Help session for the robot Project

Alexandre LETOIS

1er février 2021

1 Project

Subject

The project is separated in three main parts :

1. Robots
2. Interface
3. Code linking everything together

A few hints for each parts will be displayed after this slide

Robots

You can use whatever method your prefer to represent this part, but I will show one using OOP. By representing robots by objects, you have different advantages :

- All robots will be sharing the same properties and characteristics
- HP, Weapon slots, Movements, etc are object attributes
- No need to code each robot separately
- Deactivated robots are also robots, making them easier to create
- If you want to add specific properties for fighting robots and the deactivated robots, you can create an object "Robot" with 2 children :
  1. Fighting_robot for the robots of each team
  2. Deactivated_robot for the deactivated robots

Weapons and Bodies

The weapon and bodies are equipment that the robots can use. Only one body can be equipped at a time, but multiple weapons may be equipped at the same time if the robot has enough weapon slots.

- They can be represented as objects
- You can use inheritance to assign the different properties of each equipment without re-coding everything from scratch
- This way, a robot can use an "attack()" method that will directly call the properties of the equipped weapons.
- If a body is changed, then the robot lose X HP before equipping the new body (it can result with the destruction of the robot this way)

Interface

- Use buttons for different options :
  1. Run a game until one team is the winner
  2. Run 1 turn
  3. Other possibilities
- Use menus for different options :
  1. Quit the app
  2. Chose the number of robots in each team
  3. Chose the number of deactivated robots
  4. Chose the size of the grid

You can colorize the cells to differentiates each team or import a picture. Also, don't forget to show the direction each robot is facing !

2 Robots

Attributes

- All the characteristics needed can be contained in the attributes of the Robot object
- HP are the number of hits that the robot can endure before being destroyed
- Mvt are the number of cells that the robot can travel at each turn
- A robot can only move horizontally or vertically, not diagonally
- Weapon slots are the number of weapons that can be equipped. Each weapon takes 1 weapon slot when equipped
- Other attributes can be used here if you need them, some examples here :
  1. Name or number of the robot
  2. Robot behaviour (offensive, defensive, etc)
  3. Etc

Weapons

- Every time a robot attack, it uses a method of its equipped weapons that will make the attack
- The weapon has a specific pattern of attack and specific damages, depending of the weapon
- A weapon uses a weapon slot when equipped
- When a robot fires, it shoots with every equipped weapon

Bodies

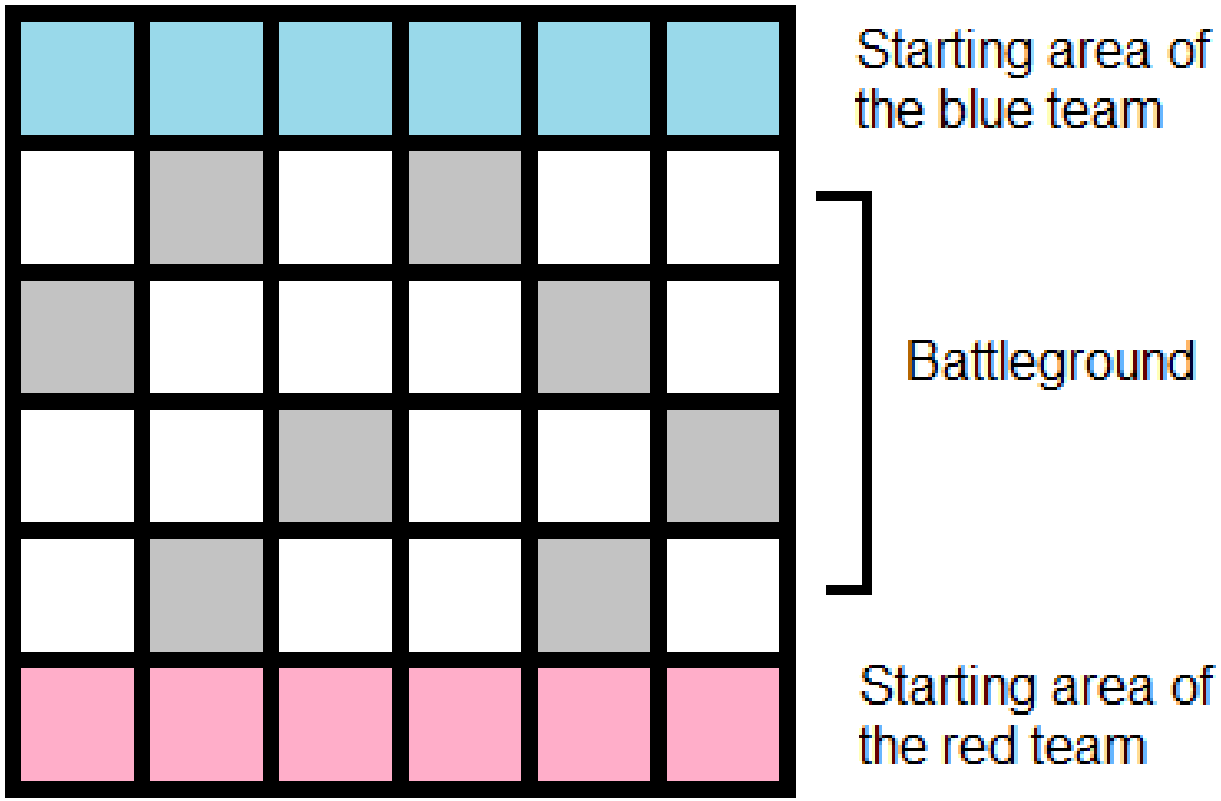
- Bodies are specific objects
- When you equip a body, your robot acquires new HP
- When you removes a body, the robot loses those HP
- Other specific properties can be acquired with some bodies

3 Interface

Grid

- The grid can be represented by buttons or labels
- Color your widget to represent the team
- The grid is separated in 3 different area :
  - Blue area
  - Battleground
  - Red area
- You have to place each robot in a specific area at the start
- You can separate each area in the code, so you can place each robot randomly in its own area
- Don't forget to show the direction that the robot is facing (with an arrow or a dot for example)

Area



4 Code

- Code relative with the grid
- Code relative with the robots
- Other points of the code

4 Code

- Code relative with the grid
- Code relative with the robots
- Other points of the code

Grid code

This code will need to manage the movements of the robots on the grid. A few points needs to be assured

- Make sure that a robot cannot go outside of the grid
- A robot cannot walk on another robot
- Change the interface accordingly with the robots movements
- Manage the direction the robots are facing
- Change the interface when a robot is destroyed

4 Code

- Code relative with the grid
- Code relative with the robots
- Other points of the code

Robot code

This code will need to manage the movements of the robots, the combat aspect, etc A few points needs to be assured

- A robot needs to pick up an item when he walks on it
- Manage the inventory of the robots (weapons and bodies)
- Manage the destruction of robots (remove them from the grid and drop an item)

Behavior

For the behavior, a lot of different things can be done. You can decide to implement different strategies for your robots :

- Aggressive
- Defensive
- Specific strategies

If you have any ideas with the behavior, don't hesitate to be creative. Also, you don't **need** to create something specific for the behavior, so don't worry. If you have no ideas or you do not think you can do it, the expected behavior is to just randomize all the robots actions.

4 Code

- Code relative with the grid
- Code relative with the robots
- Other points of the code

Code

A few things need to be implemented in the code to assure that the app is working correctly

- Check the winning conditions at the end of each turn
- Be sure that when a robot is destroyed, it can't take its turn anymore