Username:	khcy2igt
Full Name:	Irene Gohtami
Student ID:	012041

1. Overview

- This program is a form of an arcade tile-based game of the popular Pac-Man game that has been around for quite a while.
- There will be 5 sprites, one is the player controlled sprite (the Pacman) and four CPU ghost sprites.
- The aim of this game is to eat all fruits/pellets to win and avoid the ghosts in order to not lose lives.
- There are 2 level-stages that player can play with which consists of different maze.
- There are features like booster to increase the speed of the pacman and slimy jelly to slow down its speed.
- Player will be able to see lives and score boards.
- Player will also be able to pause and unpause the program while it is running.

2. Main Screenshot(s)

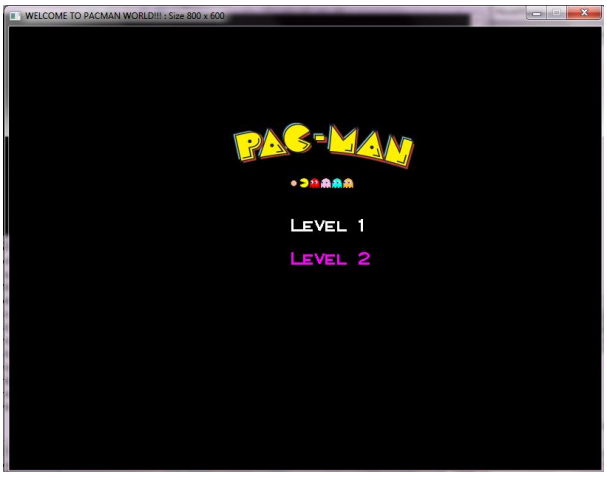


Fig 1. Main menu screen

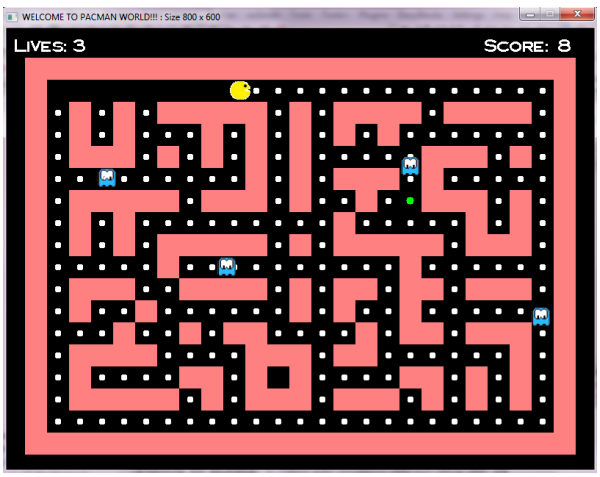


Fig 2. Level 1 Map

3. Usage

- Run the program and you will see the main menu screen. *(fig 1)*
- Player get to choose which level he/she would like to play by clicking on level 1 or level 2 texts. Selected level will be highlighted for clarity. *(fig 1)*
- After clicking on a level, the chosen level map will be shown and player can start moving the pacman sprite by pressing the arrow keys to control the directions. *(fig 2)*
- Each white pellet that the pacman passed by will increase the score by 1 and each time the pacman collides with a ghost sprite, the lives will decrease by 1 and will return the pacman back to starting position.
- Press space bar to pause or unpause the game and "Escape" key to exit the game.
- When all fruits are eaten (score=225), a 'stage cleared' screen will be displayed to show player that he has win. Change the following code fragment in "Pacman.cpp" class to get to the win state faster:

```
if(score==225)  //win state - change the number if user wants
                //to go to win state earlier/faster
        wins = true;
```

- When lives are 0, a 'game over' screen will be displayed to show player that he has lost.
- Try to eat the green and purple pellets to see any changes occur to the pacman.

4. Known problems

- If the pacman collides with a ghost at the initial starting point (1,1), it will immediately go to the losing state even though we still have lives.

5. Files

Files which I added/are mine:

File name(s)	Purpose
Pacman.cpp	Main program for pacman's movement and behaviour
Pacman.h	Pacman class header file
Ghost.cpp	Main program for ghost's movement and behaviour
Ghost.h	Ghost class header file
GameMain.cpp	Main program to integrate both pacman and ghost classes
GameMain.h	Main program header file
PacmanTileManager.cpp	Main program tile manager sub class
PacmanTileManager.h	Tile manager sub class header file

Base class files which were modified, and why:

File name(s)	Changes and reasons
BaseEngine.cpp	Only a small fragment of code is changed which is in the GetColour method 'case 2' to change the tile colour.

6. Specific requirements

❖ Method fragments (red text highlighted in black) are included to indicate in which part of the code the functionality is implemented.

- **Draw an appropriate background**
  - There are several screens and backgrounds implemented in this program, which consists of the main menu, level 1 map, level 2 map, paused state, game over state and win state which will change according to player's interaction.

```
void GameMain::SetupBackgroundBuffer()
```

- **Have moving objects**
  - There are 2 types of moving objects which is the player controlled pacman and the 4 ghosts. Both type of objects moved inside the maze in the game's main state and when the pacman collides with any of the ghosts, it moves back to its starting position.

```
void Pacman::DoUpdate( int iCurrentTime )
void Ghost::DoUpdate( int iCurrentTime )
```

- **Have interactions between the moving objects and the environment**
  - The white pellets will disappear when the player passed them over and when a green pellet is consumed, the player gain more speed in its movement or when the purple pellet is consumed, the player's movement is slowed down.

```
void Pacman::DoUpdate( int iCurrentTime )

    if ( m_oMover.HasMovementFinished(iCurrentTime) )
    { ... }
```

- **Provide player interaction**
  - The player can move the pacman around by using the arrow keys.

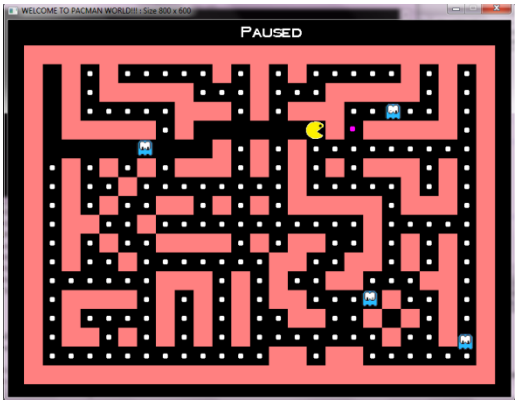
- **Provide AI-controlled objects**
  - There is no AI implemented for this program.

- **Save and load information to and from files**
  - Loading information: player could load a different layout map and depends on which level, the speed of the ghost differs as well. For example, level 2 will have the ghosts moving in a faster way to increase the difficulty[1]. Images for the moving objects are also loaded and redrawn based on the current movement's direction[2]. There are also some images loaded in the stateInit, gameOver and win states.

```
1.  void Ghost::DoUpdate( int iCurrentTime )
    if ( m_oMover.HasMovementFinished(iCurrentTime) )
    { ... }

2.  void Pacman::Draw()
    void Ghost::Draw()
```

– Saving information: player can pause the game by pressing space bar which will save the current game state and continue from that by pressing space again.

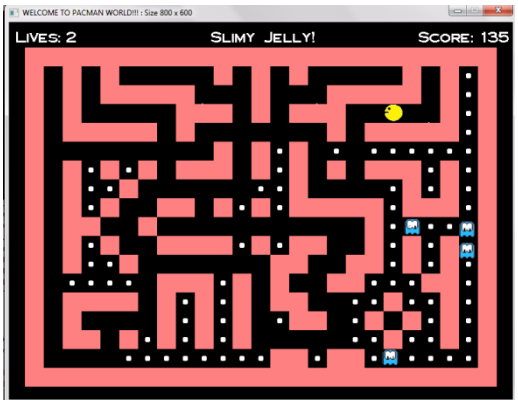


void GameMain::KeyDown(int iKeyCode)

Fig 3. Paused State

- **Display status information on the screen**

– On top of the screen are where all information is displayed, which consists of current score, current live, when a booster or slimy jelly is acquired and when the state is paused or unpaused.



void Pacman::printScreen()

Fig 4. Slimy Jelly

- **Support different states**

– There are a total of 9 states involved: stateInit (the main menu state), level1 & level2 states (load the corresponding level map), temp1 & temp2 states (for animation purposes when the mouse cursor is over a certain text in the stateInit), gameOver (when player loses), win (when player wins), stateMain (the game's main state) and lastly statePaused (when the game is paused).

Fig 5. State Game Over

Fig 6. State Win

7. Marking criteria

Complexity (VERY IMPORTANT)

- Animations of moving pictures for the pacman
- Displaying the scores and updating it to the screen constantly.

Compilation and Reliability

- Sometimes the compilation is slow because of loading several images although, this depends on the performance of the PC used.

Clarity of code, or this documentation

- None

Efficiency

- The redrawing of the pictures are not completely perfect, as sometimes you can still see white specks left by the pacman. On the other hand, you can see smooth movement of the pacman and ghost as well as the pacman's mouth opening and closing up.
- The balance of the speed between the ghost and pacman is pretty efficient because it is set to a realistic value so that a win is still achievable even when the difficulty of the game is increased.

Appearance

- Images are added in the main menu screen, game over screen and stage cleared screen to make the layout/appearance more attractive.
- Pacman images are used as the main sprite and ghost image are used to represent the enemy.


8. Additional information

- Features such as booster and slimy jelly which increases and decreases the pacman's speed are added as extra functionalities of the game. Both of them have a time limit of usage.

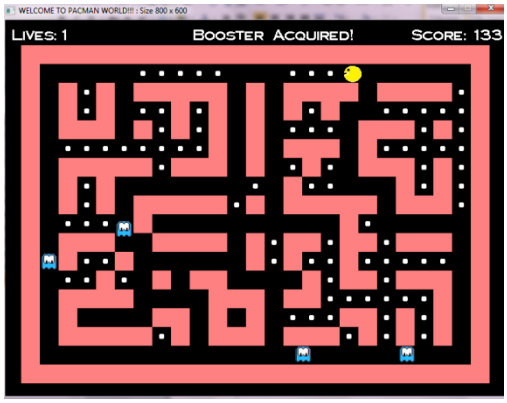


Fig 7. Booster