Game Document: AR Tortoise Maze Adventure

Miggiano Davide 4840761
Morando Andrea 4604844

1 Introduction

"AR Tortoise Maze Adventure" is an augmented reality (AR) game developed using Unity and AR Foundation. The player controls a tortoise navigating through a 3D maze. The tortoise is controlled via a drag-and-drop programming interface similar to Scratch, where players can create sequences of movements and actions to guide the tortoise to the finish line.

2 Gameplay

The player uses an intuitive UI to drag and drop command blocks, creating a sequence of movements for the tortoise. The game begins with the player scanning a flat surface to place the 3D maze in their physical environment. The objective is to navigate the tortoise through the maze, avoiding obstacles and reaching the finish line.

3 Key Features

- Augmented Reality: Place and interact with a 3D maze in the real world.
- Scratch-like Programming: Drag and drop blocks to control the tortoise's movements.
- Sequential Execution: Commands are executed in the order they are arranged.
- Interactive Obstacles: Various obstacles and interactive elements within the maze.

4 User Interface

Main UI Elements

- Start Block: The starting point for the sequence of commands.
- Side Bar: Contains available command blocks (e.g., Move Forward, Turn Left, Turn Right, Attack).
- Work Table: The area where players arrange command blocks to form a sequence.
- Info Text: Displays game instructions and feedback.
- Reset Button: Resets the current level.
- Levels Button: Provides access to different maze levels.
- Sprites: Visual representation of active and inactive UI states.

5 Technical Details: Scripts and functions

UIManager.cs

- Manages the visibility and state of UI elements.
- OnButtonClick(): Toggles the activation of all UI elements.

ResetCanva.cs

- Handles resetting the Blockly canvas by reloading the scene.

Direction.cs

- Defines movement directions and associated values.
- Enum Movements: Forward, Left, Right, Attack.

FinishLine.cs

- Handles reaching the finish line and loading the next level.
- OnTriggerEnter(): Checks for collision with the finish line and triggers the next level.

Executor.cs

- Executes the sequence of commands created by the player.
- Methods: MoveStraight(), RotateLeft(), RotateRight(), Attack(), IsFinish().

DragDrop.cs

- Handles drag-and-drop functionality for command blocks.
- Implements IBeginDragHandler, IDragHandler, IEndDragHandler.

DropPosition.cs

- Manages the drop positions for command blocks.
- Implements IDropHandler.

BlockDirection.cs

- Defines behavior for direction command blocks.
- Implements IBlock interface.

IBlock.cs

- Interface for command blocks.
- Properties: isDragged, isInMain.
- Method: Execute().

StartBlock.cs

- Initiates the execution of the command sequence from the start block.

TortoiseHandler.cs

- Controls the tortoise's movements and interactions in the maze.

6 Execution Flow

1. **UI Initialization**
   - The game starts with the `UIManager` initializing and managing the UI elements.
2. **Player Interaction**
   - Players drag and drop command blocks from the sidebar to the work table to create a sequence.
   - Each command block defines a movement or action (forward, left, right, attack).
3. **Command Execution**
   - The sequence starts from the `StartBlock`.
   - The `Executor` processes each block, sending movement commands to the `TortoiseHandler`.
4. **Tortoise Movement**
   - The `TortoiseHandler` moves or rotates the tortoise based on the commands.
   - The tortoise interacts with the maze environment, encountering obstacles and reaching the finish line.
5. **Level Progression**
   - Upon reaching the finish line, the `FinishLine` script triggers the loading of the next level.
   - If the tortoise does not follow the correct path, its position is resetted to its initial one.

7 Future Enhancements

- **Additional Levels:** Introduce more complex mazes with varying difficulty.
- **Level selector**: In the UI there will be some buttons to select a specific level to be played.
- **Power-ups and Collectibles**: Add items that enhance gameplay and provide rewards.

8 Conclusion

"AR Tortoise Maze Adventure" offers an engaging way for players to interact with augmented reality and learn basic programming concepts through a fun and educational game. The combination of AR and drag-and-drop programming provides a unique and immersive experience for players of all ages.