**Pacman Game ~ Interactive Maze Chase Game (Design Document)**

**Created By: Ankeet & Prabhjot**

## 1. Introduction

* The Pacman game is a classic arcade-style game in which the player controls a character (Pacman) navigating through a maze.
* The objective is to collect all the coins in the maze while avoiding ghosts that chase the player.
* The player earns points by collecting coins, and the game wins when all coins are collected. If the player is caught by a ghost or hits the wall, the game ends.

## 2. Objective

* The goal of the game is to navigate through the maze, collect all the coins, and avoid being caught by the ghosts to achieve victory.

## 3. Gameplay Overview

* **Starting the Game:** The game begins with Pacman placed in the maze, with coins scattered throughout and ghosts moving around the maze.
* **Controls:** The player controls Pacman using the arrow keys to move in the maze (up, down, left, right).
* **Scoring:** Players earn 1 point for each coin collected.
* **Winning:** The game ends in victory when all coins are collected.
* **Losing:** If a ghost catches Pacman or Pacman hits the wall, the game ends, and the player loses.
* We have added a page with clear instruction and start game option.
* **Features:**
* Ghosts move unpredictably, making it challenging for the player.
* A reset feature allows players to restart the game after losing.
* Sound on collecting the coins.
* Press Enter to play again.
* (Optional Future Features): Power-ups to temporarily make ghosts vulnerable or increase Pacman's speed.

## 4. Form-Design



A screen shot of a game

Description automatically generated

## 5. PSEUDO Code

// SoundManager Class

START:

Initialize SoundManager:

Create `coinsound` as WindowsMediaPlayer object for coin sound

Create `playsound` as WindowsMediaPlayer object for game sound

Method `GetResourcePath`:

Input: fileName (name of the sound file)

Output: full path to the sound file in the "Resources" folder

Logic:

Combine the current directory with "Resources" and the file name to get the complete path

Method `PlayCoinSound`:

Logic:

Set `coinsound.URL` to the path of "coin.mp3" using `GetResourcePath`

Play the `coinsound`

Method `PlaySound`:

Logic:

Set `playsound.URL` to the path of "play.mp3" using `GetResourcePath`

Play the `playsound`

// Main Game Form (PacManGame)

START:

Initialize game variables:

`score` = 0

`playerSpeed` = 8

`redGhostSpeed` = 5

`yellowGhostSpeed` = 5

`pinkGhostX` = 5

`pinkGhostY` = 5

`isGameOver` = false

Initialize `SoundManager` object: `soundManager`

Method `keyisdown` (on key press):

Logic:

If "Up" key is pressed: Set `goup` = true

If "Down" key is pressed: Set `godown` = true

If "Left" key is pressed: Set `goleft` = true

If "Right" key is pressed: Set `goright` = true

Method `keyisup` (on key release):

Logic:

If "Up" key is released: Set `goup` = false

If "Down" key is released: Set `godown` = false

If "Left" key is released: Set `goleft` = false

If "Right" key is released: Set `goright` = false

If "Enter" key is pressed and `isGameOver` is true:

Call `resetGame`

Method `mainGameTimer` (Main game loop):

Logic:

Play main game sound using `soundManager.PlaySound()`

Hide win/loss indicators

Update score display on screen

Move PacMan based on direction:

If `goleft` is true: Move PacMan left

If `goright` is true: Move PacMan right

If `godown` is true: Move PacMan down

If `goup` is true: Move PacMan up

Implement screen wrapping (PacMan reappears on the opposite side of the screen)

Loop through all game elements:

If element is a coin:

If PacMan collects the coin:

Increase score

Play coin sound using `soundManager.PlayCoinSound()`

Hide the collected coin

If element is a wall:

If PacMan collides with wall:

Trigger `gameOver` method with "You Lose!" message

Show loss screen

If element is a ghost:

If PacMan collides with ghost:

Trigger `gameOver` method with "You Lose!" message

Show loss screen

Move ghosts and reverse their direction if they hit boundaries:

Red ghost moves based on `redGhostSpeed`

Yellow ghost moves based on `yellowGhostSpeed`

Pink ghost moves based on `pinkGhostX` and `pinkGhostY`

Check if all coins are collected (win condition):

If `score == 46`:

Trigger `gameOver` method with "You Win!" message

Show win screen

Method `resetGame`:

Logic:

Reset score to 0

Reset game variables (`redGhostSpeed`, `yellowGhostSpeed`, `pinkGhostX`, `pinkGhostY`, `playerSpeed`)

Set PacMan's position to the starting coordinates

Set ghost positions to starting coordinates

Make all game elements visible

Start game timer

Method `gameOver` (End the game):

Logic:

Set `isGameOver` = true

Stop game timer

Display final score and message on screen

Show "Game Over" message if the player loses or "You Win!" message if the player wins

// Main Form (Starting Form)

START:

Initialize main form components (e.g., buttons, labels)

Method `button1\_Click` (on clicking the "Start" button):

Logic:

Create a new instance of `PacManGame` form (game window)

Show the `PacManGame` form

Hide the current main form (the starting screen)

## 6. Class Diagram

