

“Pac-Man Desktop Port” Project Proposal

[Strange_Team_Name_Here]

(Rishi Basdeo, Ryan Tarr, Akshay Antony, Shreyas Misra)

Why Pac-Man

Pac-Man is a classic game where the player guides Pac-Man through a maze while collecting pellets and avoiding ghosts in Chase mode. Pac-Man can eat power-pellets which causes the ghosts to go into Scatter mode, allowing him to eat them. Although it might seem simple at first, Pac-Man is more computationally intensive than one would expect due to the game animations and unique ghost trajectories. Hence, we chose Pac-Man due to its focus on background computation over graphic complexity. Furthermore, Pac-Man was one of the most played games in the 1980s and is still widely played all over the world, thereby imperative to ensure that the game can still be played in all modern operating systems so that its legacy lives on.

Program Description

Our main goal in this project is to replicate the original Pac-Man gameplay very closely to how it was implemented in 1980 when it was first released. We intend for our program to mimic the user experience, functionality, and visualization of the original game. This includes properly animating both Pac-Man and each of the four ghosts as they navigate through the map, implementing pellets and power pellets as in the original gameplay, programming path planning for the ghosts to navigate to the proper target locations, tracking of the current score, and high scores, including the original intermission between levels, and more.

We plan to use OpenGL and the *fssimplewindow* library for gameplay visualization and the *yssimplesound* library for gameplay audio. We may also use the *simplebitmap* library for graphics.

Project Challenges

As a preface to these goals, a key emphasis will be placed on the modularization of this project, with the intent that game components such as map generation and score tracking can be further used for other game ports. Hopefully, this will encourage others to continue preserving historic games.

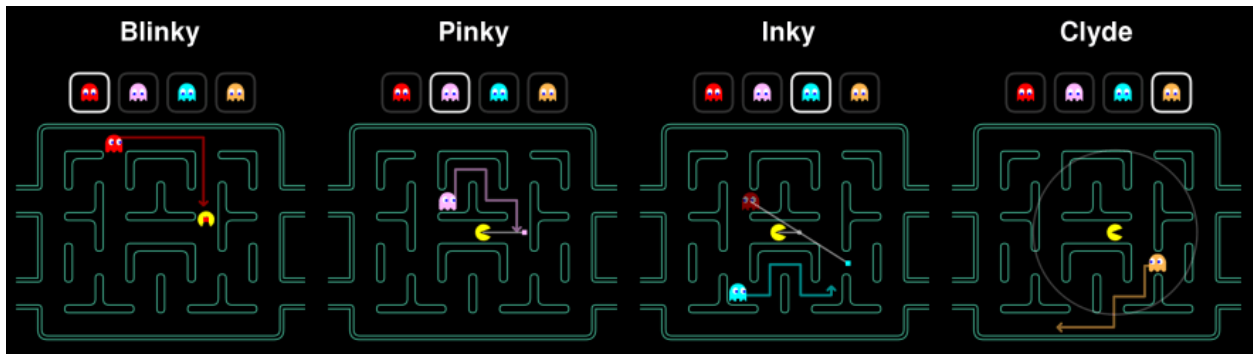
With the goal to replicate the original Pac-Man game as closely as possible, there are two main technical challenges. The first of these is the transition animations that take place between levels. Many current Pac-Man ports are primarily faithful to the original but lack these details that made it such a groundbreaking game. As such, we want to place special attention on these sections. Timing will be critical in this area.

The second main technical challenge is in making sure that the ghosts follow their intended chase patterns in both game modes. Their behaviors are summarized in the chart and image below.

Table 1: Pac-Man Ghost Behavior

	Blinky	Inky	Pinky	Clyde
Chase Mode	Direct Follow	Double Blinky's Target Distance	Follows 2 Dots Ahead	Remain in Bottom Left of 8-Dot Radius
Scatter Mode	Top Right	Bottom Right	Top Left	Bottom Left

Figure 1: Pac-Man Ghost Behavior During Chase Mode [1]



However, this behavior only determines their target locations, not how they will be achieved. As such, the second subcomponent of ghost behavior is their path planning. We will implement an A* algorithm for this, which is one of the most widely-accepted path planning strategies.

Other key challenges include menu creation, score tracking, and offline scoreboard storage, as well as game asset creation such as power-ups, power-pellet tracking, and map generation.

Potential Stretch Goals

If we can complete all the goals mentioned above before the timeline, we would like to work on the following:

1. In addition to the default behavior of ghosts, we will apply different targeting algorithms for new ghosts. For instance, Blinky's goal is double Inky's target. We plan to add new ghosts whose goals will be even beyond Blinky's, eventually making a more challenging game.
2. On the graphics side of the game, we would like to introduce color schemes and animation styles used in later releases.

References:

<https://dev.to/code2bits/pac-man-patterns--ghost-movement-strategy-pattern-1k1a>