|  |  |  |
| --- | --- | --- |
| **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

## Allocate Starting Resources

|  |  |
| --- | --- |
| **Flow of Events for the *Allocate Starting Resources* use-case** | |
| **Objective** | Distribute initial resources (e.g., materials, points, or tokens) 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. |

|  |  |
| --- | --- |
| **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:       1. Small → Medium       2. Medium → Large    2. New potholes may randomly appear on the grid (starting as small) – include*(Handle New Potholes Mid-Game)* |
| **Alternative Flows** | At 4, if there are no unrepaired potholes, no size increases occur but new potholes may still appear. |
| **Post-condition** | The game state is updated for the next round, with Players' scores updated and resources replenished, and potholes potentially larger. |

## Resolve End of Round

## Handle New Potholes Mid-Game

|  |  |
| --- | --- |
| **Flow of Events for the *Handle New Potholes* use-case** | |
| **Objective** | Introduce unexpected challenges by adding new potholes after each round. |
| **Precondition** | A new round is beginning, and the game determines whether new potholes should appear. |
| **Main Flow** | 1. The Game System checks if any new potholes will be added based on predefined randomness (e.g., a generated number) 2. If new potholes are to be added, they appear randomly on the grid as small potholes. |
| **Alternative Flows** | At 2, if no new potholes are added, Players continue with the existing grid state. |
| **Post-condition** | The grid is updated, and new challenges are introduced for the Players. |

## Determine Game Winner

|  |  |
| --- | --- |
| **Flow of Events for the *Determine Game Winner* use-case** | |
| **Objective** | Declare a winner when a Player reaches the target score. |
| **Precondition** | A Player’s score is close to the winning threshold (e.g., 50 points) |
| **Main Flow** | 1. The Game System checks the Player’s score at the end of each turn – include(*Track Player Score)* 2. If the Player’s score is 50 or higher:    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** | N/A |
| **Post-condition** | The game concludes with a clear winner, and the final scores are displayed. |

## 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. |
| **Precondition** | * The Player has taken a turn and completed an action that affects their score (e.g., repairing a pothole). * The Game System is tracking points based on pothole repairs. |
| **Main Flow** | 1. The Player completes an action (such as repairing a pothole) that impacts their score. 2. The Game System evaluates the type of pothole repaired:    1. **Small Pothole** = 1 point    2. **Medium Pothole** = 3 points    3. **Large Pothole** = 5 points 3. Based on the pothole type, the system adds the corresponding points to the Player’s score. 4. The updated score is displayed to the Player on the game interface. |
| **Alternative Flows** | At 1, if the Player encounters a pothole but does not have enough resources to repair it, no points are added to their score, and they proceed to the next part of their turn. |
| **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. |

## 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 an interactive 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 10x10 grid. 5. Four random potholes are generated on the grid (with random sizes: small, medium, large). |
| **Alternative Flows** | If the host wishes to change the default settings, the use case *Adjust Point Values* will be called. |
| **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. |

## Move Player & Manage Resources

|  |  |
| --- | --- |
| **Flow of Events for the *Move Player & Manage Resources* use-case** | |
| **Objective** | Allow a Player to move across the grid and repair potholes to gain points. |
| **Precondition** | It’s the Player’s turn and they have the resources to fill potholes if encountered. |
| **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 tiles. 5. If the Player lands on a tile with a pothole, they can choose to repair it using their materials:    1. **Small Pothole**: Costs 1 material, awards 1 point.    2. **Medium Pothole**: Costs 2 materials, awards 3 points.    3. **Large Pothole**: Costs 3 materials, awards 5 points. 6. Points are added to the Player's score based on the repaired pothole’s size – *include(Track Player Score)* |
| **Alternative Flows** | * At 3, if the Player does not have enough resources to repair the encountered pothole, they skip the repair and end their turn. * At 3, if the Player wants to conserve resources for a future opportunity, they skip the repair and end their turn. * If all Players have completed a turn, the use case *End Round & Manage Resources* will be called. |
| **Post-condition** | The Player's turn ends, with points updated and potholes potentially repaired. |

## Plan Resource Use Strategically

|  |  |
| --- | --- |
| **Flow of Events for the *Plan Resource Use Strategically* use-case** | |
| **Objective** | Encourage Players to make tactical decisions about resource allocation. |
| **Precondition** | The Player has moved and encountered one or more potholes within reach. |
| **Main Flow** | 1. The Player assesses their current resources and available potholes. 2. Decide whether to repair a pothole immediately or conserve resources for future opportunities – include(*Move Player & Manage Resources*) |
| **Alternative Flows** | N/A |
| **Post-condition** | The turn ends, with resources used or conserved based on the Player’s decision. |

## 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 or play a short tutorial to gain new knowledge. |
| **Precondition** | The Player has landed on a Knowledge tile. |
| **Main Flow** | 1. The Game System detects that the Player has landed on a Knowledge tile. 2. The Player is presented with two options:    1. **Answer Knowledge Question** (for Players confident in their knowledge).    2. **Play Tutorial** (for Players who want to learn more). |
| **Alternative Flows** | * At 2, if the Player selects Answer Knowledge Question, the *Answer Knowledge Question* use case will be called. * At 2, if the Player selects Play Tutorial, the *Play Tutorial* use case will be called. * At 2, if the Player chooses to skip both the question and tutorial, the Game System allows the Player to continue with their turn without any knowledge-related reward or penalty. |
| **Post-condition** | Depending on the chosen path, the Player may gain points, resources, or additional knowledge that could assist in future turns. |

## 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 tile. |
| **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. |

## Answer Knowledge Question

|  |  |
| --- | --- |
| **Flow of Events for the *Answer Knowledge Question* use-case** | |
| **Objective** | Allow the Player to answer a question for points, rewarding correct knowledge. |
| **Precondition** | The Player has selected the option to "Answer Knowledge Question" on the Knowledge tile, or the Player has watched the tutorial after selecting “Play Tutorial” |
| **Main Flow** | 1. The Game System presents a multiple-choice or true/false question to the Player, relevant to the game’s themes. 2. The Player selects their answer within a set time limit. 3. The Game System evaluates the Player’s response. |
| **Alternative Flows** | * At 1, 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 3, if correct, the Player is awarded a set number of points, which are added to their score. * At 3, if incorrect, the Player receives no 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. |

# 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 tile 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. |