

Final Project

Design Document Draft

By James Rowe-Cooper for CSC221 Final Project

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

Introduction

Project Functionality

Design Process

Project Development

Pseudocode

This is for your pseudocode. Please provide it and explain it.

Flowchart

UML Diagram

Requirements

Introduction

This project is a simplified version of the classic arcade game Pac-Man, developed using Python and the Pygame library. It involves navigating a character called Pac through a board, collecting food pellets and avoiding walls. The game utilizes game development concepts like collision, timed movement, sprite rendering, and score tracking.

Project Functionality

Player Movement: Pac-Man moves one square at a time in a chosen direction and continues in that

Pellets & Scoring: Small pellets ('.') give 10 points; large power pellets ('o') give 50 points. Pellets disappear once collected.

Maze Design: A grid is created using a list where characters like #, ., and o represent walls, and food

Win Condition: When all food has been collected, the game ends with a win message.

Random Food Spawning Food can respawn at random intervals in empty spaces.

Tunnel: There is an opening on both sides of the maze where Pacman can enter and teleport to each side.

Graphics: Png used for Pac man and RGB colors are used for render walls, food, and background.

Design Process

Project Development

Pseudocode

Import Libraries: Pygame, sys, random

Initialize Pygame

Set constants: Screen width, screen height, cell size, fps, colors, starting direction, starting position, starting score.

Load screen and caption

Design Game Board

Draw Board

Draw Pacman

Define function to move Pac-Man

- If direction is LEFT and the space to the left is not a wall:

 - Move Pac-Man one space to the left

- Else if direction is RIGHT and the space to the right is not a wall:

 - Move Pac-Man one space to the right

- Else if direction is UP and the space above is not a wall:

 - Move Pac-Man one space up

- Else if direction is DOWN and the space below is not a wall:

 - Move Pac-Man one space down

- If Pac-Man lands on a small pellet ('.'):

 - Remove the pellet from the board

 - Add 10 points to the score

- Else if Pac-Man lands on a power pellet ('o'):

 - Remove the pellet from the board

 - Add 50 points to the score

Define function to check if all food is eaten

For each row in the game board:

- If the row contains a small pellet or a power pellet:

 - Return False (food still exists)

If no rows contain food:

Return True (all food is gone)

Begin main game loop:

Check for events:

 If the window is closed:

 Set running to False

 If a key is pressed:

 If the key is LEFT: set direction to LEFT

 If the key is RIGHT: set direction to RIGHT

 If the key is DOWN: set direction to DOWN

 If the key is UP: set direction to UP

 If a key is released:

 Set Pac-Man's direction to none

Check if all food is eaten:

 If true, display "You Win!" and end the game

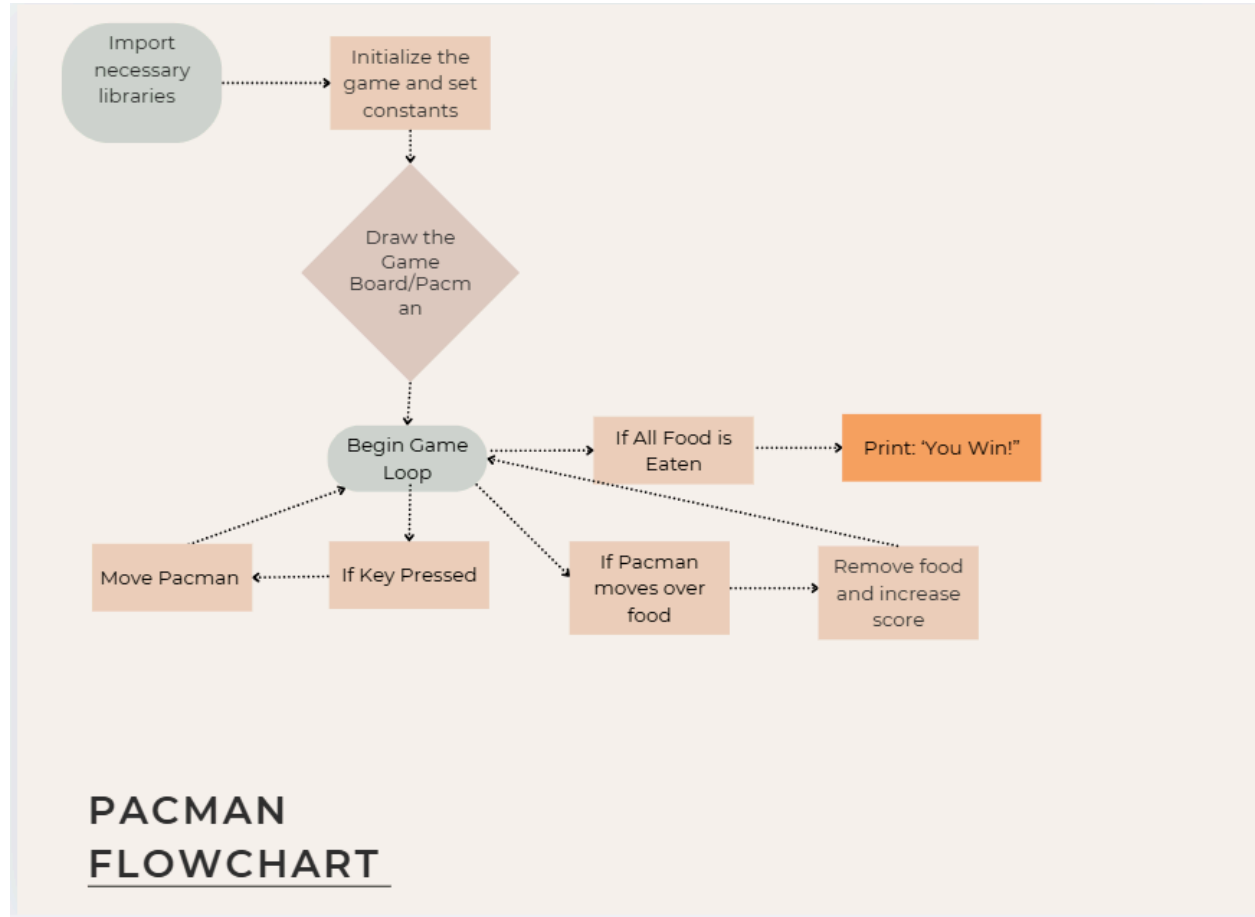
Clear the screen

Draw the game board

Draw Pac-Man

Exit the game and close the window

Flowchart



Requirements

This is for keeping track of the requirements you fulfilled during the final project. Please discuss each of the objectives/requirements listed in the final project assignment and how your final project meets/exceeds them.

Requirements:

☒ **Board Size / Play Area:**

- The game should be played on a window containing at least a 10-by-10 tile set
- Dots Should be in all valid titles for the Pac-man to be.

☐ **Pac-Man Movement:**

- The Pac-man should move in one of four directions (up, down, left, right).
- The player should control the direction using the arrow keys.
- **Incomplete:** There should be a path to the outside of the screen, and the Pac-man should be able to travel from one side to the other.

- ☐ Food Generation:
 - **Incomplete:** Food should appear randomly on the game screen after some random number of seconds.
- ☐ Collision detection:
 - Pac-Man's score increases when running into a dot or fruit
 - The dot disappears
 - **Incomplete:** The fruit disappears and the timer for the next fruit to spawn begins.
- ☐ Game Over and Score:
 - Display a game over message when all dots are collected.
 - **Incomplete:** Display the player's score, which is based on how much food and dots the Pac-man has collected.
- ☐ Restart Option:
 - **Incomplete:** There should be an option to replay the game after a game over.