Change Log

# V1

- Added Pac-Man.

- Added Blinky.

- Added Pinky.

- Added Clyde.

- Added Start Screen.

- Added Intro Music.

- Added icons (mute icon).

# V2

- Added Pellets .

- Added some wall sprites.

- Fixed ratio issues (sprites were 24x24 whereas tile size was 8x8). Fixed by making the tiles 12x12 so that they appeared to match up with the size of the sprites without scaling the sprites strangely.

# V3

- Added remaining wall sprites by finishing algorithm

- Fixed issue where Pac-Man appeared to be stopping before walls. Fixed by creating a separate smaller hitbox for Pac-Man.

# V4

- Added algorithm that pushed Pac-Man out of a wall if he collided with it. Pac-Man would appear inside some walls and also glitch through them, so this reduced the number of times he glitched through walls but did not eliminate the issue entirely.

# V5

- Added Inky.

- Fixed glitching through wall issue. Fixed by changing the update tile function to update the sprites current tile-based on position. Before, (for whatever strange reason) a prediction of what the next tile was made and if the sprite collided with this tile then the tile would be updated. This meant that Pac-Man would not collide with tiles if the current tile did not update. Now, occasionally Pac-Man freezes on a wall but no longer goes through.

-Fixed input lag by checking whether a key is pressed down rather than checking whether the key had been pressed once in a given frame.

# V6

- Added power pellets.

- Added scared mode for ghosts.

- Added Pac-Man death animation.

- Fixed font glitch. For some reason PyGame didn’t recognise TTF files that weren’t installed on the computer. Now this is fixed (likely through a PyGame update as I didn’t change anything).

# V7

- Added lives. Now if Pac-Man dies the game will begin again until his lives run out.

- Added score.

-Added tunnel. Sprites can now go through the tunnel allowing them to appear on the other side of the maze.

- Fixed Pac-Man-ghost collision issue. This involved the ghosts catching Pac-Man with the slightest contact (if at all) Fixed by colliding only when the ghost hit box collides with the centre of Pac-man which matches the original game more.

# V8

- Added 1-up indicator (whatever that is)

- Added little score indicators. This is for when Pac-Man eats a ghost and gives details on how many points the player received for each ghost.

- Added ‘Ready’ and ‘Game-over’ text. This is at the beginning of a game or round and at the end of a game.

- Added flashing map at end of a level.

- Added sounds.

- Added flashing power pellets.

- Fixed ghost cornering issue. Ghosts would originally be able to ‘corner’, whereby they would turn a corner without being centred perfectly in the middle of a tile. This allowed them to be as fast as Pac-Man around corners which is not true to the original game. Fixed by not allowing x-direction movement to take over until exact y value has been reached and vice versa.

# V9

- Added Elroy.

- Fixed (partially) mode switching issue. This would swap the order in which chase and scatter happened meaning the ghosts would scatter for a far longer time. Reduced the frequency of the problem occurring.

# V10

- Added multiplayer splash screen.

- Added experimental multiplayer game mode. Works by sending every object on the screen as a pickled object, however this is far too slow and the packets are far too big to send with python sockets.

- Added game lobbies. These can now be created and joined.

- Reworked text classes. They now all take a parameter win\_scale and this multiplier is applied at the lowest level for easier usage.

# V11

- Added names in the lobby

- Added score in lobby

- Fixed crashing that would occur when players disconnected from the lobby. Fixed by testing for disconnections and correctly closing connections.

- Fixed crashing that would occur when the server disconnected from the lobby. Fixed by testing for lost connection with the server, closing the connection then exiting the lobby screen.

- Fixed all input lag. This occurred because I was calling pg.event.get() multiple times so certain events were being lost in other calls. Fixed by limiting the call to once every frame and passing this into every program run call.

# V12

- Added online multiplayer sprites.

- Added online multiplayer. Following bugs identified: when a ghost is eaten they do not move. Ghosts can sometimes get stuck on or in walls.

# V13

- Added function to transport ghosts into and out of the respawn point which I have now implemented into multiplayer.

- Added points for ghosts. Points are given for proximity to Pac-Man and for when they catch Pac-Man.

- Added countdown for the lobby.

- Added AI for when there are not enough human players.

- Added ability for ghosts to become Pac-Man when they catch him.

- Added spotlight to ghost view so they can only see 12 tiles around them.

- Fixed ghosts getting stuck in and on walls.

- Fixed bug whereby Pac-Man’s skin would not update (or not update quick enough). Occurred because the move data was being updated with None. This happened because the player\_data dictionary in the connection class was updating the move parameter in the players’ dictionary, but was not itself actually being used (it was redundant code I had worked around and not removed.

# V14

- Added score cap to the online matches

- Added error message class.

- Added awarding places (based on score) to players at the end of a multiplayer game.

- Added setting file so win\_scale can be changed when complied.

- Added error message when server or client disconnects.

# V15

- Added local database for storing game history (used for retrieving and saving high scores).

- Added Login page (doesn’t function yet).

- Added settings page (doesn’t function yet).

- Added comments to sprites file.

- Fixed glitch where maze would turn white. This occurred because the function that switched the colour of the maze at the end of a round was broken.

- Fixed glitch where ghosts would turn permanently white after Pac-Man ate a power pellet while their skin was already white. This occurred because the ‘elif’ statement that handled power pellets being eaten while ghosts were already scared didn’t change the skin pack back to blue. Fixed by adding a line to change the skins back to blue.

- Fixed issue where sometimes the ghost mode timings would be inverted. Did this by changing the statement allowing the swapping of buffer modes from if self. mode == self.scatter: to if self.buffer\_mode == self.scatter:

- Fixed issue where square brackets and asterisks wouldn’t display. Fixed by editing TTF file to include full stops square brackets and asterisks using Font Forge.

- Fixed ‘win\_scale’ usage so it is now only at the lowest level.

- Fixed small programming practises: having returns for things that don’t obviously need them. For example, instead of having program =icon.action(). Run action first then say program=icon.target\_program. Must continue this in future versions.

# V16

- Added fully functioning login and account creation system with existing database.

- Added password and username checker.

- Added tab use for login and sign up.

- Added the ability to press enter for sign up and login buttons.

- Added a slider class for settings.

- Added sound and win scale sliders in the settings screen.

- Added accounts page (not statistics yet).

- Added comments on every code file.

- Changed highlight colours on start screen choices.

- Fixed font spacing issue. Sometimes made it look like there were spaces between words when letters were in a certain order. Fixed by standardizing the spaces around each letter in the TTF using Font Forge.

# V17

- Added statistics to accounts page.

- Added the ability to log out via the accounts page

- Added ability to save high score via 3 initials after classic game.

- Added high score page (using high scores saved using 3 initials).

- Added ability to stay logged in. Did this by saving username and password to settings json file. This can also be turned off while logged in, but no turned back on again (for security).

- Added scrolling words class which will be used in the story mode class.

- Continued to change code to use more consistent checking of variables. For example, instead of Pac-Man’s state being returned by level, ‘Classic’ simply calls the Pac-Man object directly.

- Fixed volume level differences within the classic game mode (can now be controlled via game sound setting)

- Fixed issue when pressing tab to continue onto the next box when logging in or signing up. This occurred if the user clicked to highlight the first box and then pressed tab afterwards. Fixed by

- Fixed issue whereby the game would crash when a user signed up and then logged in before playing a game. This occurred as the ‘get\_statistics’ function returned nothing when the there was no game data. Fixed by adding a message ‘Statistics will appear here once you play a game’ to the accounts page if the function returns None.

- Began directory reformatting paths. Some now use os.path.join()

# V18 – version used for first draft

- Added mazes to the database. Did this using just one column in the mazes table and serialising the 2D list using JSON as I will not ever need to query a specific part of the maze. It will always all be needed.

- Added a text box that has scrolling words (letters that reveal themselves one by one). It is capable of working out when a new line is needed (if the word after is too long) and can also have multiple pages if the sentence as a whole is too big for the textbox.

- Added story mode. Did this by copying the level class from the single player script then taking out features like high score, 1UP, but most importantly lives. I have added for levels so far that repeat until you complete them and have added scrolling text boxes for each one to explain the game’s mechanics.

- Added proper path drawing to ghost class (shows the ghosts’ path but using a line that can adapt to the direction rather than squares). It is also colour coded to the particular ghost and is displayed between the maze and the pellets.

- Changed a few methods in multiplayer that were almost static to be static and functions instead.

- Fixed issue where Elroy would not actually activate. This happened because I forgot to do the relevant pellet checks for when Blinky became Elroy. Fixed by adding these checks into the level run method.

- Continued reformatting directory paths. Most now use os.path.join()

# V19 – not yet completed

- Reformatted directory paths. They now use os.path.join().

- Change the input box class to use the word and box class.

- Reworked collision mechanics using Pygame meshes which I didn’t know existed!

- Reworked maze so that you no longer have to put +3 on the y-coordinate

- Added fruit

- Larger spotlights on multiplayer when a ghost eats fruit.

V20

- Fixed tunnel issues.

-