GADE 7321 POE PART 2

SYED IMRAN ST10076292 FARAAZ SHUFLA ST10122918

GAME CONCEPT DOCUMENT

Table of Contents

MAZE ESCAPE

https://github.com/FaraazSuffla/GADE7321_Part2

High Concept Statement

Maze Escape is a 3D, turn-based, strategy game which operates like a board game, in the sense that players can only move one tile at a time. The game takes place in a maze which players must navigate to reach the endpoint. Players can only move one space at a time and take turns after each move. The strategy element is present in how players choose to navigate the maze and also how and when they choose to set back the other player. Maze Escape is unique because it takes some mechanics of board games and implements them into a 3D video game. The split screen offers an interesting nostalgic feel to the game and some of the powerup options inject a competitive and strategic feeling into the game.

Game Rules

Starting Maze Escape

- Find somebody to play the game with.
- Launch the game.
- Play one round of rock, paper scissors. The winner decides which player they want to be.
- Take your positions on either side of the keyboard. Player 1 on the left (by the WASD keys) and Player 2 on the right (by the arrow keys).
- Start the game and enjoy!

What happens in Maze Escape?

- Upon entering the level players will immediately see their respective player characters on the left and right side of the screen respectively.
- Controls can be always seen on the top left of the screen.
- Whoever’s turn it is currently will always be displayed at the bottom middle section of the screen.
- Players will now have to move their player character towards the end of the maze and compete to reach the end first.

What can players do in Maze Escape?

- Players can move forward, backwards, left, or right one space (possible movements depend on the surroundings).
- Players can collect powerups.
- Collected powerups can be used at any time during the game (think strategically).
- No one is stopping you from screen watching, just be prepared for the arguments cause it’s technically cheating.
- Powerups can be used to impede the progress of the opponent and sabotage them.

How to win Maze Escape?

- The only way to win Maze Escape is to reach the end of the maze before your opponent.
- Since the game is turn-based, there won’t be any possibility for a draw, only a winner and a loser.

Unforeseen situations?

- In the event that something unforeseen happens such as a player not being able to move or the game glitching, the game should be restarted.
- Screen watching disputes can be resolved in person and players can either continue playing or restart the game.

In Maze Escape, depicting and storing the games status essential for keeping track of progress supporting game features and delivering a smooth user experience. This guide details the approach taken to represent and save the games status in the codebase.

Game State Representation

Filimowicz (2023) explains that game state is the current state of a game including all necessary information such as score, turns and other important information which defines the game. In Maze Escape the game state involves components such, as player locations, maze setup, turn based gameplay mechanics and other key aspects that shape the status of the gaming environment and influence interactions between players and surroundings.

1) Player Locations:

- The position of each player in the game is indicated by the coordinates of their GameObjects transformations.
- Player movements are continuously monitored during gameplay updating their positions as they navigate through the maze.
- The GameManager script controls player turns to ensure that one player can move at a time maintaining consistency, in player positions.

2) Maze Setup:

- The maze structure is depicted as a grid comprising cubes that can function as walls or floors.
- A two-dimensional array (mazeArray) stores the status of each cube with '1' indicating a wall and '0' indicating a floor. The MazeGenerator script creates mazes dynamically by adjusting parameters, like width, height, and the likelihood of walls.
- If you want to make changes to the maze, such as adding walls or altering the maze layout you can do so by updating the mazeArray.

3) Turn Based Gameplay:

- The GameManager script manages turn-based gameplay by controlling the order of player actions.
- It keeps track of whose turn it's Player 1 or Player 2) and switches turns accordingly.
- During a players turn only actions, from that player are executed to keep the gameplay synchronized.

Game State Utility Function

In an escape game where players take turns it's essential to assess the value of each game state to make smart choices and plan effective gameplay strategies. This guide explains how we created a utility function to evaluate the attractiveness of game states in the context of navigating mazes and advancing through levels.

Description of Utility Calculation:

Tommy Thompson describes how an AI utility function essentially evaluates and scores how good an idea or pathway is before taking action (see How Utility AI Helps NPCs Decide What To Do Next | AI 101, 2021). The utility function aims to measure how desirable or appealing a game situation is, in escape gameplay. It considers factors that're fundamental to this kind of gameplay. The utility of a game scenario is determined by assessing the following components:

1) Distance to the Exit:

- The main goal of the game is to navigate through the maze and reach the exit point.
- The utility function calculates the route from the players location to the exit.
- Shorter distances to the exit have an impact on utility indicating progress toward achieving the objective.

2) Proximity to Obstacles:

- Obstacles like walls or dead ends hinder player movement, making escaping more challenging.
- The utility function assesses how close players are to obstacles, penalizing scenarios with high obstacle density or closer proximity to areas.
- Scenarios with paths and fewer obstacles receive utility scores.

3) Turn Based Progression:

- Since the game operates on a turn-based system the utility function takes into consideration the players turn and its influence on the game state.
- Scenarios where players have made progress, towards reaching the exit or strategically positioned themselves for advancement are assigned greater utility values. Efficiency is taken into consideration when calculating the utility focusing on minimizing the number of turns needed to reach the exit.
- There are some difficulties or obstacles to watch out for in maze navigation such, as elements like enemy encounters, time limits or limited resources.
- The utility function adapts the utility score based on any threats or challenges within the game state.
- States that offer challenges or opportunities, for overcoming obstacles receive higher utility scores.

Game States Diagrams

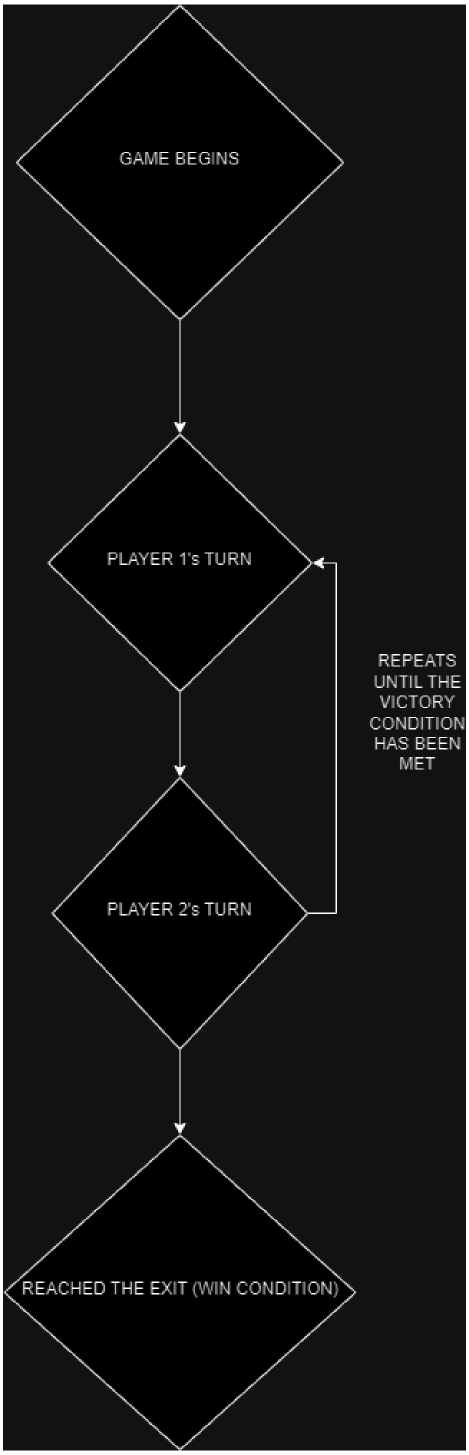
GAME BEGINS
PLAYER 1's TURN
REPEATS UNTIL THE VICTORY CONDITION HAS BEEN MET
PLAYER 2's TURN
REACHED THE EXIT (WIN CONDITION)

Figure 1: A drawing illustrating a diagram on the game states in Maze Escape, in a broad sense (Imran, 2024).

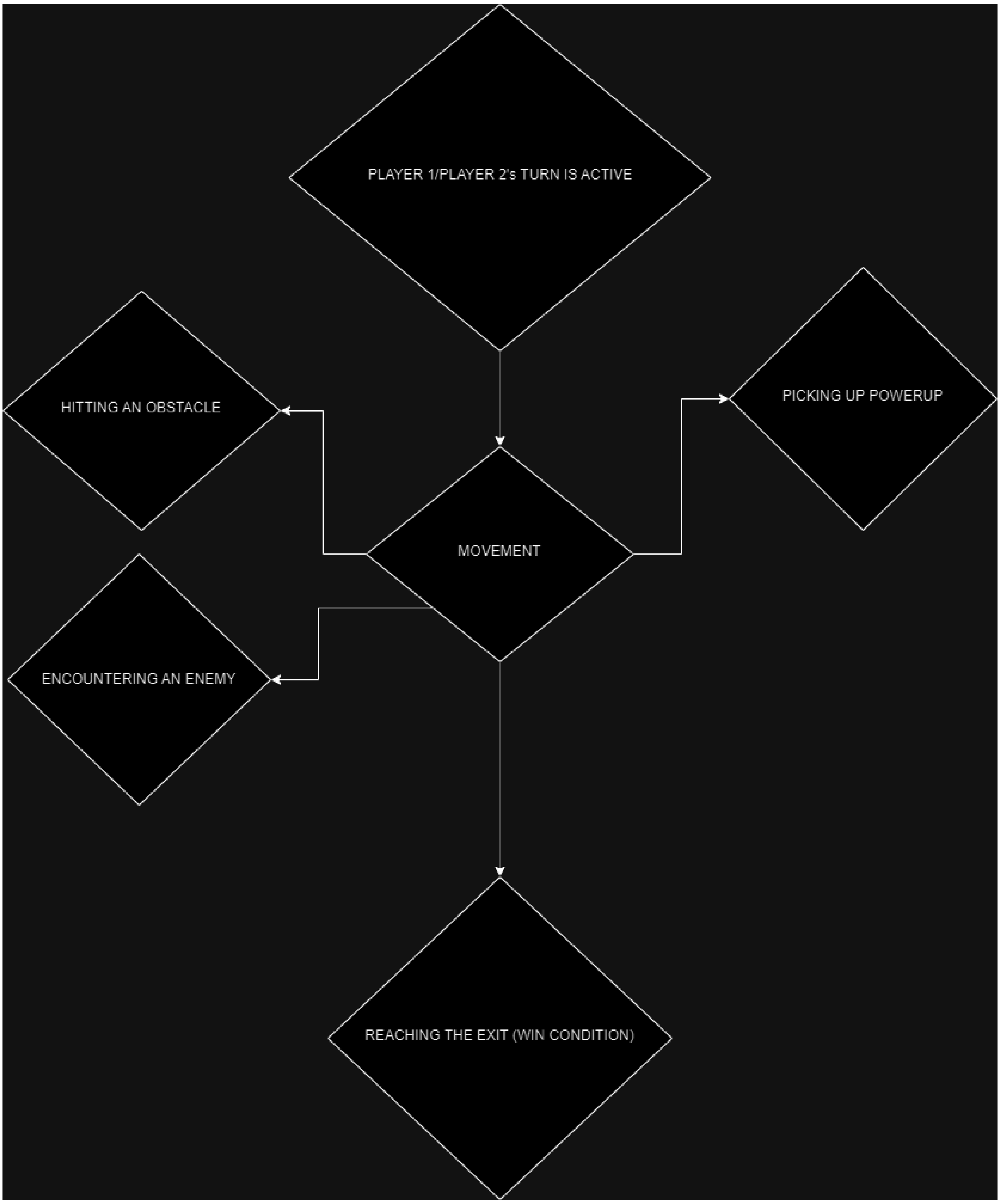
PLAYER 1/PLAYER 2's TURN IS ACTIVE
HITTING AN OBSTACLE
PICKING UP POWERUP
MOVEMENT
ENCOUNTERING AN ENEMY
REACHING THE EXIT (WIN CONDITION)

Figure 2: A drawing illustrating a diagram on the game states in Maze Escape, in more detail (Imran, 2024).

Examples of Utility Evaluation

To illustrate the utility function's evaluation process, consider the following examples:

Scenario 1: Player, to Exit

- In this scenario the player is positioned near the exit with a path and few obstacles.
- Evaluation of Utility; The utility score is high because of the proximity to the exit and favourable terrain conditions.

Scenario 2: Player Encountering Dead End

- In this situation the player encounters an end necessitating a detour to move forward.
- Evaluation of Utility; A lower utility score is given due to the density of obstacles and inefficient pathing.

Scenario 3: Planning Moves for Future Turns

- The player strategically places themselves to gain an advantage in upcoming turns.
- Evaluation of Utility; The utility score ranges from moderate to high considering the potential, for progress and strategic planning.

List of Figures

Reference List

Imran, SE. 2024. A drawing illustrating a diagram on the game states in Maze Escape, in a broad sense. [Personal drawing]. Johannesburg: Unpublished.

Imran, SE. 2024. A drawing illustrating a diagram on the game states in Maze Escape, in more detail. [Personal drawing]. Johannesburg: Unpublished.

Filimowicz, M. 2024. The Game State, Information & Movement, 3 January 2023. Medium. [Online]. Available at: https://medium.com/understanding-games/the-game-state-f57e3e512bf7#:~:text=In%20a%20video%20game%20like [Accessed 30 April 2024].

How Utility AI Helps NPCs Decide What To Do Next | AI 101. 2021. YouTube video, added by AI and Games. [Online]. Available at: https://www.youtube.com/watch?v=p3Jbp2cZg3Q [Accessed 30 April 2024].