Paddle

Thomas Fritchman - W00961639

April 29, 2014

Contents

1 User's Guide

1.1 Goal

The goal of *Paddle* is to break away all of the blocks using the paddle and the ball. If the ball leaves the screen area, the game is over. Try to get the highest score!

1.2 Powerups

There are two types of powerups in the game. The first gives you an extra ball for an extra challenge, so long as you keep one ball in the screen you're fine. The second type of powerup turns your ball(s) into a ball of fire! Use this flame to break right through blocks without bouncing off of them.

1.3 Scoring

Hitting a green block will give you 10 points. However, if you hit another block before the ball hits the paddle again, you are awarded an additional 20 points, then 30, 40, etc. As you can tell, it is advantageous to you're score to destroy as many blocks at a time as possible. Blue colored blocks multiply your combo bonus by 2x and purple blocks by 4x so try to work these into your combos!

1.4 Controls

Key	Action
Mouse movement right	Move paddle clockwise.
Mouse movement left	Move paddle counterclockwise.
Esc.	Exit to menu.
F	Fullscreen (Linux only).
Q	Quit game at any time.

2 Module Documentation

- **Modules**
    - **game.py** contains most of the game code. All of the game logic and classes are contained in this file.
    - **constants.py** all game constants. The game can be manipulated and tweaked easily using these constants.

- **utils.py** has several helper functions for loading multimedia elements into the game.
- **levels.py** is where each level in the game is specified in the form of an array of integers. Each integer value corresponds to an object on the playing field as documented below:

value	object
0	Empty
1	Green block
2	Blue block
4	Purple block
10	E pointing ball spawn
11	SE pointing ball spawn
12	S pointing ball spawn
13	SW pointing ball spawn
14	W pointing ball spawn
15	NW pointing ball spawn
16	N pointing ball spawn
17	NE pointing ball spawn
20	Ball powerup spawn
21	Flame powerup spawn

- **Functions**

  - **main()** is the entry point for the game. Pygame is initialized and the game environment is set up.
  - **menu()** is the entirety of the game menu. From here, we can see the game's levels, see in game documentation and begin a game.
  - **game()** starts a new game of Paddle on a given level. This function contains the main game loop and is called each time the player begins a new level.
  - **draw_tiles()** generates the level based on a given level array. Creates tile objects for each tile in the level.

- **Classes**

  - **Paddle** calculates the position and geometry of the paddle. Sprite class which draws image to it's surface.
  - **Ball** keeps track of a ball's position and whether to display the flame image or the default ball image. Default constructor takes a position and a direction. Speed is constant.

- **Tile** is a piece of the level's tile, default constructor takes a position.
- **Ball_Powerup** is a sprite who's collision with a ball triggers another ball to spawn at a random spawn point.
- **Fire_Powerup** is a sprite who's collision with a ball causes all balls to become a flaming sprite that breaks through tiles without bouncing off of them.
- **Spawner** is used to spawn balls and powerups at points on the playing field determined by the "level" array.
- **HUD** is the heads up display. It is used to calculate the current score and display it on the screen.

3   Cheats

Key	Action
1	Spawn another ball
2	Spawn additional ball powerup
3	Spawn flame powerup
4	Turn current balls into flame balls
5	Invincibility mode

4   Acknowledgements

5 Autobiographical Info

This game is actually the second game that I've made. It is however my first original game as my first one was a very simple clone of Tetris done also in Pygame. My goal in this project was to finish it with a fully functioning game that looked relatively polished and complete. I think that for the most part I've accomplished that goal. My biggest challenge in this project were actually some of the most trivial seeming things. For example the concept of drawing a paddle that rotated *around* the screen, a key part of the game, seemed like a pretty simple task in my mind. However, implementing the geometry of the paddle and the geometry of the ball's reflections off of the paddle and blocks were pretty difficult for me and took many hours. I was surprised by how much of a difference adding a few textures and some music to the game really livened the game up. My prototype version was just made up of solid rectangles and looked pretty boring. In the future I would put more time into cleaning up the code because I realized that by the end of the project it became a bit of a mess, but fortunately not to the point of being unmanageable.