Use Case: *encounterPunishment*

Primary Actor: Squirrel (Main Character)

Goal in Context: To subtract a number of points when the Squirrel encounters a cell with a punishment.

Preconditions: The game must be in play and the Squirrel has made a move. Certain cells has designated punishment and the enemy must have made a move.

Trigger: The Squirrel moves one cell in a direction and is detected by the sensor after a tick of the game.

Scenario:

1. Squirrel: Moves a tick.
2. Squirrel: The cell has a punishment.
3. Squirrel: Penalized a number of points that the punishment is evaluated to have.
4. Squirrel: The punishment is removed from the cell.

Exceptions:

1. If the points go below 0 then the game ends.
2. If the cell has a moving enemy instead of a punishment the game ends instead.
3. If the cell has a removed punishment, then nothing happens.

Priority: Immediate as this concerns the core functions of the game.

Availability: When the Squirrel moves a tick, after the first move of the game.

Frequency of Use: Not many times, as it only occurs with specific conditions and after each tick.

Channel to Primary Actor: User interface such as keyboard presses.

Secondary Actors: Cell sensors that keep track of the punishment/moving enemy and Squirrel.

Channels to Secondary Actors: Game A.I

Open Issues:

1. How would the player see that it is a punishment or a moving enemy?
2. What has priority, the punishment or the moving enemy in relation to score?

Use Case: *encounterBonusReward*

Primary Actor: Squirrel (Main Character)

Goal in Context: The addition of a number of points when the Squirrel encounter a bonus reward.

Preconditions: The game must still be in play and the Squirrel has made a move in any direction. The cells have to be populated with a bonus reward.

Trigger: The Squirrel moves once cell in a direction and is detected by the sensor after a tick of the game.

Scenario:

1. Squirrel: Moves a tick.
2. Squirrel: The cell has a bonus reward.
3. Squirrel: Rewarded a number of points that the bonus reward is evaluated to have.
4. Squirrel: The bonus reward is removed from the cell.

Exceptions:

1. There is a moving enemy or punishment in that cell.
2. The game has ended as the cell also contained the final regular reward.
3. The cell no longer contains the bonus reward.

Priority: Immediate as this relates to the core function of the game.

Availability: It should be available once the Squirrel has made a move.

Frequency of Use: After each tick or movement from the Squirrel

Channel to Primary Actor: The UI

Secondary Actors: Cell Sensors as well as the bonus reward itself.

Channels to Secondary Actors: The game AI.

Open Issues:

1. The cell contains the final regular reward and a bonus reward, there is a priority concern.
    The number of ticks that the bonus reward spends in that specific cell.
2. A moving enemy moves into a cell that contains the bonus reward and the interactions resulting from that.
3. What happens when there is the final reward, a punishment, a bonus reward, and a moving enemy?

Use Case: *encounterEnemy*

Primary Actor: Squirrel (Main Character)

Goal in Context: Ending of the game once encounter with a moving enemy.

Preconditions: The game must be in play and the Squirrel has made a move. The moving enemy must have made a move.

Trigger: The Squirrel moves once cell in a direction and is detected by the sensor after a tick of the game. The cell contains the moving enemy.

Scenario:

1. Squirrel: Moves a tick.
2. Squirrel: The cell has a moving enemy.
3. Squirrel: The game ends.

Exceptions:

1. The game has started and the moving enemy is 1 cell away from the main character

Priority: Immediate as this concerns the core functions of the game.

Availability: When the Squirrel moves a tick, after the first move of the game.

Frequency of Use: Not many times, as it only occurs with specific conditions and after each tick.

Channel to Primary Actor: User interface such as keyboard presses.

Secondary Actors: Cell sensors that keep track of the punishment/moving enemy and Squirrel.

Channels to Secondary Actors: Game A.I

Open Issues:

1. How would the moving enemy determine the closest cell towards the enemy when there are obstacles such as walls.

Use Case: *encounterReward*

Primary Actor: Squirrel (Main Character)

Goal in Context: The calculation of points as well as the ending of the game by collecting all rewards.

Preconditions: The game must be in play and the Squirrel has made a move. The moving enemy must have made a move.

Trigger: The Squirrel moves once cell in a direction and is detected by the sensor after a tick of the game. The cell contains the moving enemy.

Scenario:

1. Squirrel: Moves a tick.
2. Squirrel: The cell has a reward.
3. Squirrel: Rewarded a number of points that the reward is evaluated to have.
4. Squirrel: The reward is removed.
5. Squirrel: The number of needed rewards decreases by 1.

Exceptions:

1. The cell contains the moving enemy and so the game ends immediately.
2. There are no more rewards to collect and the game ends immediately.

Priority: Immediate as this concerns the core functions of the game.

Availability: Before the game starts as the cells must populate with the rewards and the checks begin once the Squirrel has made a move.

Frequency of Use: Not many times, as it only occurs with specific conditions and after each tick

Channel to Primary Actor: User interface such as keyboard presses

Secondary Actors: Cell sensors that keep track of the punishment/moving enemy and Squirrel.

Channels to Secondary Actors: Game A.I

Open Issues:

1. The initial tile cannot have any rewards and the rewards cannot stack.
2. How would it be differentiated from other objects on the map?
3. Would it be hidden?