Justin Hu

Vincent Hwang

Marcus Pearce

Period 5

***APCS Final Project:* Pac-Man**

In this project, we are re-creating the iconic game *Pac-Man*. The player controls Pac-Man through a maze, eating pac-dots (also called biscuits or just dots). When all pac-dots are eaten, Pac-Man is taken to the next stage. Between some stages, one of three intermission animations plays. Four enemies (Blinky, Pinky, Inky and Clyde) roam the maze, trying to catch Pac-Man. If an enemy touches Pac-Man, he loses a life. Whenever Pac-Man occupies the same tile as an enemy, he is considered to have collided with that ghost. However, if Pacman eats a big pellet, Pacman is able to eat the ghosts, making the ghosts return to their spawn point.

To create Pac-Man, we are using the software program Greenfoot. Greenfoot teaches object orientation with Java. Create 'actors' which live in 'worlds' to build games, simulations, and other graphical programs. Greenfoot is visual and interactive. Visualisation and interaction tools are built into the environment. The actors are programmed in standard textual Java code, providing a combination of programming experience in a traditional text-based language with visual execution.

**Structural Design:**

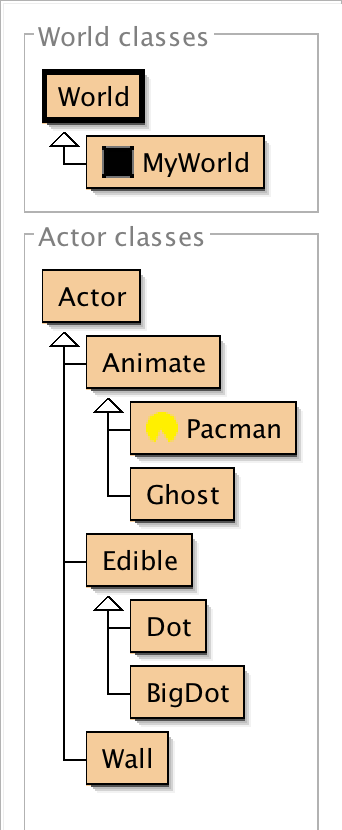
Array - Used to create grid for map

Stack - Track user input

Queue - Track the order of dead enemies.

**Object-Oriented Design:**

The below diagram is a class diagram for this project. It includes four abstract classes: World, Actor, Animate and Edible. Concrete classes include MyWorld, Pacman, Ghost, Dot, BigDot and Wall.



Class Summaries:

*World* - World in which Actors live in. It is a two-dimensional grid of cells

*MyWorld* - Customized world that initializes positions of all actors. Saves conditions such as “flipped” and individual status of Actors (if dot is eating, if Pacman is dead, etc.)

*Actor* - An object that exists in the world. Every Actor has a location and interacts with other Actors.

*Animate* - Abstract class that encompasses the moving Actors (Pacman, Ghost). These objects move around the world.

*Pacman* - The Animate Actor controlled by the user. Pacman can eat Dots, BigDots, and Ghosts when the condition is “flipped.” It moves according to user input.

*Ghost* - The Animate Actor controlled by algorithm that chases Pacman. Main objective is to eat Pacman, does not interact with Dots or BigDots. Can be eaten by Pacman if condition is “flipped.”

*Edible* - Abstract class for items made solely to be eaten by Pacman (Dot, BigDot).

*Dot* - Edible Actor that increments points of Pacman - these must all be eaten for the game to be won.

*BigDot* - Edible Actor that temporarily changes the status of the world to “flipped.”

*Wall* - Actor that establishes the boundaries for other actors; Animate actors cannot move through Walls.

**Detailed Design:**

The documentation for the game will be made soon.

**Testing:**

The movement of the ghosts and Pacman will need to be tested. The interactions between Pacman and the ghosts will also need to be tested.