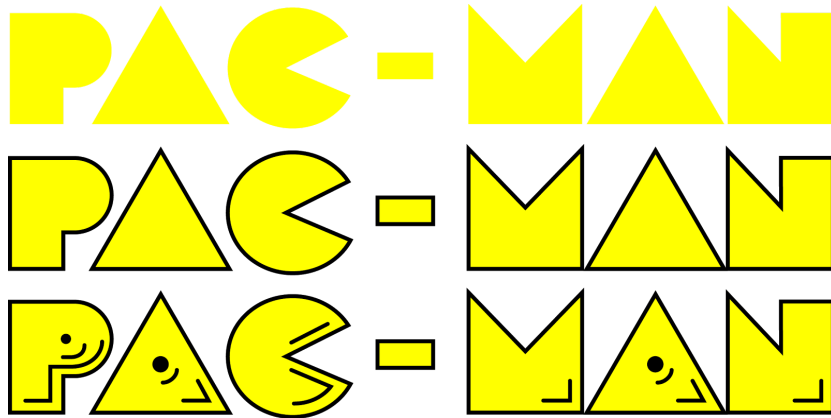


CS 110 Final Project, Fall 2017



https://github.com/lizroedel/CS110_Final_Project

Team:

Megha Patel

Lizzy Roedel

Rachel Yap

Description of Project

Our group decided to replicate the popular 1980's arcade game known as PAC-MAN. The object of the game is to have the main player, PAC-MAN, eat all of the pac-dots in order to win the game. With each pac-dot that PAC-MAN consumes, the player's score increases by one out of a total of 141 pac-dots. The challenge is for PAC-MAN to avoid contact with the ghosts that are programmed to randomly move throughout the maze. In our version, if the player touches a ghost it kills PAC-MAN and ends the game.

II. GUI Design

1. Main Menu/Start Screen

This screen is the first thing a user sees when the game is called. The Start Screen allows the user to click anywhere on the screen to continue with the game.

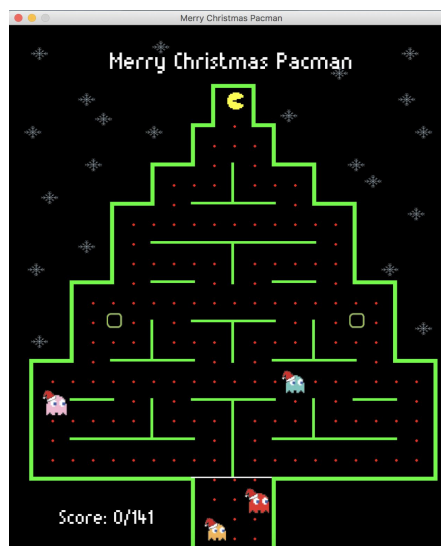
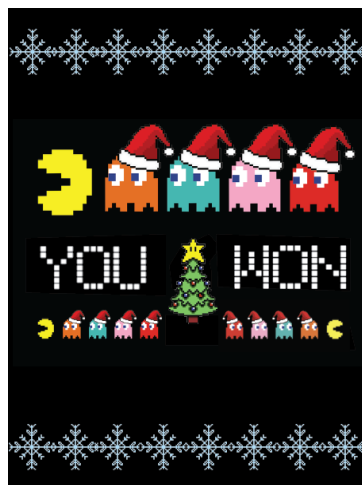


GUI Concept

The Start Screen is composed of an image made by Adobe Illustrator. The Screen appears with the `pygame.blit` function and asks for the user to click the screen to carry on with the Game Screen.

2. The Game Screen

This screen is where the actual game of Merry Christmas Pacman takes place. Using the keyboard arrow buttons, they direct the pacman to move throughout the maze and eat the pellets. When Pacman collides with a ghost the Game Over Screen appears. If Pacman eats all of the necessary pellets, the Winner Screen appears.



GUI Concept

The Maze is composed of vectors. Sprites are used and they appear as the ghosts and pacman. The pellets were coded using ellipses.

3. Game Over Screen

This screen appears when Pacman collides with a ghost.



GUI Concept

The Game Over Screen is composed of an image made by Adobe Illustrator. The Screen appears with the `pygame.blit` function.

5. Final GUI Interface

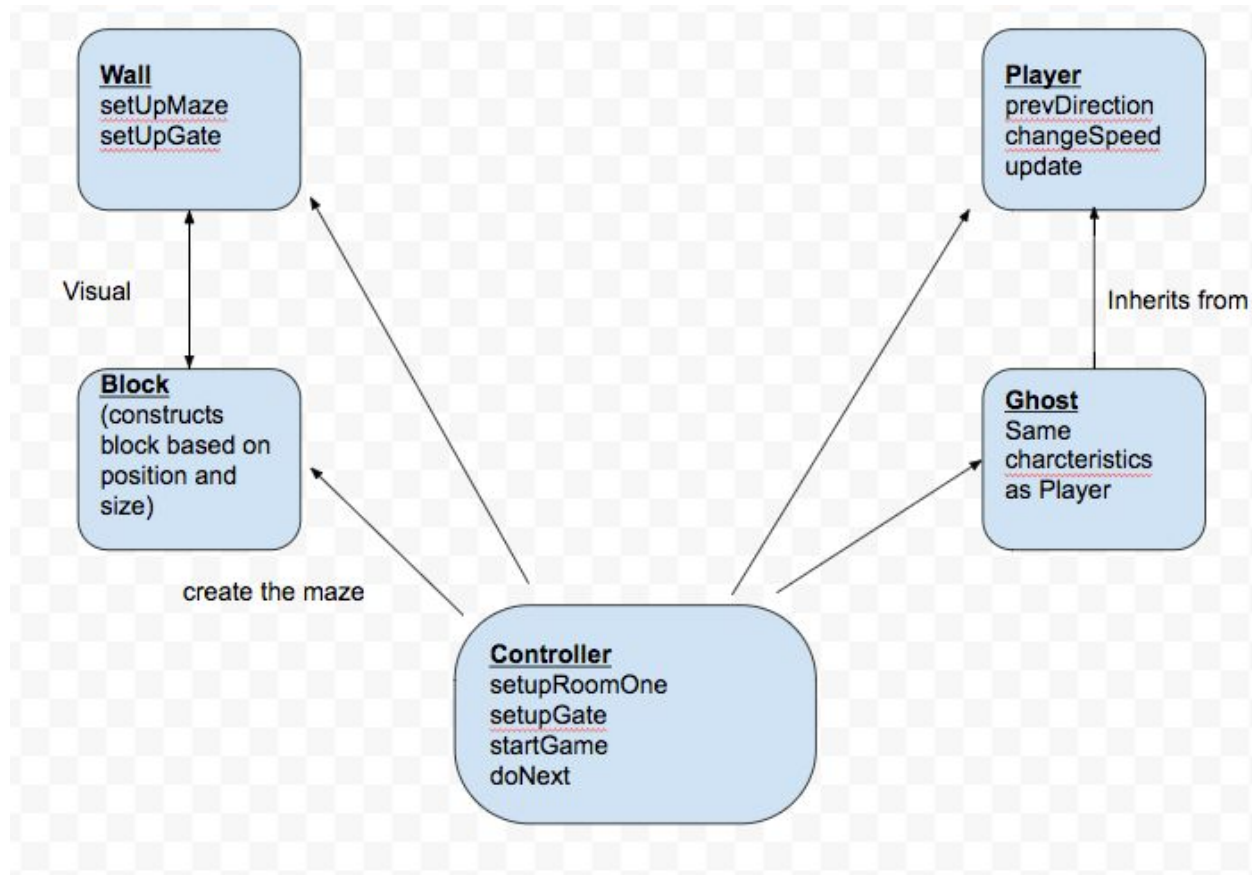


III. Program Design

Non-Standard Libraries and Modules Used

- **Pygame** (<https://www.pygame.org/>) - Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

Class and File Relationships (Flowchart)



List of Classes

- **Wall** - A class that defines the blocks of the maze and sets the color.
- **Block** - A class that defines the pellets in which PACMAN eats. This Class sets the color and the number of pellets on screen.
- **Player** - A class that defines PAC-MAN as the main player of the game. The player class contains PreviousDirection, Update, and ChangeSpeed modules. This class is launched by the single press of the arrow keys.
- **Ghost** - A class that defines the monsters of the game. They are objects created to kill PAC-MAN as they move to random positions within the maze. They also have a ChangeSpeed method and a defined list of coordinates that represent each one's position.
- **Controller** - A class that defines the “rules” and logic of the interlocking relationships between the other classes and UI, as well as establishes the data permanence of the high score, the sound effects and musical playback, and opens the actual game window itself. The collision functions, key bindings, and refresh rate are established here, and it calls each of the prior classes and their UI elements to “load” them when ran.

IV. Tasks and Responsibilities

Software Lead - Elizabeth Roedel

Worked as integration specialist by helping organize the code for the main game into the proper MVC format. Checked over both GUI and backend code to make sure the code was of high quality and that there were no breaks in the codes. She checked to make sure the code was concise and that variable names were consistent on both ends. Ran game test, wrote docstrings for each function and organized

Front End Specialist - Rachel Yap

Front-end lead conducted significant research on using pygame to create visual aspects such on-screen text and images. She used this information to design and program a consistent UI to help the player navigate the title screen, the instructions page, and the “GAME OVER” screen. In addition to implementing the wide majority of the visual element for the UI, she also collaborated with the Software Lead to create a jukebox function that played music and to add sound effects to the menu navigation buttons.

Back End Specialist - Megha Patel

The back end specialist helped with the “Model” portion of Pacman by writing the major classes that would be used in the main game, as well as implementing major pygame functionality into each of them. She hardcoded the movement of the ghosts as well as design the structure of game flow. She collaborated with the Front End Specialist in the implementation of the classes into our Controller file.

V. Testing

Testing Strategy

1) Game Testing

Once the Start screen appears, we test if once the screen is clicked the game screen appears. Before the player starts moving we check to make sure the score at the bottom is set equal to zero before any pac-dots are eaten. During game play, we test the movements of the player through the single-press of each arrow key. We make sure movement is constant if a key is held down. Throughout each movement, a pac-dot should be eaten by PAC-MAN and the score count increases by 1. Lastly, if the player comes in contact with a ghost during game play, PAC-MAN dies, the program quits, and the game over screen is displayed. We check that the window closes once the "X" in the top left corner is pressed.

Acceptance Test Procedure

Step	Procedure	Expected Results	Actual Results	✓ or AI
1	Run GUI	GUI Window appears with title Merry Christmas PAC-MAN with “Press screen to start”		
2	Click START	Window displays maze, ghosts, and pacman Displays food counter = 0		
3	Hold Up Arrow	Moves in upward direction		
4	Hold Down arrow	Moves in downward direction		
5	Hold Left arrow	Moves in left direction		
6	Hold Right arrow	Moves in right direction		
7	Pacman eats food	Food disappears and food counter increases by 1; display results		
8	Pacman collides with ghost	Kills Pacman; Display on screen: “Game Over”		
9	Click on X in right corner of the	Window is dismissed and program ends		

--	--	--	--	--