

Programming 'Ricochet Robot' with pygame

4TPU140U

2022 - rev. 0.1

The aim of this project is to implement a game named “ricochet robot” created by Alex Randolph.

First we start to learn how to implement a simple game by using pygame.

1 Pygame and the bouncing ball example

Pygame is a library allowing to write some games in Python. The website of that library is <https://www.pygame.org/> and the page of it's documentation is <https://www.pygame.org/docs/>.

First, download from https://www.labri.fr/perso/boussica/archives/cours/4TPU140U/bouncing_game_in_one_file.tar.gz

a simple implementation of a bouncing ball. Execute the project and analyse the source code to understand how the application works.

Find in the source code when and how :

- a window is created,
- the images and the fonts are loaded in memory,
- the keyboard events are processed,
- the ball position and the ball velocity are updated,
- the text and the ball are displayed.

Now, download from

https://www.labri.fr/perso/boussica/archives/cours/4TPU140U/bouncing_game.tar.gz, the same source code, written in order to become a more scalable and complex game project.

What are the differences? What functionality are added in that version? Why, this project is more scalable and more maintainable?

2 Ricochet Robot

Ricochet Robot is a game created by Alex Randolph.

The rule of that game can be viewed on that video :

<https://www.youtube.com/watch?v=0QOMmftaWAQ>.

The game you will implement, will allow a player to :

- loads some maps ;

- plays the game by placing some robots randomly, defining randomly a sequence of color targets, and allowing the player to move the robots. The player wins if each robot goes to each target color with respect to the order of the given sequence of color targets,
- implements a map editor that allow to create easily new maps.

The following four part will be discussed and noted in that project :

- the part that display the game,
- the part that update the data of the game/editor,
- the part that load/save maps in files,
- the part that process all the incoming mouse and key events.

The board game can be coded in different ways. A way to do it is to implement the board game with three arrays. An array of 0 and 1 for the vertical walls (0 means there is no wall), an array of 0 and 1 for the horizontal walls and an array of integers or string for the cells.

For example, the board game of Figure 1, can be encoded with those three following arrays :

```
vertical_walls = [
  [1, 0, 0, 1, 0, 0, 0, 1],
  [1, 0, 0, 0, 1, 0, 0, 1],
  [1, 0, 1, 0, 0, 0, 1, 1],
  [1, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 0, 0, 0, 1, 0, 1],
  [1, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 0, 0, 1, 0, 0, 1],
]
horizontal_walls = [
  [1, 1, 1, 1, 1, 1, 1],
  [0, 0, 0, 0, 1, 0, 0],
  [0, 0, 0, 0, 0, 1, 0],
  [0, 0, 1, 0, 0, 0, 0],
  [1, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 1, 0, 1],
  [0, 0, 0, 0, 0, 0, 0],
  [1, 1, 1, 1, 1, 1, 1],
]
cells = [
  [0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0],
  [0, 't1', 't2', 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 'r2', 0],
  [0, 0, 0, 0, 0, 0, 0],
  [0, 'r1', 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0],
]
```

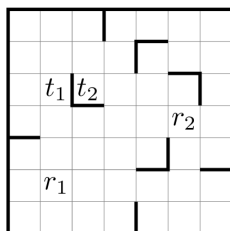


FIGURE 1 – An example of board game (this image is from the article [1]).

3 Bibliography

- [1] Samuel MASSEPORT et al. “Ricochet Robots game : complexity analysis Technical Report”. working paper or preprint. Juill. 2019. URL : <https://hal.archives-ouvertes.fr/hal-02191102>.