Covi-Wars

COP290 Assignment

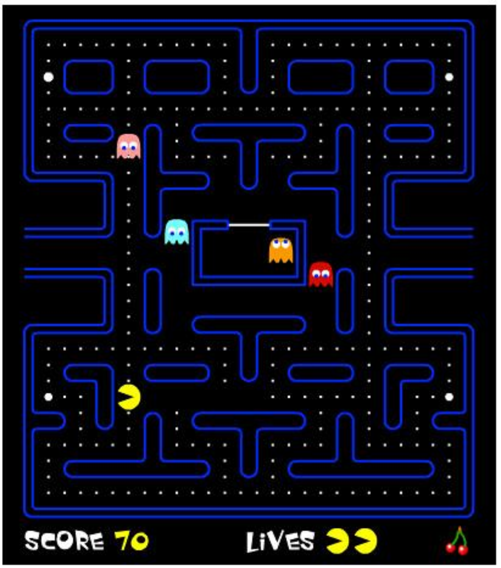
Supervisor: Prof. Rijurekha Sen

Shrey J. Patel
2019CS10400

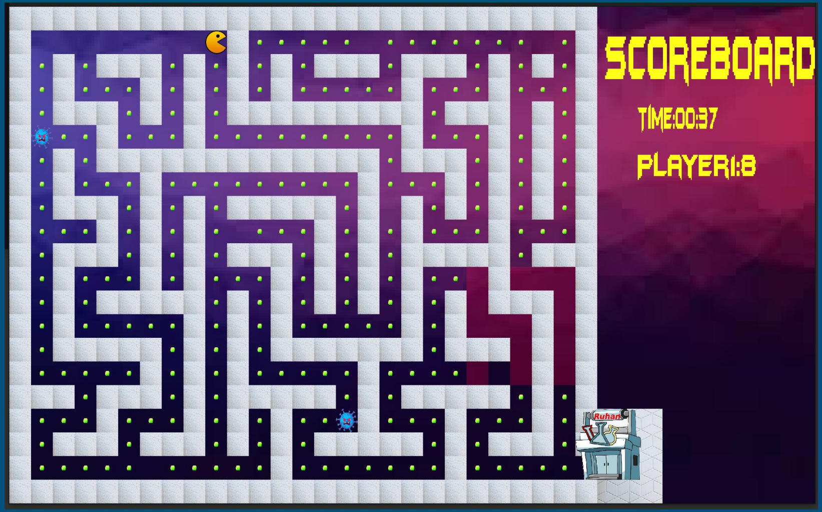
Rahul Chhabra
2019CS11016

What is Pacman?

- Pacman is a classic maze solver game in which the main objective is to navigate through a maze, while avoiding various kinds of enemies and collecting points by eating dots.
- The main character called “pacman” can also pick up some power-ups or boosters along the maze which can be used to collect extra points or to grant the power to disable the enemies temporarily.



Our variant: Covi-Wars



- The theme of our game is based on the current situation of COVID-19 pandemic.
- Our hero is a healthy civilian which faces the corona-virus as his enemy.
- And masks, vaccines and medicines help our hero to fight against the virus.

Rules and Instructions

1. Single-player mode:
    - The single player game has a progression system, with 5 levels of increasing difficulty.
    - Any level can be completed after scoring a sufficient number of points, by eating the tablets laid throughout the empty parts of the maze.
    - After collecting the required amount of tablets, a key appears and collecting it opens a door to the next level.
    - The player wins after completing all five levels.

2. Multiplayer mode:
    - In case of multiplayer mode, there is only one level with randomised maze, in which multiple players try to survive and compete against one another.
    - The player who outlasts all the other players, while surviving all the enemies, wins.
    - Or if multiple players are alive until all the tablets of the maze are eaten, then the player with the highest score wins.

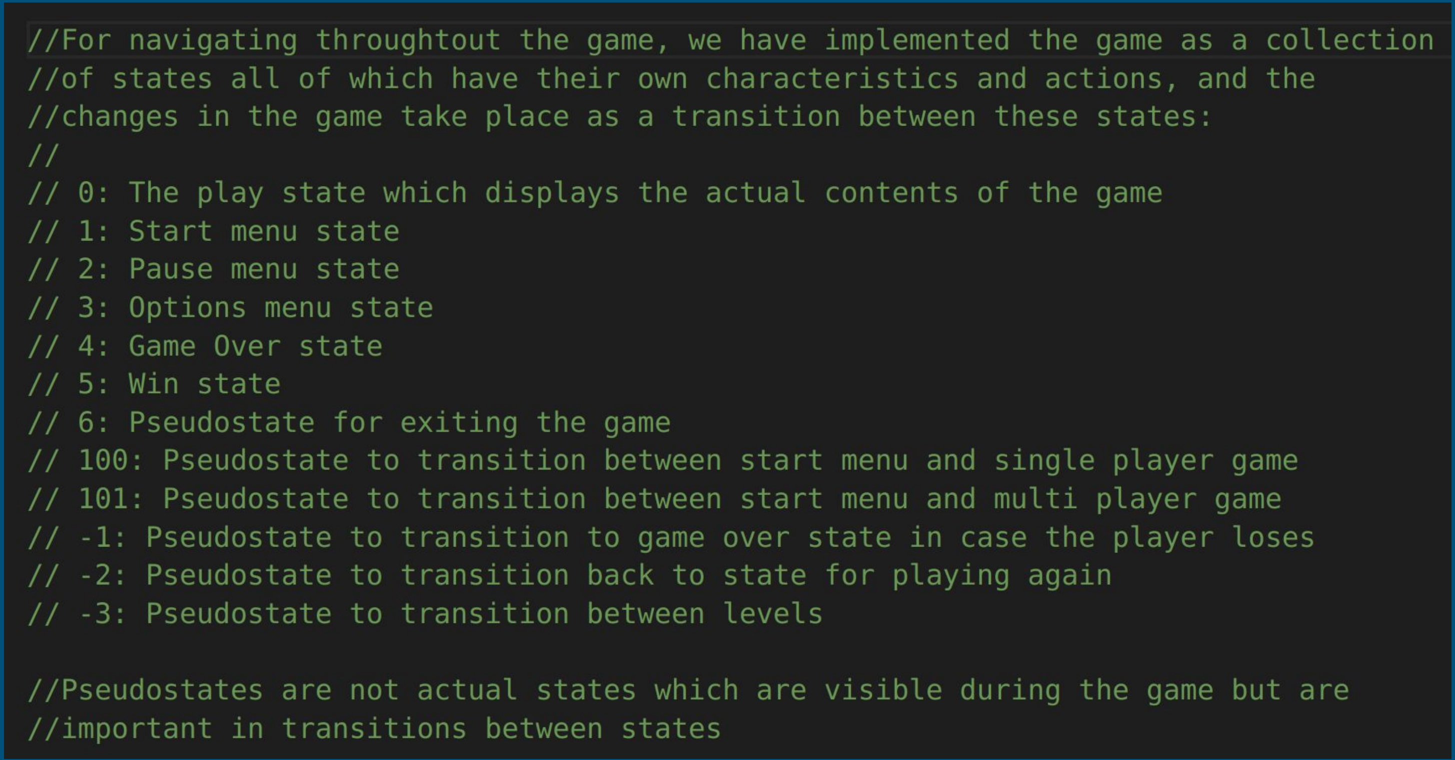
Common Instructions

- In both modes, the player has to avoid various hazardous objects to survive. Collision with them decreases the health of our character and the player loses if the health of his/her character reaches zero.
- However, there are various power-ups laid spread across the maze which give different abilities to the pacman which may help to survive.
- In this game, we have adopted a unique way to calculate the health of our character, where the current health is equal to the number of tablets collected, so the player with greater score will have better survivability.
- This is a fair way to track health as it gives more chance to a player who has accumulated more points, thus rewarding hard-work.

Game Design and Implementation

- The game has been designed as a collection of states, each of which have their own methods to handle events(like mouse/keyboard inputs, requests and responses through a network, etc.), process and update the current state and render the graphical elements on the screen.
- All the states have their own local data which is displayed on the screen using the local render function. This local data is sometimes passed between different states to take decisions which span several states.
- The main benefit of this design is modularity in code and the ease in implementation and debugging.

Game States



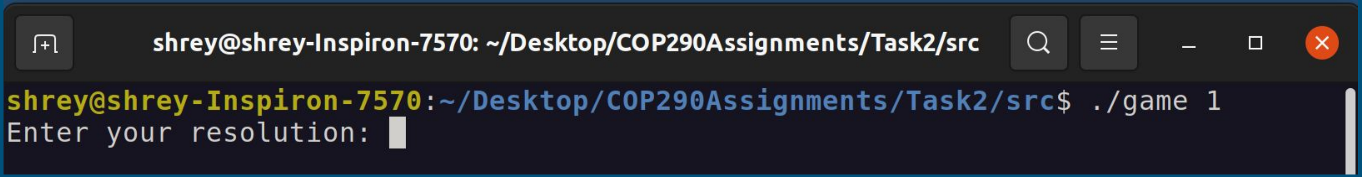
```
//For navigating throughtout the game, we have implemented the game as a collection
//of states all of which have their own characteristics and actions, and the
//changes in the game take place as a transition between these states:
//
// 0: The play state which displays the actual contents of the game
// 1: Start menu state
// 2: Pause menu state
// 3: Options menu state
// 4: Game Over state
// 5: Win state
// 6: Pseudostate for exiting the game
// 100: Pseudostate to transition between start menu and single player game
// 101: Pseudostate to transition between start menu and multi player game
// -1: Pseudostate to transition to game over state in case the player loses
// -2: Pseudostate to transition back to state for playing again
// -3: Pseudostate to transition between levels

//Pseudostates are not actual states which are visible during the game but are
//important in transitions between states
```

Resolution Independence

One of the most important features of our game is that it can be scaled to any standard resolution without clipping. The desired resolution can be chosen from the command line while executing the game. (Reference resolution = 3700 x 2100)

Before entering the game, the user will be prompted to enter their desired resolution(space separated format) in the command line:



shrey@shrey-Inspiron-7570: ~/Desktop/COP290Assignments/Task2/src

```
shrey@shrey-Inspiron-7570:~/Desktop/COP290Assignments/Task2/src$ ./game 1
Enter your resolution: 
```

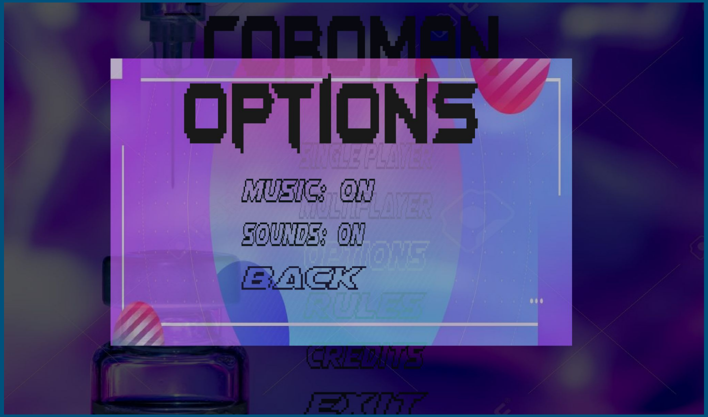
User Interface

- The most important part of any graphics related program like a game is its user interface (GUI) which provides a medium for the user to interact with the game.
- There are many menu states which help the player to navigate through the game.
- We have implemented two types of menus:
    a. Complete menus
    b. Pop-up menus

Start Menu

- This menu will be used to create the play state of the game in both the single player and the multiplayer mode. It also gives access to the options pop-up menu which is used to change global settings of the game, like turning the music and/or sounds on or off, etc.
- Choosing the single player button starts a single player game from the first level.
- Choosing the multiplayer button opens a lobby pop-up menu which awaits other players and transports all the players to their play states simultaneously once all the players have joined.
- We have made

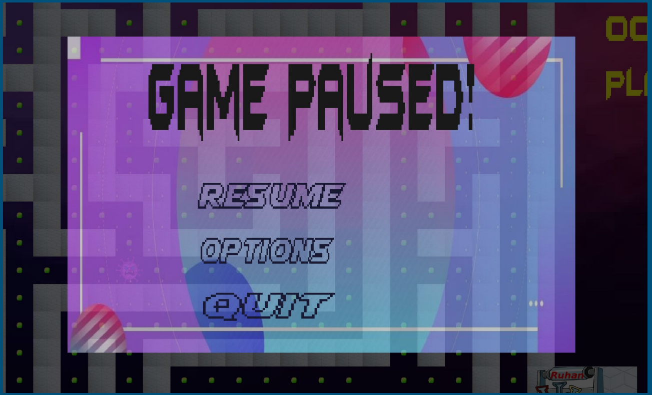
Pop-up Menus



An options pop-up menu in the foreground of the start menu

- Some menus have less impact on the overall state transition of the game. We have converted such menus into pop-up menus, which don't have their own state, but are considered to be objects of the current state.
- These pop-up menus are faster and easier to access as no state transitions are involved. They also don't occupy the whole screen, but open in a foreground, while all the rendered elements of the current state are still visible in the back of the pop-up menu.

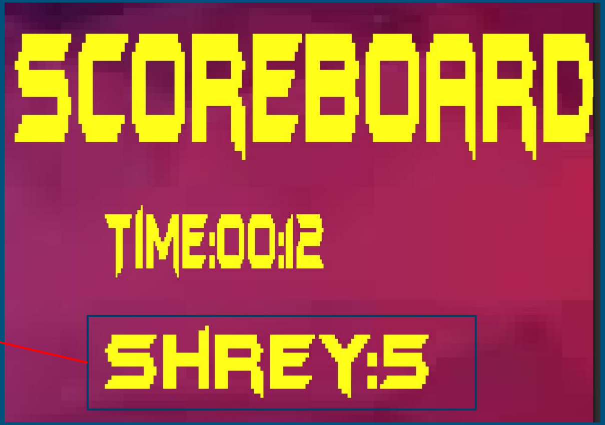
- Because of this, a pop-up menu feels more connected and less cluttered as compared to the start menu which is disconnected from other states.



A pause pop-up menu in the foreground of the play state

The Scoreboard

- Another important piece of user interface is the scoreboard which keeps track of the time as well the scores of all the players playing the game.
- This display is especially important because of out decision to keep the health the same as the score for any player. So, in a way, the scoreboard is also the health bar.



SCOREBOARD

TIME:00:12

SHREY:5

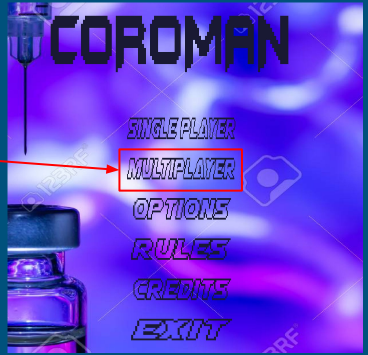
Score tracker that also acts as a health bar

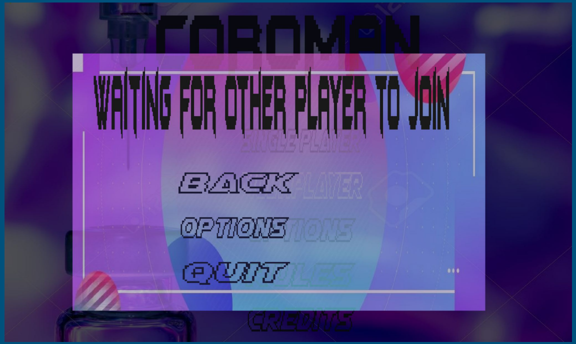
Guide to start a multiplayer game

The player who is hosting the game, i.e. the server, should start their game first. After all the players have reached the start menu,

Step 1:

The first step for all the players is to click the “Multiplayer” button on the menu. It doesn’t matter who presses this button in what order, as we have made our network manager to work symmetrically without any lag. The only important thing is the proper initialisation of the network manager, for which the server should be “compiled and run” first. The entry into the game can be in any order.



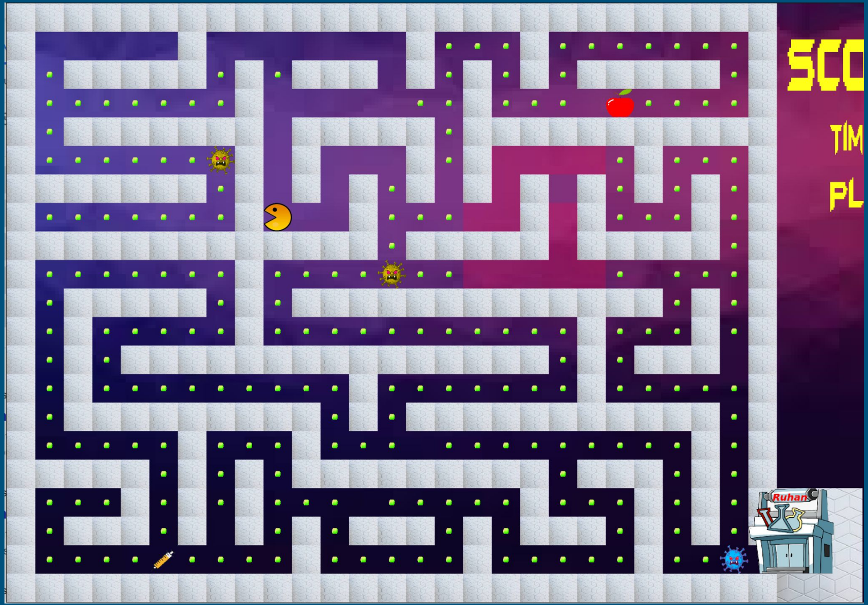


Step 2:

This opens a lobby pop-up menu, which waits for all the players to join. Once all the players have joined, they will automatically be transported to the play state.

The Play State

The main state of the game which contains the gameplay elements like the maze, the AI/program controlled entities like the enemies, powerups, etc.

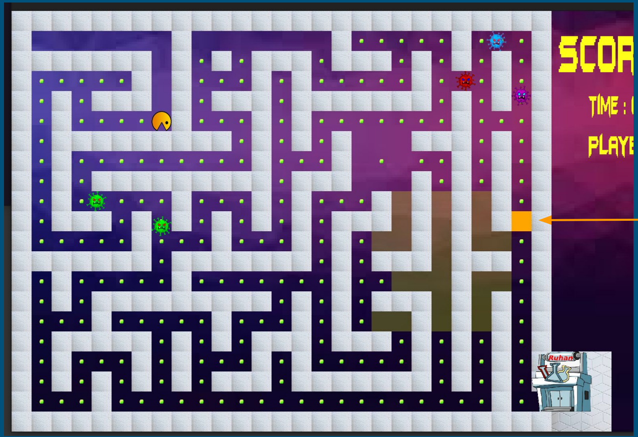


Construction of Maze

- We have constructed the maze using the randomised Depth First Search algorithm, which starts with an unexplored matrix and starting from a random cell, the algorithm starts exploring the matrix. At every iteration of the algorithm, we consider the unvisited neighbours of the last visited cell and randomly select one of them to visit next.
- At the end, the unvisited cells will be the walls of the maze while the visited ones will be the empty cells.
- To make sure that the maze has alternate walls and empty cells, adjacent cells are not taken as neighbours, but the cells at a distance of two units along all the four directions.
- But the maze generated here is a tree, which means there are dead-ends(or leaves) at which our character cannot move forward. Due to this, there is only one path from one point to another in the maze which makes it difficult for the player to avoid the enemies.
- We have removed some or most of these dead-ends depending upon the current level, so the more difficult level will have larger number of dead-ends.

Dynamic Maze

- We have stored the neighbours of the dead-ends in a vector from which we choose random cells, and if it is a wall, it will be converted to an empty cell and vice-versa, while making sure that the maze is not disconnected at any moment.



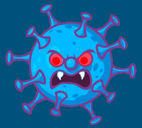
A wall transitioning to an empty cell (vice-versa)
Is also possible)

- This effect is also accompanied this with an animation so that the changes are visible to the player.
- This dynamic nature of the maze preserves the randomness of the map and reduces predictiveness, thus making the game more exciting and challenging.

Types of Enemies

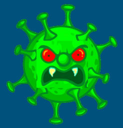
SINGLE PLAYER MODE:

The different types of enemies in single-player mode can be identified from their colour codes:

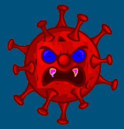


Freezer Virus : This type of virus freezes the pacman for 3 seconds on collision and deducts 15 health from the pacman

Knockback Virus : This type of virus knocks the pacman in opposite direction for 3 seconds on collision and deducts 20 health. During the knockback, the player cannot control the pacman, nor shoot any fireballs



Queen Virus : This is a special type of virus, which continuously divides to form new viruses of random types after every 15 seconds. On collision with pacman, the pacman loses 10 health points.



Explosive Virus : This is the most dangerous type of virus which explodes either while colliding with pacman, or after 15 seconds, whichever comes first. The explosion destroys any walls in a 3 x 3 area around the virus. If the pacman happens to be in the vicinity during the explosion, it loses 25 health points.

MULTIPLAYER MODE:

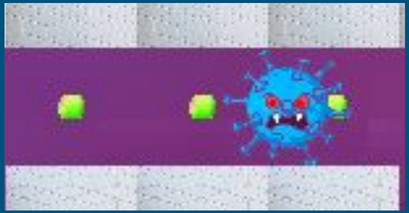
The multiplayer mode consists of only one type of enemy which when colliding with any of the players, deducts 20 health from that player:

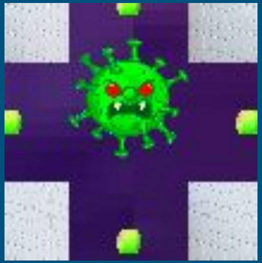
To give it a unique look, we have animated the enemy so that it periodically changes its color from one of the above eight colors.

Enemy AI

- Regardless of the type of enemy, their movement around the maze is controlled by a common algorithm.
- Each enemy will track the position of walls in their neighborhood, decide their next direction of motion based in this information.

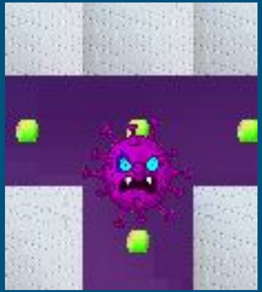
If there are only two empty spaces out of the four neighbouring cells around the enemy, then the enemy will have only two possible directions of motion, in which case, the enemy will continue to move in the same direction.





A four-way junction

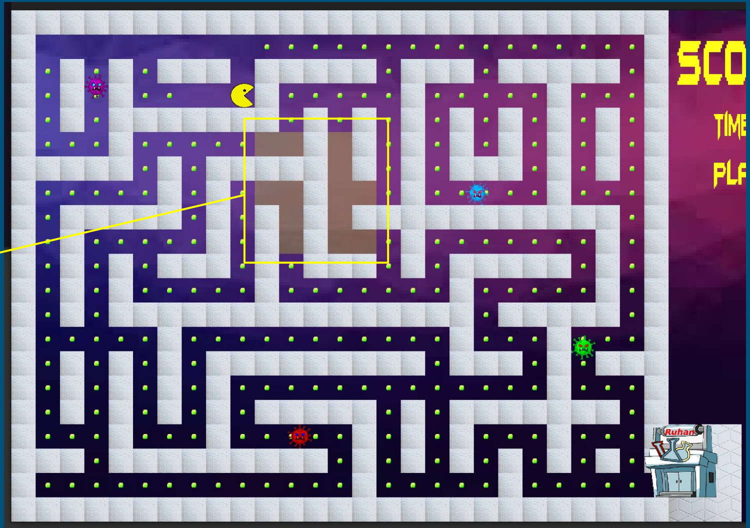
However, if there are more than two empty cells, (three in a T junction or 4 in a four-way junction, then the enemy will have more than one possible way to go. In this case, it will randomly choose its next direction to move.



A T-junction

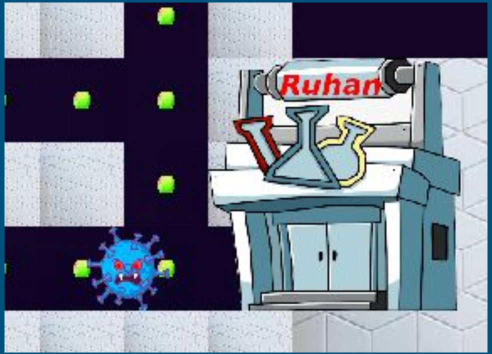
The Quarantine/Containment Zone

- The maze consists of an infected zone coloured in brown red which we call “the Quarantine Zone”, because it is a hazardous cell for pacman to stand on.
- Without a mask, the player loses health(i.e. points) while standing on it.
- The more the pacman stays in the containment zone, the more will be the rate of losing health, thus giving the player the initiative to leave it as soon as possible.



Containment Zone

The R(W)uhan* Lab



As if the number of enemies were still not enough, we have added another feature to the maze known as the Ruhan Lab in single player mode.

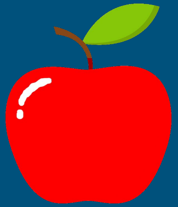
This lab, which is present at the bottom right of the maze, periodically spawns new viruses of random type onto the map. The rate at which the lab spawns new viruses increases in subsequent levels.

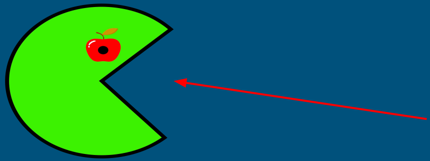
*Until the lab-leak theory is proved

Power-ups

We added so many enemies that it would be unfair if we didn't add any power-ups to help pacman from tackling them. The pacman changes its appearance while the effect of a power-up is active, so that the player can be aware of its duration.

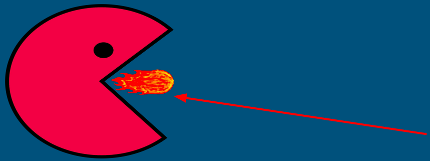
1. **Apples:** We all know that 'an apple a day keeps the doctor away'. In the same way, collecting an apple in the maze grants temporary invulnerability to the pacman, as it doesn't take any damage even while passing through enemies.





Invulnerable pacman

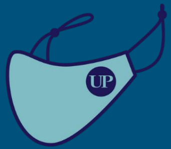
2. **Vaccine:** The vaccine is the ultimate weapon against any kind of enemy, as it grants the pacman the ability to destroy enemies if it passes through them.



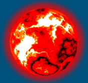
Vaccinated pacman

Power-ups(Continued)

1. **Mask:** Wearing a mask makes the pacman unaffected by the containment zone. This power-up is permanent, in the sense that once collected, the pacman can roam freely in the quarantine zone throughout the duration of the game.



2. **Fireball:** This is not exactly a power-up but a property of the pacman. The pacman can shoot fireballs to destroy any enemy in the front. The fireballs are shot in a straight line at a very high speed in the direction that the pacman is currently facing. However, the pacman has a limited number of fireballs.



Controls

- The player can control the pacman using the arrow keys on the keyboard.
- The fireball can be fired by pressing the ‘Spacebar’.
- The pause menu can be opened using the ‘P’ key.
- The game can be exited from any point using the ‘Esc’ key or pressing the exit button on the top right of the window.
- All the other elements and features of the game can be controlled by mouse.

Smooth Character Movement

- All the moving characters in the game, including the pacman, are of the exact same size as the width of the maze.
- The enemies are controlled by the program, so the enemies have perfect movement across all parts of the maze, even at T junctions and four way junctions.
- But for pacman, the human reflexes aren't precise enough to take a sharp turn at T junctions in the maze, and consequently, it is very difficult for the player to make the pacman turn at the junctions.
- To account for this human error, we have implemented a key buffer, which stores the last key pressed by the player after the most recent turn.
- This key buffer makes sure that even if the player presses a key for a turn prematurely, the pac-man will still make a turn in the desired direction. This provides the player a margin of error and thus ensures smooth movement.
- We have also animated the pacman using eight different textures, two for each direction, so that the pacman feels alive and so that it actually faces in the direction it is moving.

THANK YOU

Cheat Codes

How is a game, a game, if it doesn’t have any cheat codes? We have made sure to include some (Only in the single player mode though) :)

1) ‘E’ to spawn enemies from the Ruhan Lab
2) ‘A’ to spawn an apple in the maze
3) ‘V’ to spawn a vaccine in the maze

We didn’t want this page to be noticed, so we included it after the “Thank You”. Only the curious ones will notice.