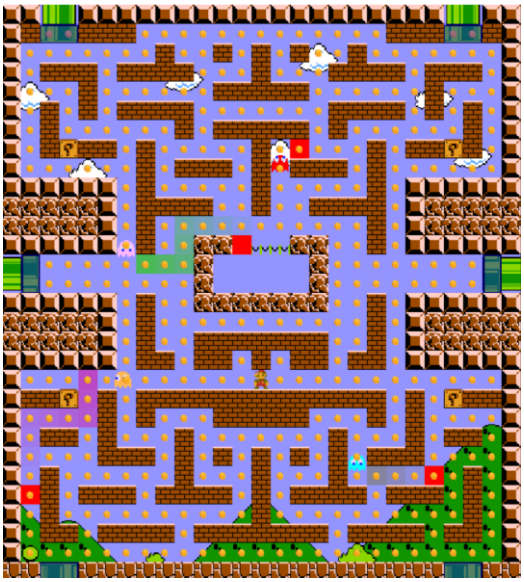
Mash-Room



For this end assignment we have derived a game from Mario and Pacman which we call: “Mash-Room”. The idea is that you are Mario within a Mario-themed world and physics but the level is Pacman-shaped and you have the ghosts and objective of Pacman, namely collecting all the coins.

Interactions:

Of course you can play our game by using the plain old “WASD” keys (modify with shift to run instead of walk), but that would be boring. On start-up, the game also initializes your camera and body tracking using OpenCV and MediaPipe. You can run left and right by pointing left and right to your camera, and you can jump by opening and closing your mouth.

The ghosts are loosely based on the original Pacman ghosts. Therefore they alternate between pathfinding to you and scatter mode, where each one will choose a random square in ‘their’ corner and path find to that. Half of the ghosts use the A* path find algorithm and the other half use the greedy search algorithm. They also path find to different positions around you based on the original 4 ghosts behaviour and try to ‘trap’ you with that.

The interactions between the player (Mario) and the level are that the ghost will deduct lives of the player if they hit you, that you can walk into the pipes to get teleported to the other paired pipe, you can jump against the bottom of the question blocks to activate ‘scared ghost’ mode for a certain amount of time, where the ghosts will flee from you instead of scattering or pathfinding and where you can ‘kill’ the ghosts by walking into them whereafter they first need to go to their home-base in the middle. You can of course also collect the coins around the level, which if you manage to collect them all satisfies the goal of the game and ends the level!

We have built this game in a tiled fashion. This means that almost everything is an instance of the Tile class. This also means that we can quite easily build new levels in a tile-editor like ‘Tiled’ and export that to a csv file, as our program reads the csv file that you export from tiled and cuts with that the right frames out of a sprite.

We use MediaPipe for hand and face tracking off the webcam. This is a library built by Google with which we can quite easily get the coordinates from the features of your body. It uses OpenCV to track your features. These coordinates are then used to detect the gestures of pointing and mouth opening.

Classes:

Tiles: Our whole image-system is based around tiles. In the following overview inheritance is relayed by which indentation the Tile has (very python-esque).

- Tile (keeps track of neighbours, position, drawable and overrides the __eq__ and __repr__)
    - SpriteTile (adds an image to the tile, and the draw method)
        - AnimatableTile (adds the ability of animated frames to a SpriteTile)
        - StaticTile (Deals with the rotated flags of Tiled and has one image)
    - PassageTile (Tile For empty spaces, the spaces ghosts can search in, therefore implements useful methods for that, and overrides __lt__)

Game:

Initializes and updates all the main components, makes the window and initializes the tracking thread and event handler. Calls Level, Player, UI, event Handler, Coin and Ghost. Also keeps track of dt with the clock.

Level:

Creates instances of Tiles in all the right places according to our CSV data, which creates all the staticTiles for the scenery etc. Also upates everything that is in a level. Calls Player, Coin, Ghost

EventHandler:

Gets the game and event from the game class and deals with all input accordingly.
Calls methods of Player, and Game for resize events.

Player(AnimatableTile):

Handles the player, player movement and all collisions with other objects.

Calls methods of Ghost, Coin and Pipe_Head

Ghost(AnimatableTile):

Parent class of all different ghosts, deals with animation and pathfinding-helpers.

Pinky(Ghost), Inky(Ghost), Blinky(Ghost) and Clyde(Ghost):

Subclasses of Ghost. Implement the different pathfinding targets of the different ghosts.

Coin(AnimatableTile):

Deals with being a coin.

QuestionBlock(AnimatableTile):

Implements the animation of when it gets hit by the player, then calls all the ghosts to get scared. Can only be hit once.

Pipe_Head(StaticTile):

The head of the transportation pipes, know which ones they are linked to. When collided with the Player teleports itself to the position of the paired pipe.

Camera:

Gets the frames from your webcam.

TrackerThread:

Deals with the second thread we need for the webcam tracking. Calls event Handler accordingly.

Face Tracker:

Uses the MediaPipe model to get the positions of your mouth and determine when it is open. Calls the event handler when input is detected.

Face:

Keeps track of the features of your face.

HandTracker:

Uses the MediaPipe model to get the positions of your hands and determine when it is open. Makes two instances of Hand. Calls the event handler when input is detected.

Hand:

Keeps track of a position of a hand and if that hand is pointing a certain way.