**Pacman Game**

Benjamin Kwon

CSC-17B-44085

June 3, 2019

|  |  |
| --- | --- |
| **Member** | **Role** |
| Benjamin Kwon | Ghost behavior |
| James Hartley | Game engine, game mechanic |
| Sebastian Hall | Login/Signup page, front end graphic |
| Ghislain Muberwa | High score page |
| Samuel J Hernandez | Database |
| Christopher Schlenker | Database connection between server and front end |

**1. Introduction**

Pacman is an arcade game and is regarded as one of the most influential video games of all time. For this project, the group decided to try and create an accurate representation of the original Pacman. The project was created using the Phaser game framework, which allows fast rendering of HTML5 games.

**1a. Personal Contribution**

Regarding individual work in the group, I have primarily focused on game development. Some of the features I have worked on include:

Editing the tile map to include power pellets

Implementing game mechanics and logic:

Pacman eating power pellet and the subsequent effects

Pacman eating ghost

Ghost movement and behavior

Four patterns: Chase, Scatter, Scared, Eaten

Added animations for ghost when scared

Added sound effects

**2. Description/Gameplay**

As an arcade game, the gameplay and rules are relatively simple. The basic controls make use of each directional arrow key on a standard keyboard.

*Table 1. Basic Controls*

|  |  |
| --- | --- |
| **Key** | **Action** |
| Up arrow | Move up |
| Down arrow | Move down |
| Right arrow | Move right |
| Left arrow | Move left |

The player, playing as Pacman, finds him or herself in an enclosed maze containing four ghosts in total. Each of these ghosts are programmed to chase Pacman with their own unique method. The details of these methods are explained more in the following table:

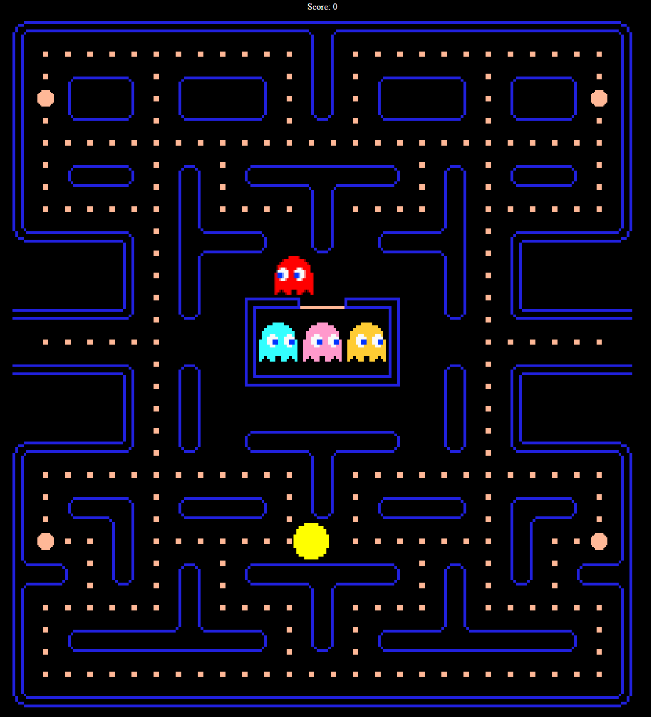
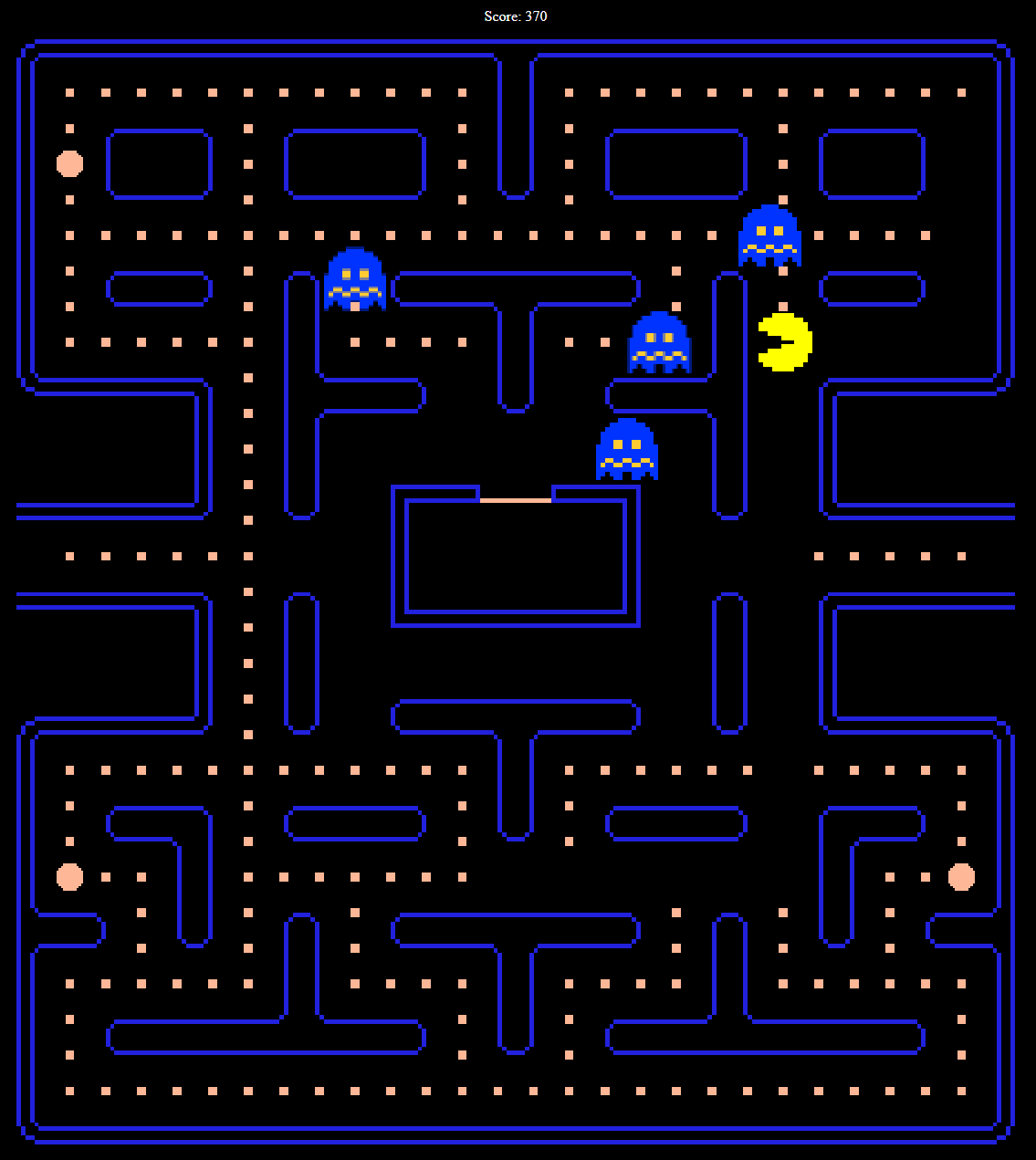
*Table 2. Ghosts and their target tile (pathfinding algorithm)*

|  |  |  |
| --- | --- | --- |
| **Ghost name** | **Color of ghost** | **Target tile** |
| Blinky | Red | Tile Pacman is on |
| Pinky | Pink | 4 tiles in front of Pacman |
| Inky | Teal | 1. Get position 2 tiles in front of Pacman  2. Double distance Blinky is away from (1)  The resulting vector end point is target tile |
| Clyde | Orange | If more than 8 tiles away from Pacman,  target tile is identical to Blinky’s  If less than 8 tiles away from Pacman,  flees to bottom left portion of map |

The maze contains tiny dots called pellets as well four larger dots called power pills. The goal of the player is to collect every pellet without being caught by a ghost. If Pacman collects a power pill, Pacman becomes invincible and will not die if contact is made with a ghost. Instead, it will eat the ghost, which will run away back to the center of the map and revive a few seconds later. While Pacman is invincible, the color of every ghost will turn blue, to signal their vulnerability.

To goal of the game is the obtain the highest score possible. Eating pellets, power pills, and ghosts all give points in increasing order. The player must try to collect every pellet on the map without being caught by a ghost. If caught, the player will lose a life. The total lives or chances available is three.

To play the game, users must register an account and log in. Once logged in, they will be granted access to the game. When the player loses all his lives and the game is over, the score will be sent to the database and be connected with the player’s username. If the score is within the top ten current scores, it will be shown on the webpage hosting the game.

*A capture of the starting gameplay* *A capture showing the effects after eating a power pill*

**3. Midterm Concepts**

Code structure/Objects/Model-View-Controller

The structure of the code is divided up into distinct sections. When the game runs, it is initialized, preloaded, and then created.

In the initialize function, the libraries and graphics that run the game are scaled and rendered. In the preload function, all the assets of the game including images, sprites, tilemaps, and audio are loaded. Lastly, in the create function, using the loaded assets, all the characters and map data are constructed and placed. Here, Pacman and the ghosts are constructed objects of their respective class. This can be seen in the MainGame.js file.

The main structure of the game engine is the update loop, which continuously calls on the necessary functions required to keep up with a real-time game. Here, the positions of each sprite are tracked, and the movement of each sprite is determined by the current state of the game. For example, the keys (controller) pressed by the player are constantly being checked and sent to the model, which determines the path taken by the player. This is then shown through the view, which is what the user sees while playing the game. The update loops can be seen in Pacman.js and Ghost.js files.

**4. Final Concepts**

The project opens to a login screen where a user can enter their credential to start playing or create a new user account where after the information is successfully entered will open to the game which starts immediately. There is a list of high scores from all previous players ranking from highest score on top to bottom with the lowest with their chosen username next to their score. The game has no pause or delay in game starting so as soon as the user signs in the game has started. The player starts with three lives and the goal is for the player to eat all pellets on the level to complete the level before the ghosts eat pacman(player) three times and its game over.

There are bigger pellets that allow pacman to eat the ghosts as a defense mechanism to fight them. Each ghost have their own movement pattern they follow and each times the player clears the level, the ghosts movements will become faster and improve more. The board is the same with each level, but there is no level indicator to identify the player how far they are. The score gets larger the longer the player has lives and when the player is eaten when they have zero lives, their acquired score during the play is saved in a database, the high score board is updated and a new game automatically starts where the game resets the board, ghost movement pattern score, and lives to initial state. There is a sign out button on the top left side of the screen for the player to quit the game.

**Database SQL**

In the database schema called ‘pacman,’ the following tables are stored:

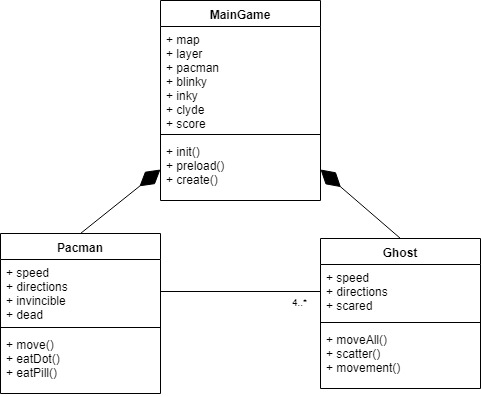
Entities: entity\_users, entity\_highscore

Xref: xref\_user\_highscore

In the database, the username and high scores are connected to each other through a cross reference table. The top 10 high scores in descending order along with the associated username are shown on the game page.

**5. Diagrams**

UML Class Diagram



Flowchart

A close up of text on a white background

Description automatically generated

**6. Code**

The source files for the project can be found at the group’s GitHub repository:

<https://github.com/SebastianHall69/PacmanGroup2>

A web server will be needed to properly play the game on a browser.

The game is also hosted on the following webserver:

<http://www.mischiefcrew.com/pacman/LoginPage/>