

Project Overview

1. Purpose/Objective

In Prison Escape players are faced with the goal of escaping Arkham prison. Filled with guards and traps, players are forced to collect all the keys scattered around the map and escape

2. Game Details

2.1 Main Character

The main character of the game consist of a prison inmate (dressed in white and black jail clothes)

2.2 Enemies

Moving Enemies: Consists of jail guards that are scattered around the map

Punishments: Sensor traps that once the player comes in contact with it, the player will lose points

2.3 Location

Setting: Prison

Initial Position: inside of a prison cell

End Location: Gate/main door

2.4 Score

The score system is based around the number of keys and chicken drumsticks collected

2.5 Rewards

Regular rewards: Keys are collected that opens the door to win the game

Bonus rewards: Clocks are scattered around the map that once collected grant increased round time

Chicken drumsticks/food that just gives the player a score increase

2.6 Barriers

There will be barriers like concrete walls, beds, toilets, barrels, tables, and locked doors

2.7 Difficulty

The game increases in difficulty lowering the completion times for every round, increasing number of enemies (guards and traps), and increasing overall map size

2.8 Restrictions

Players cannot go outside of the map and through the listed barriers

3. Team Plan

We have opted to adopt a scrum-based development process model with frequent 15-30 minute meetings every two days, either remote or in person, as situations and personal schedules allow. Remote meetings are held in discord as the platform allows for both quick chat message updates which helps notify teammates of the overall progress and the project timeline, as well as voice calls with screen sharing features for easier discussion of the issues at hand. With respect to responsibility distribution, currently, Ilia is developing the project mockups to create an image of what the game will look like. Simon has designed the UML diagrams portraying the various complex relationships between the different entities and objects that will be present in the final game. The UML diagrams also help bring clarity to the coding phase when the team will begin programming the game. Parshan is responsible for the various use cases of our project which depict the interactions between the features of our project, including collectibles, enemy entities, characters, maps, and game flow. Finally, Vishal has undertaken the documentation and planning accompanying the "Prison Escape" game. This includes the detailed listing and logging of requirements, specifications, and features throughout the process. These responsibilities are subject to change in the future. In the next phase of the project, all members of the team will be involved in the programming, UI design and backend development. We have decided to split the team into two pairs in the future, with one pair working on the front-end development and the other pair working on the backend aspect of the game.

Individual progress will be monitored and rectified as soon as a backlog is discovered. This is enforced in the form of a short run down of individual progress at the beginning of every meeting. This is an efficient method to quickly detect any potential hazards that may be detrimental to the timely progression and accomplishment of set goals on the project progression timeline. These hazards are rectified through quick re-assignments of responsibilities with members with lower workloads during the week of backlog discovery being added to the lagging task at hand. Sprints are also being integrated into the workflow with a shorter weekly time period as opposed to the usual 30-day time period. As such, the predefined backlogs and goals to be accomplished within this shorter time frame are smaller than usu

Use Cases Template

Use case: Start new game

Iteration: 2. Last modification 2/17/22

Primary actor: User/Player

Goal in context: To begin the game and pick map difficulty

Preconditions:

- The game application is running
- User sees the Main Menu and the “New game”, “Continue”, and “Quit Game” buttons
- The “New game” button in the main menu is functional
- Difficulty display window is functional, view use case **Selecting game difficulty**
- Window manager can switch windows/displays

Trigger: Pressing the “New game” button in the games main menu window

Scenario:

1. Running Game application and user sees the main menu
2. User clicks the “New game” button is clicked in the main menu display window
3. User now sees the difficulty display window containing the buttons “Easy”, “Normal”, and “Hard”
4. The user clicks on one of the 3 difficulty button options
5. The game displays game window with the the first level of the difficulty option

Exceptions:

1. User does not select a difficulty level

Priority: Moderate priority

When available: 2nd-3rd sprint

Frequency of use: Frequent

Use case: Continue/Load game

Iteration: 1. Last modification 2/17/22

Primary actor: User/Player

Goal in context: Loads game from the last saved/last exited instance (game is saved automatically when the game console is closed)

Preconditions:

- The game application is running
- There exists a saved instance of a game
- User sees the Main Menu and the “New game”, “Continue”, and “Quit Game” buttons
- Main menu buttons are functional when the player presses on them
- Window manager switches windows/displays

Trigger: To load the progress made from a instance of a game played before

Scenario:

1. User exits game by clicking the “Return to main menu” button in the in-game menu (See use case **Return to main menu**)
2. The game now shows the main menu display window
3. Game instance automatically saves and stores with the locations of the all objects, characters, and the score
4. Game shows the main menu display window with a functional “Continue” button
5. “Continue” button is clicked
6. The saved Game instance is loaded and shown in the game window

Exceptions:

1. Player has not started a new game prior, see use case **Start new game**
2. Player never presses the “Return to main menu” button in the in-game menu, see use case **Return to main menu**

Priority: low priority, to be implemented after game is functional

When available: 3rd sprint

Frequency of use: Frequent

Use Cases Template

Use case: Main character moves up

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate (main character)

Goal in context: For inmate to move up to an adjacent cell

Preconditions:

- The user/player has started the game application (See use case **Continue/Load game or Start new game**)
- The Inmate should have been spawned in the game, see use case **Inmate Spawn**

Trigger: User presses or holds W key on the keyboard

Scenario:

1. Player can now see the map and the inmate (main character) that can be controlled
2. Player presses or holds the W key on the keyboard to move up on the map
3. After pressing/holding W key, Inmate stops moving

Exceptions:

1. There is a obstacle blocking the upward movement

Priority: high priority

When available: 1st sprint

Frequency of use: Very frequent

Use case: Main character moves down

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate (main character)

Goal in context: For inmate to move down to an adjacent cell

Preconditions:

- The user/player has started the game application (See use case **Continue/Load game or Start new game**)
- The Inmate should have been spawned in the game, see use case **Inmate Spawn**

Trigger: User presses or hold S key on the keyboard

Scenario:

1. Player can now see the map and the inmate (main character) that can be controlled
2. Player presses or holds the S key on the keyboard to move down on the map
3. After pressing/holding S key, Inmate stops moving

Exceptions:

1. There is a obstacle blocking the downward movement

Priority: high priority

When available: 1st sprint

Frequency of use: Very frequent

Use Cases Template

Use case: Main character moves right

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate (main character)

Goal in context: For inmate to move right to an adjacent cell

Preconditions:

- The user/player has started the game application (See use case **Continue/Load game or Start new game**)
- The Inmate should have been spawned in the game, see use case **Inmate Spawn**

Trigger: User presses or holds D key on the keyboard

Scenario:

1. Player can now see the map and the inmate (main character) that can be controlled
2. Player presses or holds the D key on the keyboard to move right on the map
3. After pressing/holding D key, Inmate stops moving

Exceptions:

1. There is a obstacle blocking the upward movement

Priority: high priority

When available: 1st sprint

Frequency of use: Very frequent

Use case: Main character moves left

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate (main character)

Goal in context: For inmate to move left to an adjacent cell

Preconditions:

- The user/player has started the game application (See use case **Continue/Load game or Start new game**)
- The Inmate should have been spawned in the game, see use case **Inmate Spawn**

Trigger: User presses or hold A key on the keyboard

Scenario:

1. Player can now see the map and the inmate (main character) that can be controlled
2. Player presses or holds the A key on the keyboard to move down on the map
3. After pressing/holding A key, Inmate stops moving

Exceptions:

1. There is a obstacle blocking the upward movement

Priority: high priority

When available: 1st sprint

Frequency of use: Very frequent

Use Cases Template

Use case: Return to main menu

Iteration: 1. Last modification 2/17/22

Primary actor: User/player

Goal in context: To return to the main menu through the in-game menu

Preconditions:

- A game instance is created or loaded, see use case **Start new game or Continue/Load game**

Trigger: Pressing the ESC key on the keyboard and Clicking the “Return to main menu” button in the in-game menu

Scenario:

1. User presses the ESC key on the keyboard to display the in-game display menu
2. Display menu contains “Resume” and “Return to main menu” buttons
3. User clicks on “Return to main menu” button
4. Game now displays the main menu display window
5. User selects the “Quit Game” button, see use case **Exit**

Exceptions: N/A

Priority: moderate priority

When available: 2nd sprint

Frequency of use: Frequent

Use case: Exit

Iteration: 1. Last modification 2/17/22

Primary actor: User/player

Goal in context: To exit the game through the main menu

Preconditions:

- The game application is running

Trigger: Clicking the “Return to main menu” button in the in-game menu

Scenario:

1. Users sees the main menu display window containing the “New game”, “Continue”, and “Quit Game” buttons
2. User clicks on the “Quit Game” button
3. User exits and game application window closes

Exceptions: N/A

Priority: moderate priority

When available: 2nd sprint

Frequency of use: Frequent

Use Cases Template

Use case: Inmate collides with clock

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate

Goal in context: To add 10 seconds to the round timer

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**

Trigger: The Inmate coming in contact with the clock reward

Scenario:

1. The game application is running
2. The user/player has started the game application (See use case **Continue/Load**)
3. Main character has spawned, see use case **Inmate spawn**
4. Inmate can move in the four cardinal directions towards the clock sprite/icon on the map, see use cases **Main character moves right, Main character moves left, Main character moves up, Main character moves down**
5. Inmate collides with the clock sprite/icon and increases the 10 seconds
6. Clock icon disappears from the map after collision

Exceptions: N/A

Priority: high priority

When available: 2nd sprint

Frequency of use: Very Frequent

Use case: Inmate collides with drumstick

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate

Goal in context: To add 100 points to the scoreboard

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**

Trigger: The Inmate coming in contact with the drumstick reward

Scenario:

1. The game application is running
2. The user/player has started the game application (See use case **Continue/Load**)
3. Main character has spawned, see use case **Inmate spawn**
4. Inmate can move in the four cardinal directions towards the drumstick sprite/icon on the map, see use cases **Main character moves right, Main character moves left, Main character moves up, Main character moves down**
5. Inmate collides with the drumstick sprite/icon and increases the score by 100 points
6. Drumstick icon disappears from the map after collision

Exceptions: N/A

Priority: high priority

When available: 2nd sprint

Frequency of use: Very Frequent

Use Cases Template

Use case: Inmate collides with Key

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate

Goal in context: To unlock the gate and escape the prison maze

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**
- Inmate movements should be operational
- Scoring system should be functional

Trigger: The Inmate coming in contact Key reward/collectable

Scenario:

1. The game application is running
2. The user/player has started the game application (See use case **Continue/Load**)
3. Main character has spawned, see use case **Inmate spawn**
4. Inmate can move in the four cardinal directions towards the clock sprite/icon on the map, see use cases **Main character moves right, Main character moves left, Main character moves up, Main character moves down**
5. Inmate collides with the key sprite/icon and increases the score by 200 points
6. Key icon disappears from the map after collision

Exceptions: N/A

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Use case: Player wins round collecting all 10 key

Iteration: 1. Last modification 2/17/22

Primary actor: Player

Goal in context: To win the round by collecting all scattered 10 scattered map gate keys

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**
- Inmate should be able to collide with keys, see use case **Inmate collides with key**
- Inmate should be spawned on map, see use case **Inmate Spawn**

Trigger: The Inmate coming in contact Key reward/collectable

Scenario:

1. Inmate collides with the cell containing the key sprite/icon
2. On collision the scoreboard gets updated; increasing the score by 200 points
3. Key icon disappears from the map after collision
4. Collecting all 10 keys grants the user entry through the prison gate to escape the prison
5. Inmate collides with the cell containing the prison gate

Exceptions: N/A

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Use Cases Template

Use case: Inmate collides with static enemy

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate

Goal in context: To penalize the score by the amount of the punishment and to end game if score is below zero

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**
- Inmate movements should be operational
- Scoring system should be functional
- Player score is not lower than 0 beforehand

Trigger: The Inmate coming in contact with the static enemy.

Scenario:

1. The game application is running
2. The user/player has started the game application (See use case **Continue/Load**)
3. Main character has spawned, see use case **Inmate spawn**
4. Inmate can move in the four cardinal directions towards the clock sprite/icon on the map, see use cases **Main character moves right**, **Main character moves left**, **Main character moves up**, **Main character moves down**
5. Inmate collides with the static enemy icon and the score is decreased by 200 points
6. Static enemy icon disappears from the map after collision
7. Game ends if score is below zero after deduction, see use case **Game over**.

Exceptions: N/A

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Use case: Inmate collides with dynamic enemy

Iteration: 1. Last modification 2/17/22

Primary actor: Inmate

Goal in context: To end the game and display the **Game over** screen.

Preconditions:

- The game application is running
- A game instance is created or loaded, see use case **Start new game or Continue/Load game**
- Inmate movements should be operational
- Scoring system should be functional

Trigger: The Inmate coming in contact with the dynamic enemy.

Scenario:

1. The game application is running
2. The user/player has started the game application (See use case **Continue/Load**)
3. Main character has spawned, see use case **Inmate spawn**
4. Inmate can move in the four cardinal directions towards the clock sprite/icon on the map, see use cases **Main character moves right**, **Main character moves left**, **Main character moves up**, **Main character moves down**
5. Inmate collides with the dynamic enemy icon.
6. Dynamic enemy icon disappears from the map after collision
7. Game ends and the **Game over** screen is displayed, see use case **Game over**.

Exceptions: N/A

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Use Cases Template

Use case: Game Over

Iteration: 1. Last modification 2/17/22

Primary actor: User/player

Goal in context: To end the game and display the **Game over** screen which includes **restart** and **exit** use case options.

Preconditions:

- The game application is running
- Either inmate score is below zero or the inmate has collided with a moving dynamic enemy.
- The game has ended.

Trigger: The Inmate coming in contact with the dynamic enemy or the player score is below zero.

Scenario:

1. The game application is running
2. Inmate collides with the dynamic enemy icon or player score drop below zero.
3. Dynamic enemy icon disappears from the map after collision.
4. Game ends and the **Game over** screen is displayed.

Exceptions:

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Use case: Instructions

Iteration: 1. Last modification 2/17/22

Primary actor: User/player

Goal in context: To display instructions to the user on how to play the game, this includes main character movements, winning, and losing conditions.

Preconditions:

- The game application is running
- The user/player has started the game
- Either inmate score is below zero or the inmate has collided with a moving dynamic enemy.
- The game has ended.

Trigger: The user chooses the **Instructions** button on the main menu.

Scenario:

1. The game application is running
2. Inmate can move in the four cardinal directions towards the clock sprite/icon on the map, see use cases **Main character moves right**, **Main character moves left**, **Main character moves up**, **Main character moves down**.
3. The player has not begun playing the actual game.

Exceptions:

Priority: high priority

When available: 1st sprint

Frequency of use: Very Frequent

Project Mockups



Main menu of Prison Escape.
The user has the option of starting a new game, continuing where they left off, or quit the game.

Option to choose the difficulty. The player is given 3 difficulty options to choose from. The easy difficulty consists of fewer enemies and more time to complete each level. As the difficulty increases, more enemies will be present on the map and you have less time to complete the level.



The main character's starting position is at one of the cell blocks. The player must collect all the keys located around the map to unlock the exit door. While scattering around looking for the keys, the player must avoid contact with guards and traps in order to successfully finish the level.

A screen showing the stats of the players performance on the level that was successfully completed. The player is able to move on to the next level or return to the main menu.



Game over screen when the player fails. The game will end if the player collides with an enemy or the timer on the level reaches zero. The player has the option to retry the level or exit to the main menu.

The player has the ability to pause the game while playing. The pause menu has two options for the player, one option is to resume the game and the other is to return to the main menu. If the player decides to leave the game, the game will automatically save the current level, player and enemy positions, and current player stats.



UML Design

