|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Untitled* use-case** | | |
| **Objective** |  |
| **Precondition** |  |
| **Main Flow** |  |
| **Alternative Flows** |  |
| **Post-condition** |  |

# Template

A screenshot of a white sheet

Description automatically generated

# Implemented in diagram

## Start Game Setup & Player Introduction

|  |  |
| --- | --- |
| **Flow of Events for the *Start Game Setup & Player Introduction* use-case** | |
| **Objective** | Familiarise all Players with the game rules and initiate the game with random potholes on the board. |
| **Precondition** | All Players are present, and the game has been launched. |
| **Main Flow** | 1. The Game System plays a tutorial showing Players how to fix potholes. 2. Players are guided through the rules, including movement and resource management. 3. Each Player rolls a die to determine playing order; the highest number goes first. 4. Players are placed at opposite corners of the grid. 5. A random number of small potholes (e.g. 4) are generated on the grid. |
| **Alternative Flows** | If the host wishes to change the default settings, extension point: *Adjust Point Values* |
| **Post-condition** | All Players are ready to play, and the initial game state is set with Players in position and potholes on the board. |

## Adjust Point Values

|  |  |
| --- | --- |
| **Flow of Events for the *Adjust Point Values* use-case** | |
| **Objective** | Customise the scoring system to adjust difficulty and game dynamics. |
| **Precondition** | The game is in setup mode, and no gameplay has started. |
| **Main Flow** | 1. The Player hosting the game accesses the game settings menu. 2. The host adjusts point values for pothole types:    1. e.g., increasing a small pothole’s point value from 1 to 2. 3. The Game System saves the new values, and Players are notified of the changes. |
| **Alternative Flows** | At 2, if the host chooses not to adjust any values, the game proceeds with default settings. |
| **Post-condition** | The game settings are locked, and the adjusted scoring system applies to the gameplay. |

## Resolve End of Round

|  |  |
| --- | --- |
| **Flow of Events for the *Resolve End of Round* use-case** | |
| **Objective** | Handle the end-of-round conditions, updating scores and resources, and preparing for the next round. |
| **Precondition** | All Players have completed their turn for the round. |
| **Main Flow** | 1. The Game System verifies that all Players have completed their actions for the round. 2. The Game System updates the scores based on the Players’ completed actions. 3. The Game System refreshes each Player’s resources. 4. The Game System applies any round-based events, such as:    1. Any unrepaired potholes increase in size. For example:       1. Small → Medium       2. Medium → Large    2. New potholes may randomly appear on the grid:       1. The Game System checks if any new potholes will be added based on predefined randomness (e.g., a generated number) |
| **Alternative Flows** | * At 4, if there are no unrepaired potholes, a winner is declared, and the game concludes – extension point: *Declare Game Winner.* * At 4, if new potholes are to be added, they appear randomly on the grid as small potholes. * At 4, if no new potholes are added, Players continue with the existing grid state. |
| **Post-condition** | The game state is updated for the next round, with Players' scores updated and resources replenished, existing potholes growing larger, and new potholes potentially appearing. |

## Declare Game Winner

|  |  |
| --- | --- |
| **Flow of Events for the *Declare Game Winner* use-case** | |
| **Objective** | Declare a winner when a Player reaches the target score or when all Potholes have been repaired. |
| **Precondition** | * A Player’s score is close or past the winning threshold (e.g., 50 points), or there are no more potholes left on the board. * The Game System checks the Player’s score and the state of the board at the end of each round. |
| **Main Flow** | 1. Pause the game. 2. Announce the Player as "King of Makers Valley." 3. Display a summary of notable accomplishments (e.g., key pothole repairs) |
| **Alternative Flows** | At 2, if all potholes were repaired before someone crossed the threshold, declare the Player with the most points as the winner. |
| **Post-condition** | The game concludes with a clear winner, and the final scores are displayed. |

## Allocate Starting Resources

|  |  |
| --- | --- |
| **Flow of Events for the *Allocate Starting Resources* use-case** | |
| **Objective** | Distribute initial resources (e.g., materials) to each Player at the start of the game. |
| **Precondition** | The game has been set up, and all Players have been identified and positioned on the board. |
| **Main Flow** | 1. The Game System calculates the starting resources each Player should receive. 2. The Game System allocates resources equally (or according to predetermined rules) to each Player. 3. Each Player's resources are displayed on the game interface. |
| **Alternative Flows** | N/A |
| **Post-condition** | Each Player starts the game with their allocated resources displayed, ready for gameplay. |

## Track Player Score

|  |  |
| --- | --- |
| **Flow of Events for the *Track Player Score* use-case** | |
| **Objective** | Update the Player’s score based on the actions they complete during their turn, particularly after repairing potholes or gaining knowledge. |
| **Precondition** | The Player has taken a turn and completed an action that affects their score (e.g., repairing a pothole). |
| **Main Flow** | 1. The Game System adds the corresponding points to the Player’s score. 2. The updated score is displayed to the Player on the game interface. |
| **Alternative Flows** | N/A |
| **Post-condition** | The Player’s score is accurately updated and reflects any points gained during the turn. The updated score is displayed on the game interface. |

## Move Player

|  |  |
| --- | --- |
| **Flow of Events for the *Move Player* use-case** | |
| **Objective** | Allow a Player to move across the grid and potentially repair potholes and advance knowledge to gain points. |
| **Precondition** | It’s the Player’s turn. |
| **Main Flow** | 1. The Player initiates a dice roll. 2. The Game System generates a random number (based on the dice roll) to determine movement. 3. The Game System displays the result to the Player. 4. The Player moves horizontally or vertically, respecting the number of rolled squares. |
| **Alternative Flows** | 1. If all Players have completed a turn – extension point: *Resolve End of Round*. 2. At 4, if the Player lands on a Knowledge square – extension point: *Encounter Knowledge* 3. At 4, if the Player lands on a Pothole square – extension point: *Encounter Pothole* 4. At 4, if the Player lands on a Resource square – extension point: *Encounter Resource* |
| **Post-condition** | The Player's turn ends, with points updated. |

## Encounter Pothole

|  |  |
| --- | --- |
| **Flow of Events for the *Encounter Pothole* use-case** | |
| **Objective** | Allow the Player to encounter a pothole on the board and decide whether to repair it using their materials to earn Satisfaction points. |
| **Precondition** | * The Player has landed on a square with a pothole during their turn. * The Player has a certain number of materials available for repairs. |
| **Main Flow** | 1. The Game System detects that the Player has landed on a Pothole square and displays the pothole’s size (e.g., small, medium, or large). 2. The Game System shows the Player an option to repair the pothole along with the material cost and points that would be earned:    * Small Pothole: Costs 1 material, awards 1 point.    * Medium Pothole: Costs 2 materials, awards 3 points.    * Large Pothole: Costs 3 materials, awards 5 points. 3. The Player chooses whether to enact a repair. |
| **Alternative Flows** | * At 3, if the Player does not have enough resources to repair the encountered pothole, the Game System displays an error message. Then, the Player skips the repair and ends their turn. * At 3, if the Player wants to conserve resources for a future opportunity, they skip the repair and end their turn. * At 3, if the Player enacts the repair, the Game System deducts the required materials from the Player’s inventory and updates their score. |
| **Post-condition** | * The Player’s score and material count are updated based on their choice. * The encounter concludes, allowing the game to proceed with the next turn or action in the game loop. |

## Encounter Knowledge

|  |  |
| --- | --- |
| **Flow of Events for the *Encounter Knowledge* use-case** | |
| **Objective** | Allow the Player to answer a question based on their existing knowledge for points. |
| **Precondition** | The Player has landed on a Knowledge square. |
| **Main Flow** | 1. The Game System detects that the Player has landed on a Knowledge square. 2. The Game System presents a multiple-choice or true/false question to the Player, relevant to the game’s themes. 3. The Player selects their answer within a set time limit. 4. The Game System evaluates the Player’s response. |
| **Alternative Flows** | * At 2, if the Player decides to exit the question screen without answering, the Game System cancels the question attempt, and the Player receives no points. * At 4, if correct, the Player is awarded a set number of extra Knowledge points, which are added to their score. * At 4, if incorrect, the Player receives the regular number of Knowledge points, and feedback with the correct answer is displayed for learning purposes. |
| **Post-condition** | The Player's score is updated based on the answer received, and their turn proceeds, potentially with a better understanding of the topic from the feedback provided. |

## Encounter Resource

|  |  |
| --- | --- |
| **Flow of Events for the *Encounter Resource* use-case** | |
| **Objective** | Allow the Player to replenish their materials by landing on a Resource square, with the quantity of materials received based on their accumulated knowledge points. |
| **Precondition** | * The Player has landed on a Resource square during their turn. * The Player has an existing score of knowledge points (tracked by the system) to determine the number of materials they will receive. |
| **Main Flow** | 1. The Game System detects that the Player has landed on a Resource square and displays a notification of the event. 2. The Game System calculates the number of materials to be awarded to the Player based on their current knowledge points:    1. **Low Knowledge** (0–5 points): 1 material awarded.    2. **Moderate Knowledge** (6–10 points): 2 materials awarded.    3. **High Knowledge** (11+ points): 3 materials awarded. 3. The Player’s material inventory is updated by adding the awarded materials. 4. The Game System displays the updated material count to the Player. 5. The Player is notified of the quantity of materials they received and reminded that knowledge points improve material rewards. |
| **Alternative Flows** | N/A |
| **Post-condition** | * The Player’s material inventory is replenished based on their knowledge points. * The encounter concludes, allowing the game to proceed with the next turn or action in the game loop. |

# Come back to later

## Use Resource Card

|  |  |
| --- | --- |
| **Flow of Events for the *Use Resource Card* use-case** | |
| **Objective** | Allow a Player to use a special resource card that can impact gameplay, like gaining extra resources or modifying a pothole. |
| **Precondition** | The Player has a resource card in their possession. |
| **Main Flow** | 1. The Player selects a resource card from their hand. 2. The Player activates the card by placing it on the board or selecting it in the game interface. 3. The Game System processes the card's effect (e.g., awarding resources, adjusting another Player’s score, modifying a pothole’s repair cost). 4. The Game System notifies all Players of the card's effect. |
| **Alternative Flows** | At 2, if the card requires resources to activate, the Player must have the necessary resources. If not, the use case ends without effect. |
| **Post-condition** | The card's effect has been applied, and the card is removed from the Player’s hand. |

## Draw Pothole Challenge Card

|  |  |
| --- | --- |
| **Flow of Events for the *Draw Pothole Challenge Card* use-case** | |
| **Objective** | Introduce randomness by drawing a challenge card that affects pothole conditions or Player resources. |
| **Precondition** | The Player lands on a specific square or condition that triggers a challenge card. |
| **Main Flow** | 1. The Player picks a challenge card from a deck. 2. The Game System reveals the challenge (e.g., an increase in pothole size, resources stolen by a competitor). 3. The Player must react to the challenge, which could involve adjusting resources or planning new moves. |
| **Alternative Flows** | If the Player lacks resources to handle the challenge, they lose points or miss their next turn. |
| **Post-condition** | The challenge card's effect has been resolved, and it is discarded or returned to the deck. |

## Challenge Opponent’s Move

|  |  |
| --- | --- |
| **Flow of Events for the *Challenge Opponent’s Move* use-case** | |
| **Objective** | Allow a Player to challenge another Player’s move if they suspect an error or want a verification of rules. |
| **Precondition** | The opponent has completed a move that is challengeable (e.g., repairing a pothole or using a resource card). |
| **Main Flow** | 1. The Player indicates a challenge to the Game System. 2. The Game System verifies the opponent’s move against the rules. |
| **Alternative Flows** | * At 2, if the move is valid, the Game System notifies both Players, and the game resumes. * At 2, if the move is invalid, the Game System reverses the action and applies any penalties. |
| **Post-condition** | The move is verified, and play continues with any necessary adjustments. |

## Play Tutorial

|  |  |
| --- | --- |
| **Flow of Events for the *Play Tutorial* use-case** | |
| **Objective** | Allow the Player to complete a tutorial for knowledge gain, with a follow-up question that offers points or resources. |
| **Precondition** | The Player has selected the option to "Play Tutorial" on the Knowledge square. |
| **Main Flow** | 1. The Game System launches a brief tutorial (e.g., animation, text, or interactive steps) that teaches the Player about a topic related to the game’s context (e.g., resource management, repair techniques). 2. After completing the tutorial, the Game System presents a follow-up question based on the tutorial content. – *include(Answer Knowledge Question)* |
| **Alternative Flows** | N/A |
| **Post-condition** | The Player's score is updated based on the answer received, and their turn proceeds, potentially with a better understanding of the topic from the feedback provided. |