

Exercise 6 Devlog

11/7:

- Preproduction
 - I decided to create a game similar to Pac-Man.
 - Major question to resolve:
 - How can I make the ghosts follow the player?
 - To-do:
 - Create a maze of barriers in the scene.
 - Find free Pac-Man and ghost sprites.
 - Find/implement sounds for Pac-Man eating pellets, Pac-Man eating power-pellets, Pac-Man eating ghosts, ghosts spawning, and ghosts eating Pac-Man.
 - Implement scoring increase when Pac-Man eats pellets, power-pellets, and ghosts.
 - Implement three lives and losing a life when a ghost eats Pac-Man.
 - If the player gets through eating all pellets and power-pellets before three lives run out, reload pellets and power-pellets (but don't reload score).
- Aesthetic Goal One: The game should be a challenge for the player, in terms of reaction-time and decision-making.
 - Signs of success:
 - The game is difficult but becomes easier as a player practices (i.e. the more times they play).
 - The game provides high enough stakes, meaning losing a life has a significant consequence.
 - Players want to win.
 - Players are able to tell how well or poorly they are performing.
 - Signs of failure:
 - The game is too easy to be engaging.
 - Players don't care about winning.
 - Players are not able to tell how well or poorly they are performing.
- Aesthetic Goal Two: The game should offer players a feeling of mindless escape, i.e. submission.
 - Signs of success:
 - The game mechanics are easy to learn and understand.
 - The game is difficult enough to engage/distract.
 - The game is fairly repetitive so it can be mindless.
 - Signs of failure:
 - The game requires too much critical thinking.
 - The game is too easy to be engaging.

- Core Mechanics:
 - Player moves through the maze
 - Player gets pellets
 - Player gets power pellets
 - Player avoids ghosts (usually)
 - Player chases and eats ghosts (when the ghosts are in chase mode)
- Core Loop:
 - Player moves to get pellets (raise score), player moves to escape ghosts (avoid losing lives), player gets power-pellet, player chases and eats ghosts (significantly raise score), player moves to get pellets (raise score) and escape ghosts (avoid losing lives).
 - This core loop serves my aesthetic purposes because it becomes easier as the player practices the timing of each action, it provides stakes and scoring for actions, and it is fairly repetitive despite its variation so it can provide mindless escape.

11/8:

- Created the scene using barriers and pellets with BoxCollider2Ds and power-pellets with CircleCollider2Ds
- Implemented player movement using arrow keys. I first attempted to constrain player movement to straight horizontal and vertical lines like in the original Pac-Man game. However, the effect was choppy than I wanted player movement to appear so I switched to free movement along the horizontal and vertical axes, constrained only by the maze.
- Implemented player eating pellets (on collision with pellets and power-pellets, the pellet or power-pellet game object is destroyed).
- Implemented reloading the scene if all the pellets and power-pellets are gone.

11/9

- Worked on creating a grid for the ghost movement to follow.

11/10

- Tried to implement the ghost movement using the grid but encountered numerous problems and errors.

11/11

- Decided to switch from grid-based ghost movement to point-based ghost movement.
- Added points to the scene at all the turning points of the maze.
- Implemented ghost movement by choosing random next points and lerp between the current point and the next point.

- Fixed an exception in the ghost script that the path of points object did not exist, despite having set an instance of it to a public field in the path script. Instead, I set the field in the ghost script, using FindObjectOfType().
- Fixed an error where the ghost direction was always reset to the same direction by moving the line where the ghost direction was initially decided from Update() to Start().
- Fixed an error where the last random number was never selected in all my uses of Random.Range() by adding one to the end of the range.
- Implemented scoring and the text that shows the score.
- Implemented lives, losing a life when colliding with a ghost, and the text that shows the number of lives remaining.
- Added the condition that the number of lives be greater than 0 to the code for reloading the scene when no pellets and power-pellets are left.
- Implemented an ending statement showing the final score when the number of lives equals zero and no pellets and power-pellets are left, or when the number of lives is less than zero.

11/12

- Changed the ending statement implementation so that the text shows up on a black screen.
- Added the ability to restart the game after it ends by pressing escape.
- Made the player speed slightly greater to avoid feelings of stagnation and ineffectiveness.
- Created a coroutine to make the player speed double for ten seconds after it collides with a power-pellet.
- Implemented a ghost changing color for ten seconds after it collides with a power-pellet, and during this time, adding 20 to the score and not subtracting a life when the player collides with it.
- Fixed errors where restarting from the game over screen did not reset the score and resulted in 2 lives, not 3, by adding reset/add functions.
- Added barriers to prevent ghosts from spawning on the player.
- Changed power-pellet time from 10 to 7 seconds to increase urgency.

11/14

- Fixed an exception where the second generation of instantiated ghost clones did not function correctly on collisions because their color was constantly set to the chase color (magenta).
- Set the normal color of all ghosts to red for convenience and clarity.
- Fixed a problem where later iterations of the ghosts moved too often because I had not set MoveTimer equal to Time.time in Start().

11/15

- Added sound effects to the player eating pellets, the player eating power-pellets, a ghost eating the player, and the player eating a ghost when in chase mode.
- Fixed null reference exception by checking if HitData is null before I access it.
- Postmortem:
 - I originally set out to create a game as similar to Pac-Man as possible. My goals did not change significantly but I did simplify the ghost movement. The actual Pac-Man ghost AI is more complicated than I had anticipated and varies for every ghost. Instead, I made my ghost movement work by picking random spots to lerp to throughout the maze.
 - I also decided not to implement Pac-Man and ghost sprites because the game is not exactly the same and I think having the same sprites would therefore be confusing.
 - Otherwise, my goals remained pretty similar to those expressed in my preproduction design. I successfully implemented the pellets, power-pellets, lives, and scoring, as well as the required functionality and sound effects.
 - The hardest part of this assignment was implementing the ghost movement, and though I am proud of how it turned out, I wish I had not spent so much time trying to make it work with a grid, which ultimately failed. This is something I very much want to learn and work on in a future Unity project.
 - While making this project, I learned a lot about points, coroutines, and raycasting in Unity.