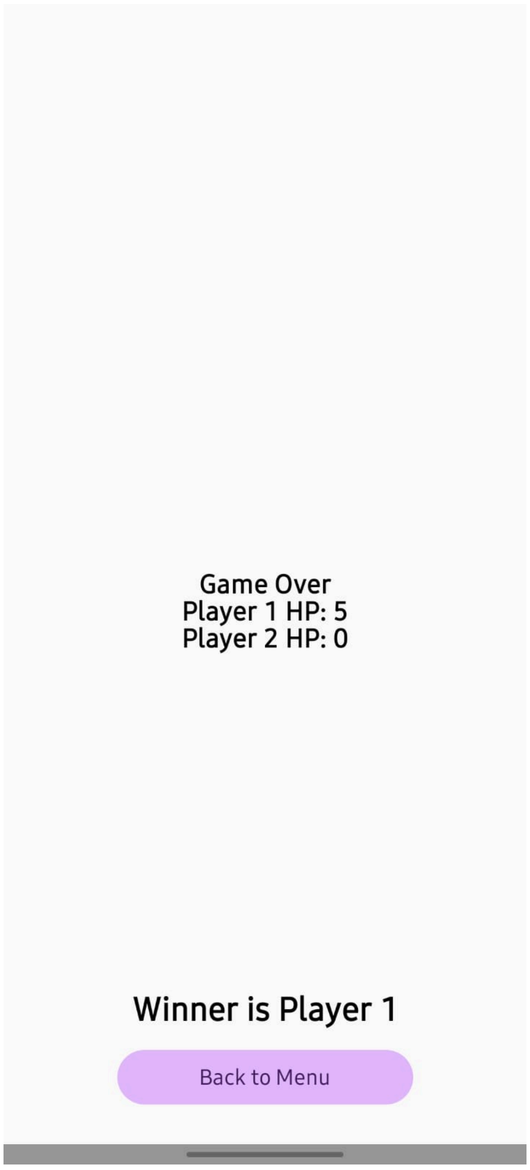
60%

[✓ Standard: 3]

Precision: 3

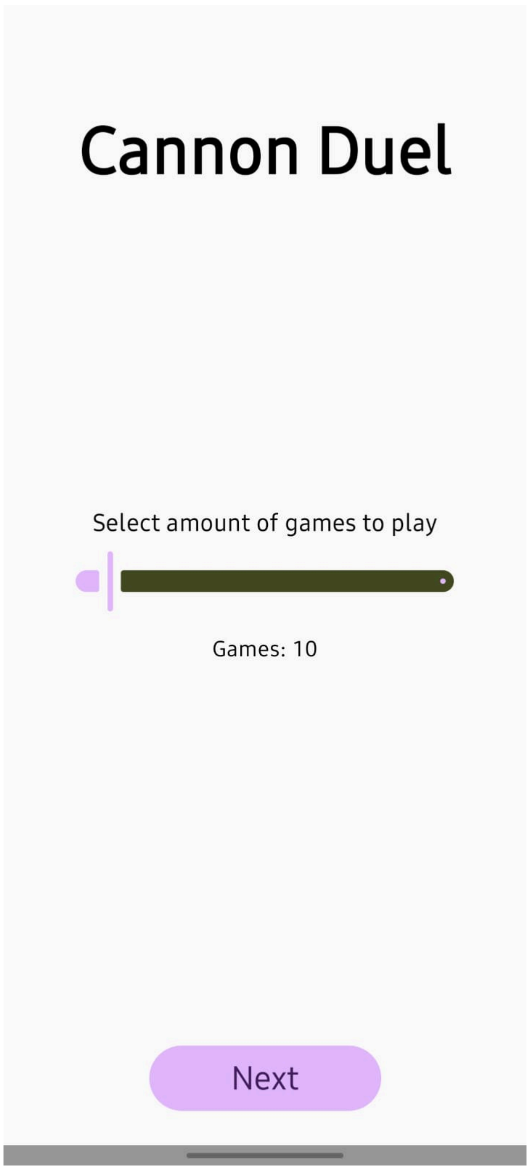
Nuke: 1

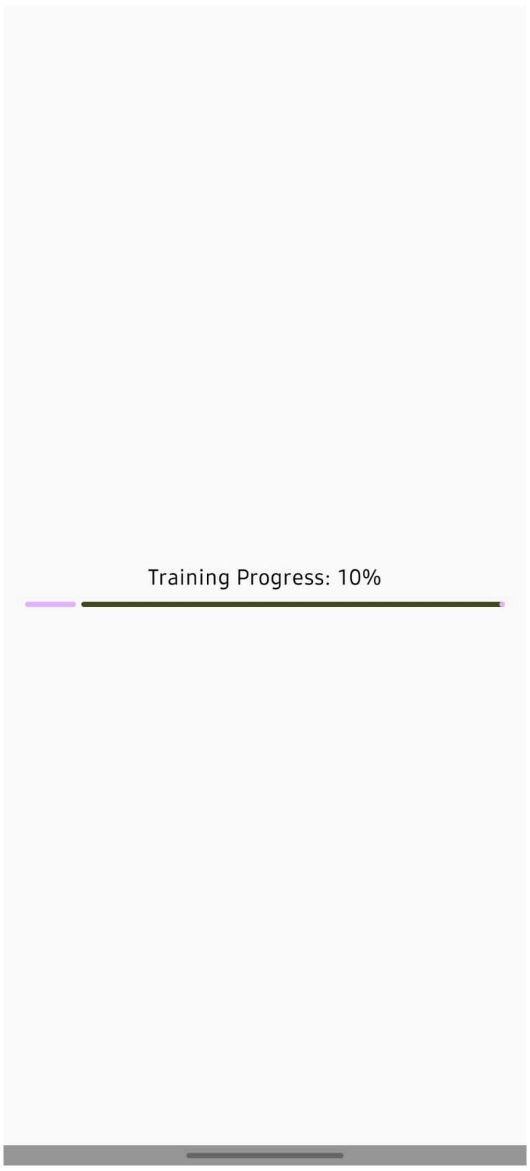
Player 1 (You) 100%



Game Over
Player 1 HP: 5
Player 2 HP: 0

Winner is Player 1





Individual work

- Renato Bedriñana Cárdenas
    - AI Implementation: ~3 hours
    - Documentation: ~4 hours
    - Testing: ~2 hours
    - Presentation: ~3 hours
- Hugo Blanco Demelo
    - AI Implementation: ~3 hours
    - GUI Developement: < 1 hour
    - Documentation: ~4 hours
    - Testing: ~2 hours
    - AI Training: ~5 hours
    - Presentation: ~3 hours
- Pablo Blanco Pumar
    - Initial design: ~2 hours
    - GUI development: ~2 hours
    - Core game development: ~5 hours
    - Testing: ~3 hours
    - Presentation: ~3 hours
- Aarón Riveiro Vilar
    - Initial design: ~3 hours
    - GUI development: ~5 hours
    - Core game development: ~20 hours
    - AI integration: ~10 hours
    - Testing: ~5 hours
    - Wiki: ~2 hours

References

- Android Studio: https://developer.android.com
- Project planning: https://www.ganttproject.biz
- Mocks: https://ninjamock.com
- Repository: https://github.com
- Python SDK for Android: https://chaquo.com/chaquopy/