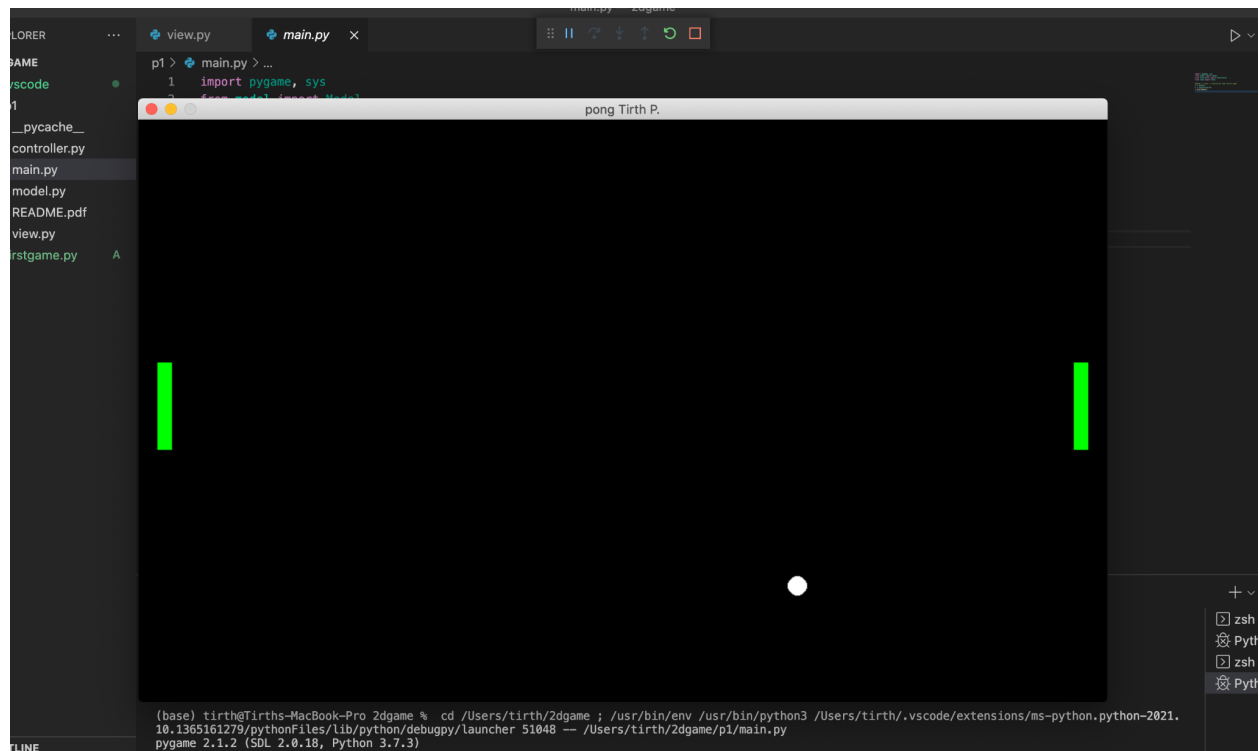


Tirth Patel
2/24/2023
CPSC4160

OS Versions: tested on macOS 10.15.5, Windows 10 and SOC virtual computer

Python version: 2.7.16

Pygame: 2.1.2



Motivation:

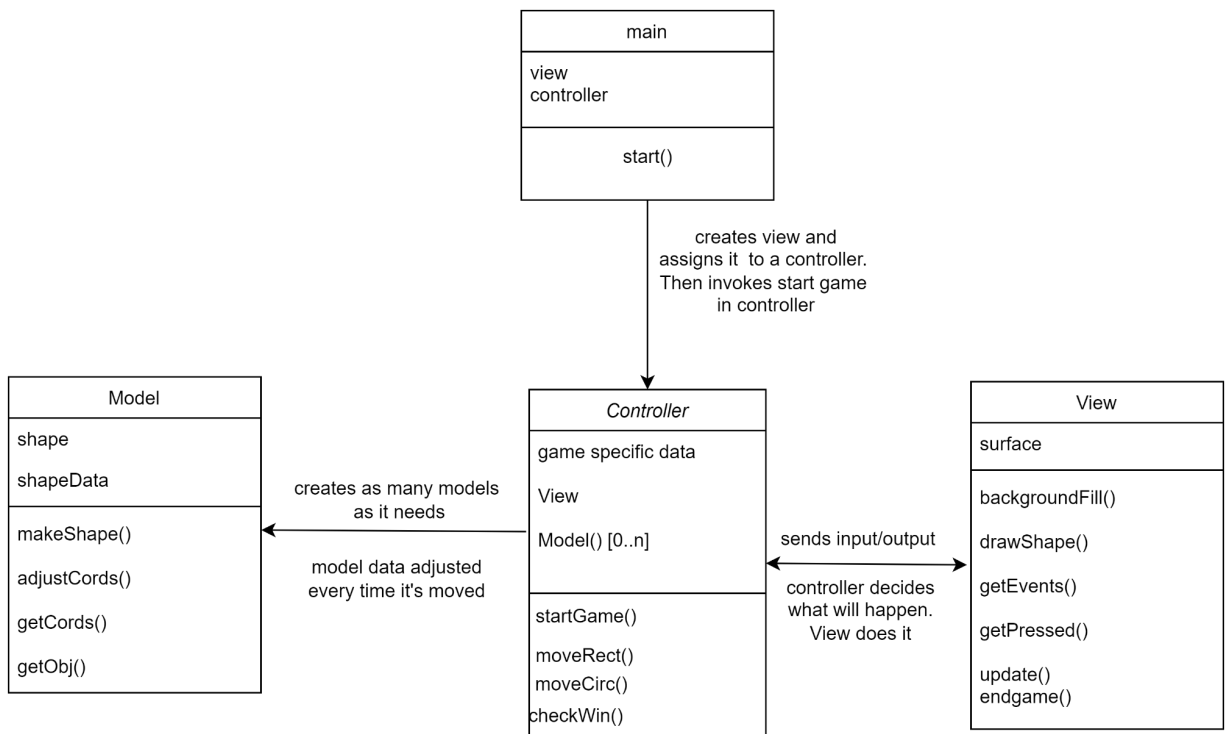
Pong seemed like a good starter game to build. Enough systems to help get a decent grasp of the basics without being overwhelmed. Implementing systems like objects controlled by a person, ones that move on their own as well as some movement physics helped me learn some of the pygame API a bit more.

Reasoning:

The structure is supposed to be the typical MVC architecture. Having written code in MVC format for 2150 certainly helped starting out but I hadn't done it before with a game. Since the 'models' and 'controller' parts are very connected due to pygame the biggest challenge was figuring out where exactly to separate them.

In terms of organization I had the model section create all the objects using data sent from the controller. The controller took and sent data from the view and held all of the code for interactions and movement. Finally the view is just responsible for starting the game window and drawing everything.

Diagram:



Future work:

Make better boundaries between controller and models, right now there is some overlap specifically with the movement of circles.

Create a better system to determine the shape of the model upon initialization. Maybe implement subclasses or something?

Fix some of the bugs that occur during collisions. Also figure out the problem with `collidirect` where the ball goes into one of the paddles.

Going forward I would also like to figure out a better way to store all the models. Right now since there are only 3 elements it's not a big deal but once that number starts to increase it will get messy.

Generalization: The overall structure can most likely be transferred to another game. The model will be improved and the controller will change based on the game but the view can most likely be built upon in the future. For the model more shapes can be added other than just rectangles and circles. Most of the getter methods can also be generalized for the other shapes.