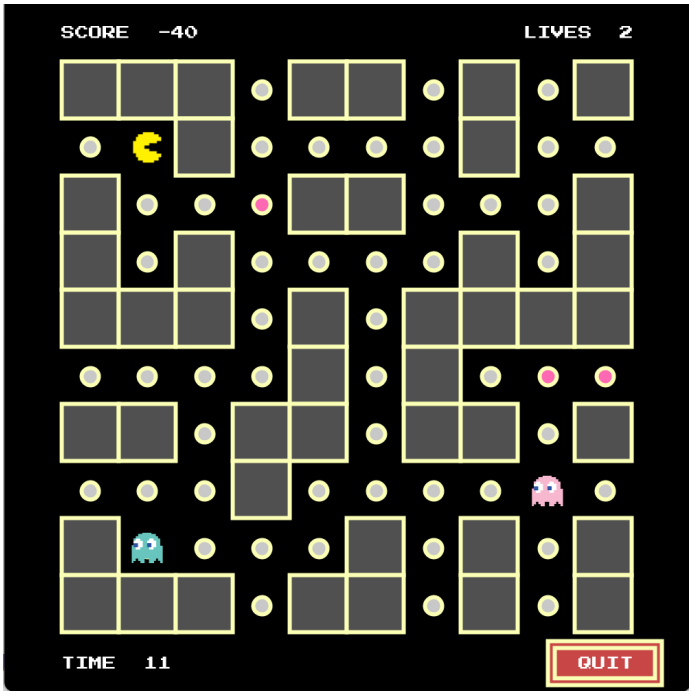
Final Culminating PAC-MAN Project:

The player controls Pac-Man, who **must eat all the dots inside an enclosed maze while avoiding two coloured ghosts**. Eating large flashing dots called Cherries" causes the ghosts to temporarily turn blue, allowing Pac-Man to eat them for bonus points.

The player must collect all the white dots and avoid all the ghosts. Each player has three lives and a life is subtracted from the player every time the ghost catches up to the player.

Eating a circle will add 50 points while a ghost will add 150 points to the scorebank. The user has the ability to quit and change their name.



Global Variable Name	Type	Description
gameGrid	2-D array/list	The 10 by 10 grid for the board of the game. 0 = a path that can be taken, 1 = wall Each object on the screen has a coordinate in the 10 by 10 board.
PAC	Class	Initialisation provides x and y coordinates for PACMAN, its initial direction (always facing right) and its speed (1 unit)
dirInput	Int	Keyboard feedback that is stored in a number from 0 to 3, for the 4 directions possible for the player. 0 = right 1 = left 2 = up 3 = down
timer	int	Timer is initialised to millis(), to regulate the movement of the pieces on the board. The pieces on the board can only move once in 0.55 seconds.
GHOST1	Class	GHOST class that is initialised with its location on the board. When drawPath is called, the GHOST will receive the gameGrid, the player's current location, and whether to choose the most efficient or second most efficient path. To find the location of PAC on the board, the ghost will move itself recursively through all the valid (not

		walls) of the game board. It will store the list that has the shortest number of moves to reach the player.
GHOST2	Class	The second ghost that is initiated by the identical class as GHOST1
changePath	int	If the two ghosts' coordinates overlap on each other, moving on the same grid, then the program will attempt to change one ghost's path.
dots	list/array	For every 0 value on the gameGrid, dots creates a 1 value that signifies a white dot that the player can eat to gain points. When the player passes over a dot, the 1 value turns into 0, to signify nothing on the coordinate.
escapeMode	boolean	If the player eats a cherry, the ghost will enter escape mode and run to a designated corner.
G1	list/array	Designated coordinates for ghost to run to when escapeMode is activated. When the cherry is eaten by the player.
G2	list/array	Designated coordinates for ghost to run to when escapeMode is activated. When the cherry is eaten by the player.
scattertime	int	When escapeMode is activated, scattertime starts to count. When millis() - scattertime > 3500, ghosts will start to blink, signifying the end of scatter mode. When millis() - scattertime > 7000, ghosts will return back

		to chasing the player.
flashtime	int	Every other cycle of flashtime, ghost will display the blue colour in scatter mode. Else, it would not show. This variable is used to display the ghost blinking when scatter mode is almost finished.
blueOff	boolean	If blueOff is False, ghost is displayed on screen. If blueOff is True, ghost will not be displayed. Used to display blinking ghost.
died1	boolean	If the GHOST1 has already been eaten once in scatter mode, this variable will turn True, meaning the player cannot eat this ghost again.
died2	boolean	Same as above, but for GHOST2
player	class	Stores the name of the player, its score, its lives. Used to determine whether the game is over. At the end of the game, the player's name and score is appended and saved to scoreL
pausetime	int	When the game is paused through pressing quit, the time displayed at the bottom of the display would still run. Therefore, the amount of time that the player has been pausing for must be stored so that when the player starts

		the game again, the time displayed will subtract out the pausetime. Pausetime would accumulate if player pauses more than one time.
paused	boolean	If game is paused, nothing on the screen would run
mousePress	boolean	If the mouse was pressed, then was released, mousePress would be true. Used for buttons.
time	int	When the game is paused, then unpaused, *pausetime* is added to the preexisting amount in the variable *time*. Ensure the clock at the bottom of the gameplay is accurate.
mode	int	Displays the various game modes/screens in the game such as leaderboard and customise screen.
scrollinc	float	When the player drags its mouse down or up in leaderboard, the leaderboard will move up and down. scrollinc is a fraction of the amount of (distance the mouse moved)/ (total display height) Dynamic variable that will change depending on how long the scoreboard is
pinky, bluey, scattery, pac	Pimage	Images that are loaded into the game
mouth	int	Timer that helps to cycle through the four eating animations for PACMAN's mouth

mcycle	int	Cycles through from 0 to 3 for everytime millis() - mouth > 200. Every value of mcycle is a different image in the pacman animations
boxoutline	List of boolean values	Three different colour customisations When the mouse selects one of the three, all the other two turn to False. Allows the program to change the outline thickness in the customisation screen and to register the change in colour theme.
Uinput	string	When key is released, Uinput concatenates the new input into Uinput. Used when user needs to type something Accepts keys from a wordbank
scoreL	List of dictionary values	Stores the score and username of each of the players after they have played the game
invalidc, invalidn	boolean	After player attempts to submit their customisation choices without making one, these turn True. If invalidc == True, player hasn't picked colour If invalidn == True, player hasn't picked username, or the username has already been taken upon checking scoreL

win	boolean	After the game ends, win is used to display a certain message to the player. If win is True, it will show “you won!” If win is False, it will show “game over”
saveEntered	boolean	If the player has tried to enter a file directory saveEntered turns True.
saveName	string	Takes the keyboard input from the player, and registered for file directory input.
wrongfile	boolean	If the file directory that was inputted was incorrect, the system will prompt the player to input the full path for the file.
themecolour	List storing int values (len == 3)	Retrieves the user selected theme from the variable *theme.*
theme	List storing int values (len == 1)	The three theme colours have rgb values that are stored in this list. (black, green, and blue)
tick	string	The flashing underscore that is shown at the end of a user input.
scrollbarlen	int	Length of the scrollbar in the leaderboard
upcycle	boolean	Used to control the animations of PACMAN. If mcycle has reached 3, the upcycle becomes False so that PACMAN’s animation can go backwards. When mcycle is 0, upcycle becomes True again.
slide	int	Countdown to start the game

		slide count is decreased by 1 everytime a second has passed. Gives program ability to count down from 3.
minput	int	In multiplayer mode, minput is the control for the ghost. (multiplayer input only)
multiplayer	boolean	If multiplayer is true, the game will go into multiplayer mode.

TASK	MARKS	COMMENTS
Using keys Get the name of each player Two player games with each player having their own controls Program should be both mouse and key driven Demonstrate 2 different mouse interactions Must demonstrate effective use of mouse routine. The best way to do this is by implementing a menu structure and/or using clicking on screen to control the program	5	User input their nickname and save onto a local file. Uses WASD to control player movement Mouse is used to click start and quit Scrolling/dragging mouse is used for the leaderboard
Use files to store information about the setup of the game and the images required. Have different versions of the game available(i.e. change dimensions, change images). Have the same code work for these variants	5	Can change player colours and ghost colours. Stores setup and scores on a local file.
Scoring Keep track of each player and their score during the running of		Score will be sorted and displayed at the end of each game.

the program. Sort those scores using bubble sort/selection sort/insertions sort when you first run the game. Learn how to use the utility pickle – check wikipedia for an explanation – to save them for the next time Allow the merging of new scores into your sorted list the next time you use the game probably with a modified insertion sort Print these scores on screen when a player chooses scores from menu showing 5 scores at a time and allow them to page down Players only keep their best score.	5	Score can be displayed from selection in the main menu. Pickle to save file.
Use the correct data structures effectively to facilitate the efficiency of the game. Store the information for a player name and their scores in a list. When you enter the name of a player search to see if they have played before and report their score.	5	Uses 2-D Array to store user data and will have a search bar on the top of the score list to search for a specific player score.
Have Help, Replay, Scores, Start, Return options on a menu	4	Has a help page that explains the game. Has an ability for the player to replay without exiting the program. Has a start button. Displays score in leaderboard and inside the game.
Allow a user to replay a game or a new player to join without needing to close the program and restart	3	YES
Be creative!	5	OK