

**Smart Computer Graphics**

**Fall 2024**

**Project Title: 2D Maze**

**Instructor Name: Rafat Damseh**

**Group members** (Section 51)

|  |
| --- |
| Salama Mohammed AlKaabi 202211492  Khawla Khaled Alneyadi 202212912 |

# Introduction

Our project is about creating an interactive maze game that combines fun and engaging gameplay with key concepts from smart computer graphics. Our goal is to design a visually appealing game where players navigate automatically generated mazes, collect power-ups, and race against the clock. To make the experience more personal, our game allows users to customize features like player colors, adding a creative touch.

Our project involves generating solvable mazes using algorithms, introducing time-based challenges, and designing a user-friendly interface with advanced GUI elements. As players progress, the mazes become more complex, with stricter time limits and intricate layouts. It also includes features like power-up collection, level progression, and retry options, ensuring that it stays challenging and engaging.

This work ties closely to smart computer graphics by incorporating dynamic visuals, real-time interactions, and procedural maze generation. By using tools like tkinter, our project shows how graphical interfaces can be practical and interactive. The focus on customization and procedural design reflects modern trends, showcasing how smart computer graphics can be used to create immersive, interactive experiences.

# Methodology

**Methods and Technologies Used:**

1. **Programming Language:**

The game was created in Python due to its simplicity, versatility, and the availability of the built-in Tkinter toolkit for GUI development.

1. **Technologies:**

The game relies on Tkinter to create a simple and user-friendly interface. Tkinter makes it easy to display the maze on a canvas and add buttons, labels, and even a home screen where players can choose their character’s color. To keep the gameplay fresh and exciting, I used the random module to generate unique maze paths, add extra walls, and place yellow power-ups in random spots. Lastly, the time module helps keep the game challenging by managing a countdown timer for each level, pushing players to finish the maze before time runs out.

1. **Smart Graphics Techniques:**

The game uses a Depth-First Search (DFS) algorithm to generate a unique and solvable maze for every level, making sure each play through feels new and exciting. To make things more challenging, extra walls are added randomly as you progress through the levels. To ensure the maze is still solvable after these changes, the game uses a Breadth-First Search (BFS) check in the background. Everything in the game is dynamically updated, the maze, the player’s position, the power-ups, and the goal, so it all changes in real time as you play, keeping the experience smooth and interactive.

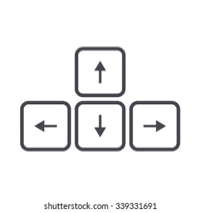
1. **Player Customization:**

Players can make their characters unique by choosing a favorite color (blue, green, purple, orange, or brown) right from the home screen. The chosen color stays with the character, showing up on the screen and throughout the game. This simple yet thoughtful feature adds a personal touch, making the game more fun and meaningful for everyone.

**How The Game Works:**

1. **Home Screen (Figure 1):**
   * The game starts with a friendly and easy-to-navigate home screen where players can:
     + Pick their favorite color to personalize their character.
     + Read simple, clear instructions on how to play the game.
     + Jump straight into the action with a big, inviting "Start Game" button.
2. **Gameplay Features (Figure 2):**
   * **Maze Adventure:**
     + Players dive into an exciting maze adventure, navigating through a grid filled with gray walls, white paths, yellow power-ups, and a bold red goal that’s easy to spot. Getting around is effortless so you must use the arrow keys to move your character and explore the maze.
   * **Challenging Levels and Timer:**
     + There are 5 progressively harder levels, each with less time to complete:

Arrow Keys

* + - Level 1: 50 seconds
    - Level 2: 40 seconds
    - Level 3: 30 seconds
    - Level 4: 20 seconds
    - Level 5: 15 seconds
    - If time runs out before finishing a level, the game resets to Level 1, keeping the challenge alive.
  + **Exciting Power-Ups:**
    - Scattered throughout the maze are yellow power-ups that you must grab before reaching the goal.
    - If you miss the power-up, you’ll just get another shot at the same level to try again.

1. **Dynamic Updates:**
   * Everything in the game updates in real time, from your character's movement to the power-ups and timer, keeping the experience smooth and immersive.
   * Your chosen character color stands out as a vibrant circle that moves through the maze, making it easy to track your progress.
2. **Retry and Reset Options:**
   * **Retry Level:**
     + If you reach the goal without collecting a power-up, the game lets you retry the same level with the same maze setup.
   * **Reset Game:**
     + A handy "Home" button on the screen takes you back to the home screen anytime, letting you start fresh from Level 1.
3. **Victory Celebration:**
   * Beat all 5 levels, and you’ll be greeted with a congratulatory message. It even shows how many power-ups you collected.

**Rationale for Design Choices**

1. **Python with Tkinter:**
   * Tkinter makes building graphical interfaces straightforward, perfect for quickly designing and customizing the game’s look and feel.
   * Python’s clean and easy-to-understand syntax made it simple to handle game logic, create visuals, and manage user interactions efficiently.
2. **Dynamic Maze Generation:**
   * By using a depth-first search (DFS) algorithm, every maze generated is not only unique but also guaranteed to be solvable, keeping the gameplay fresh and exciting.
   * Randomly placed walls and power-ups make each maze more strategic, encouraging players to fully explore and think ahead.
3. **Player Customization:**
   * Giving players the option to choose their character’s color adds a personal touch, making them feel more connected to the game and their character.
4. **Progressive Difficulty:**
   * Shorter time limits and more complex mazes as players advance create the perfect mix of challenge and fun, motivating players to improve with each level.
5. **Replayability:**
   * Features like retrying a level or restarting the game without harsh penalties make the experience forgiving and enjoyable, inspiring players to keep coming back for more.

**Connection to Smart Computer Graphics**

1. **Procedural Generation**:

* Dynamically generating mazes ensures variety and unpredictability.

1. **User Interaction**:

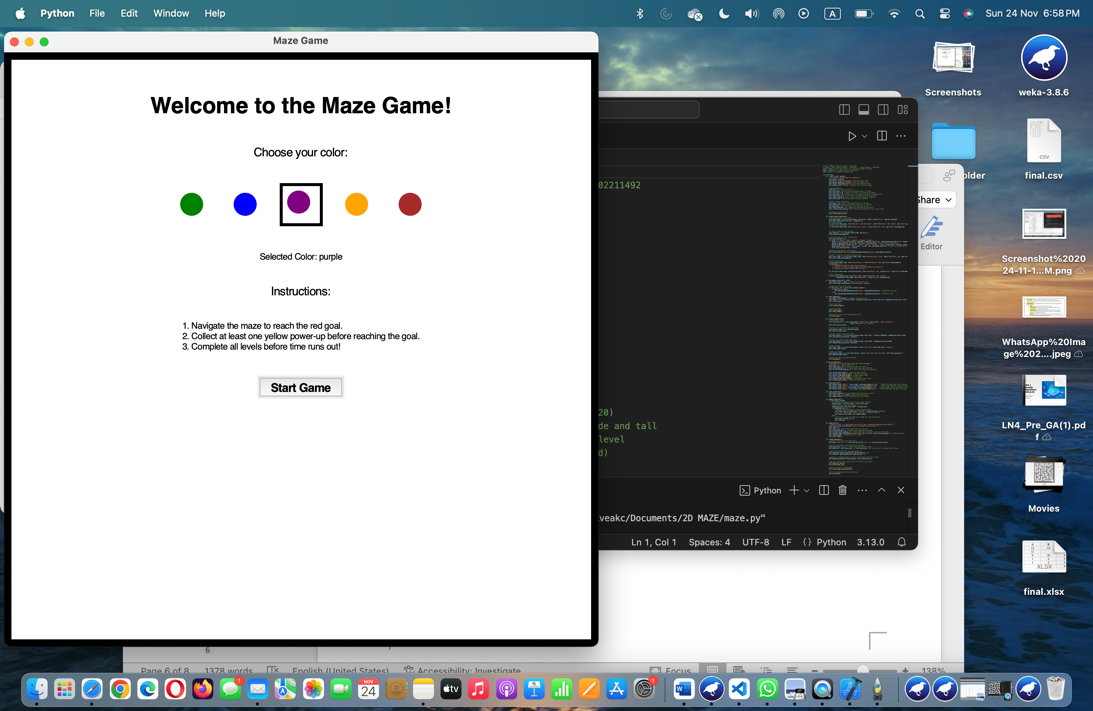
* Real-time updates to the maze, player position, and game state create a responsive and immersive experience.

1. **Customization**:

* The ability to select player colors reflects personalization, an important aspect of interactive graphics.

**Figure 1: Illustration of Home Screen: Figure 2: Illustration of 2D Maze:**

A screenshot of a game

Description automatically generated

player

Power-Up

Next (Button)

Home Page (Button)

Goal

The player selects the purple color.

# Results

In developing our maze generation game, we encountered several technical challenges that required problem-solving and persistence. Initially, we had to overcome various logic and syntax errors that disrupted the game’s functionality. These issues caused frustrating setbacks, but through careful debugging and testing, we were able to resolve them and restore the game's operation.

Another significant challenge was ensuring the maze was always solvable. The maze generation algorithm sometimes resulted in configurations where the goal was completely unreachable due to a wall being placed in the wrong spot. This required adjusting the algorithm to guarantee a viable path from the start to the finish.

We also faced difficulties with implementing the feature to change the player's color based on their choice. This required meticulous attention to detail to ensure the player’s color updated correctly without disrupting other game functions.

Despite these obstacles, we were able to refine the game design, ultimately ensuring the maze is always solvable, the player's choice of color is respected, and the overall user experience remains smooth and enjoyable.

# Conclusion

The 2D Maze Game effectively demonstrates the practical applications of smart computer graphics by integrating interactive design, procedural maze generation, and user-focused features. Utilizing Python’s Tkinter library alongside algorithms like Depth-First Search and Breadth-First Search, the project achieved a harmonious balance of functionality, creativity, and visual appeal. Dynamic maze updates, personalized character customization, and progressively challenging levels kept players engaged, highlighting the potential of smart graphics to enhance user experiences. Despite challenges in balancing randomness with solvability and optimizing performance, these obstacles were addressed through thorough testing and problem-solving, resulting in a polished, functional game. Overall, the 2D Maze Game serves as a valuable learning experience and a compelling example of how computer graphics can be used to create engaging, interactive, and innovative applications.