Part 1: Description of overall plan and customized game

1. **Theme:** Our theme is thief escape. The thief steals money, and the police come to arrest him.

2. **Main Character - Thief:**

- The main character for this game is a thief, which is controlled by the player through the keyboard. The thief can move up/down/left/right to an adjacent cell, if there are no barriers on the target cell. The player can move at most one cell at each "tick" of the game.
- The thief will not move if the player doesn't press a valid key, or if the move is not valid. Even in such cases, the enemies will move.

3. **Enemies:**

Moving Enemy - Police:

- The police that can move one cell up/down/left/right at each tick.
- The police move towards a direction that makes them closest to the current position of the thief. If the thief and the police meet each other (go into the same cell), the player loses right away.
- The police can't go through the walls/barriers either.

Punishment - Bombs:

- If the thief moves to a cell that contains a bomb, player will lose 1 score.
- The player loses if the punishment causes the overall score to become negative.

4. **Rewards**

Regular Reward - Money bills:

- The thief has to collect all the bills to win the game.
- Once the player moves to the cell containing the money bill, the reward is claimed (removed from the map) and 1 point is added to total score.

Bonus Reward - Diamonds:

- The diamonds contain a higher reward compared to the money bills (2 points).
- A diamond appears randomly during the game, and disappears after a while (a few ticks).

5. **Barriers:**

- The board is surrounded by walls from four sides. There are only two openings for the start and end points. There are other individual barriers on the map cells such as buildings, which also block the movement of the main character and enemies to those cells.

6. **Board - City Map:**

- The game board is a 2D grid city map, surrounded by walls (buildings).
- An initial map for the board is loaded when the player starts the game with all the game components.
- The board has a start point and an end cell. The game screen shows the current score and the time that has passed since the start of the game. The final score and time are shown when the player wins.

Part 2: Use-cases

Use-case 1: Claiming rewards
Primary actor: Main character (thief).
Goal in context: To gain rewards in order to receive points and win game.
Preconditions: The character is still in game (the character's point is not below 0, has not touched/moved to the same cell of any moving enemies), there is no (closed) wall between the reward and main character.
Trigger: The player wants to earn all the points and win the game.

Scenario:

1. The player moves the character to the cell of the reward by using the 4 buttons up/down/left/right.
2. The reward disappears as soon as the character enters the reward cell.
3. The point increases automatically by 1 point.

Exceptions:

1. The player touches a bonus - see use-case: Claiming bonuses.
2. The player touches a moving enemy or a moving enemy catches up with the player - see use-case: Attacked by a moving enemy.
3. The player touches a punishment - see use-case: Attacked by a punishment.
4. The player tries to go through a wall/barrier - see use-case: Going through a wall/barrier.

Priority: Moderate priority - to be implemented after basic functions.
Frequency of use: Frequent.

Use-case 2: Claiming bonuses
Primary actor: Main character (thief).
Goal in context: To gain bonuses in order to receive more points.
Preconditions: The character is still in game (the character's point is not below 0, has not touched/moved to the same cell of any moving enemies), there is no (closed) wall between the bonus and main character. The bonus is still there at the moment the main character enters the cell containing the bonus.
Trigger: The player decides to earn more than the amount of points of a regular reward.

Scenario:

1. The player moves the character to the cell of the bonus by using the 4 buttons up/down/left/right.
2. The reward disappears as soon as the character enters the bonus cell.
3. The point increases automatically by 2 points.

Exceptions:

1. The player touches a reward - see use-case: Claiming rewards.
2. The player touches a moving enemy or a moving enemy catches up with the player - see use-case: Attacked by a moving enemy.
3. The player touches a punishment - see use-case: Attacked by a punishment.
4. The player tries to go through a wall/barrier - see use-case: Going through a wall/barrier.

Priority: Moderate priority - to be implemented after basic functions.

Frequency of use: Infrequent.

Use-case 3: Attacked by a moving enemy
Primary actor: Main character (the thief), a moving enemy.
Goal in context: To kill the main character.
Preconditions: The character is still in game (the character's point is not below 0, has not touched/moved to the same cell of any moving enemies), there is no (closed) wall between the moving enemy and main character. There is at least a moving enemy present.
Trigger: Moving enemy aims to attack the main character.

Scenario:

1. The player is focused on claiming the rewards/bonuses.
2. The player moves the character so close to the moving enemy that the enemy can catch up with the main character/ The player accidentally touches the moving enemy.
3. The screen displays "GAME OVER" and an option to replay the game as soon as the player and the moving enemy go into the same cell.

Exceptions:

1. The player touches a reward - see use-case: Claiming rewards.
2. The player touches a bonus - see use-case: Claiming bonuses.
3. The player touches a punishment - see use-case: Attacked by a punishment.
4. The player tries to go through a wall/barrier - see use-case: Going through a wall/barrier.

Priority: Lower priority - to be implemented after basic, reward/bonus functions.
Frequency of use: Frequent.

Use-case 4: Attacked by a punishment
Primary actor: Main character (the thief)
Goal in context: To reduce the points gained by the main character.
Preconditions: The character is still in game (the character's point is not below 0, has not touched/moved to the same cell of any moving enemies), there is no (closed) wall between the punishment and the main character. There is at least a punishment present.
Trigger: The player accidentally (or decides to) goes into a cell that contains punishment.

Scenario:

1. The player moves the character to the cell containing the punishment by using the 4 buttons up/down/left/right.
2. The punishment disappears as soon as the character enters the punishment cell.
3. The point decreases automatically by 1 point.
4. If the total score of the player drops below 0 as a result, the screen displays "GAME OVER" and an option to replay the game.

Exceptions:

1. The player touches a reward - see use-case: Claiming rewards.

2. The player touches a bonus - see use-case: Claiming bonuses.
3. The player touches a moving enemy- see use-case: Attacked by a moving enemy.
4. The player tries to go through a wall/barrier - see use-case: Going through a wall/barrier.

Priority: Lower priority - to be implemented after basic, reward/bonus functions.
Frequency of use: Infrequent.

Use-case 5: Going through a wall/barrier
Primary actor: Main character (the thief)
Goal in context: To take a shortcut.
Preconditions: The character is still in game (the character's point is not below 0, has not touched/moved to the same cell of any moving enemies).
Trigger: The player needs to take a shortcut.

Scenario:

1. The player moves the character to the cell containing the wall/barrier by using one of the 4 buttons up/down/left/right.
2. The player stays in the same cell (not moving).
3. The playtime always runs.

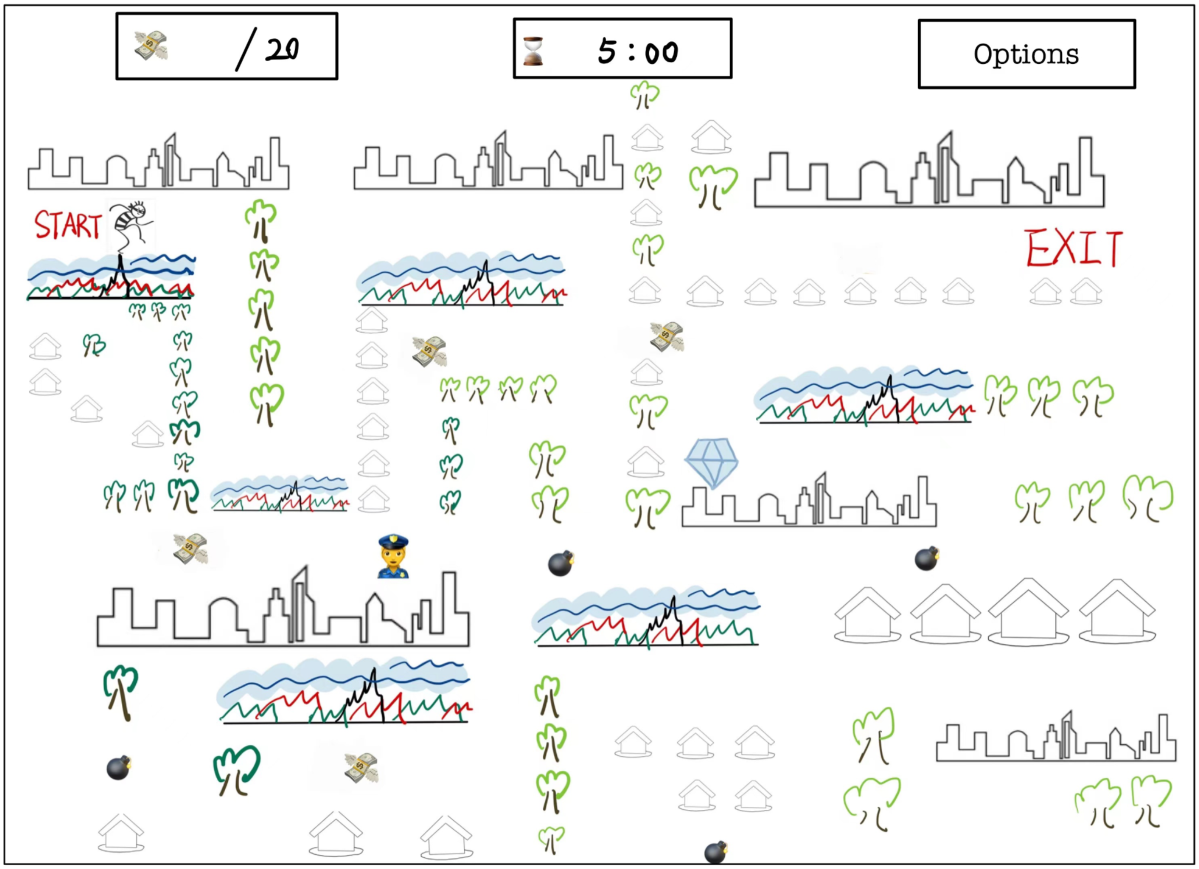
Exceptions:

1. The player touches a reward - see use-case: Claiming rewards.
2. The player touches a bonus - see use-case: Claiming bonuses.
3. The player touches a moving enemy - see use-case: Attacked by a moving enemy.
4. The player touches a punishment - see use-case: Attacked by a punishment.

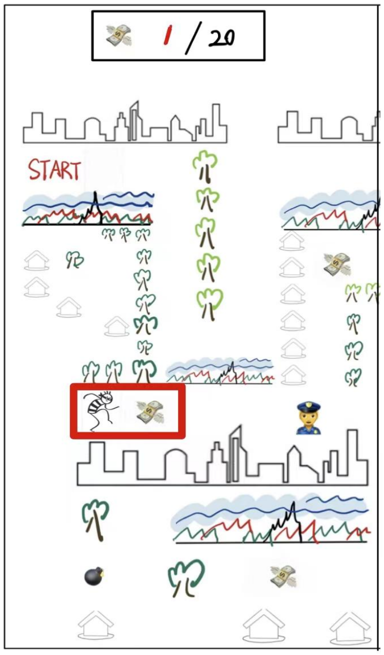
Priority: High priority - to be implemented during basic functions.
Frequency of use: Frequent.

Part 3: User interface

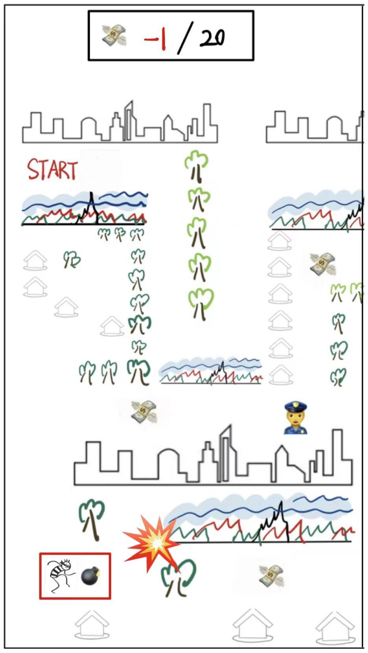
- The overall user interface



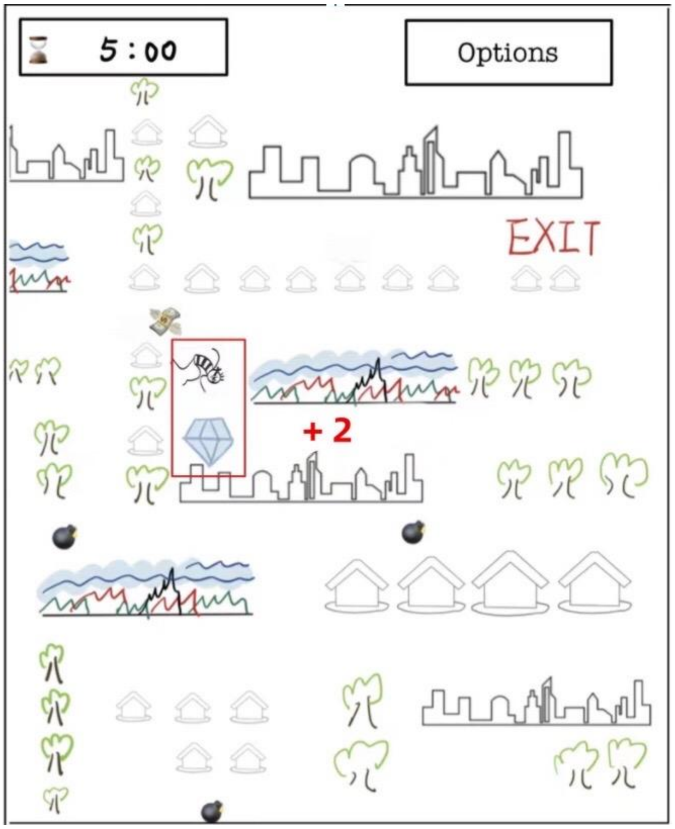
- When the thief touches the money bill, (bomb), one point is added to the score.



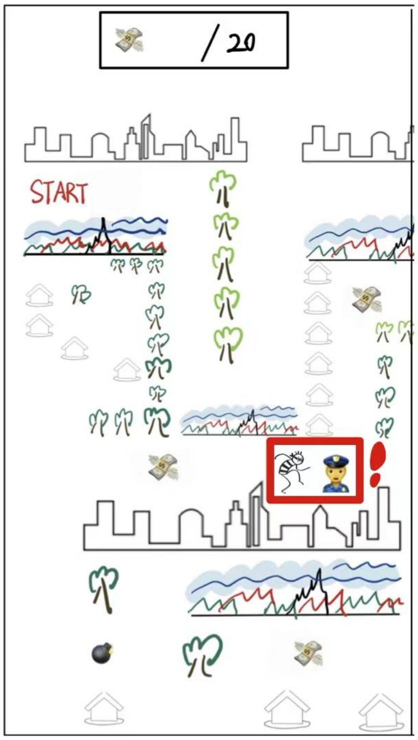
- When the thief touches the punishment the user will lose one point.

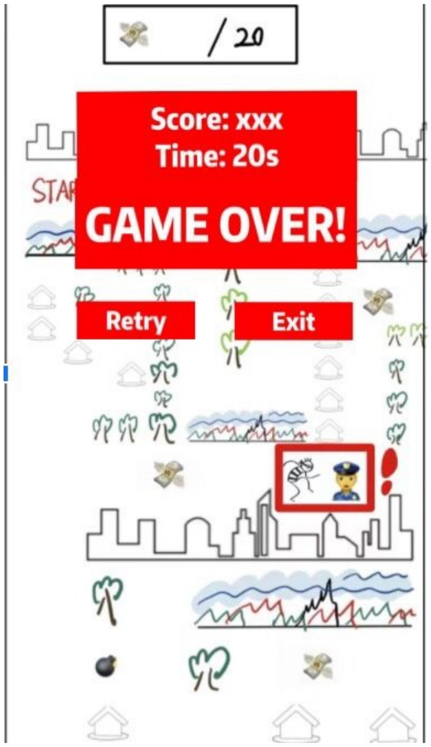


- When the thief gets the diamond (bonus), two points will be added to the score.

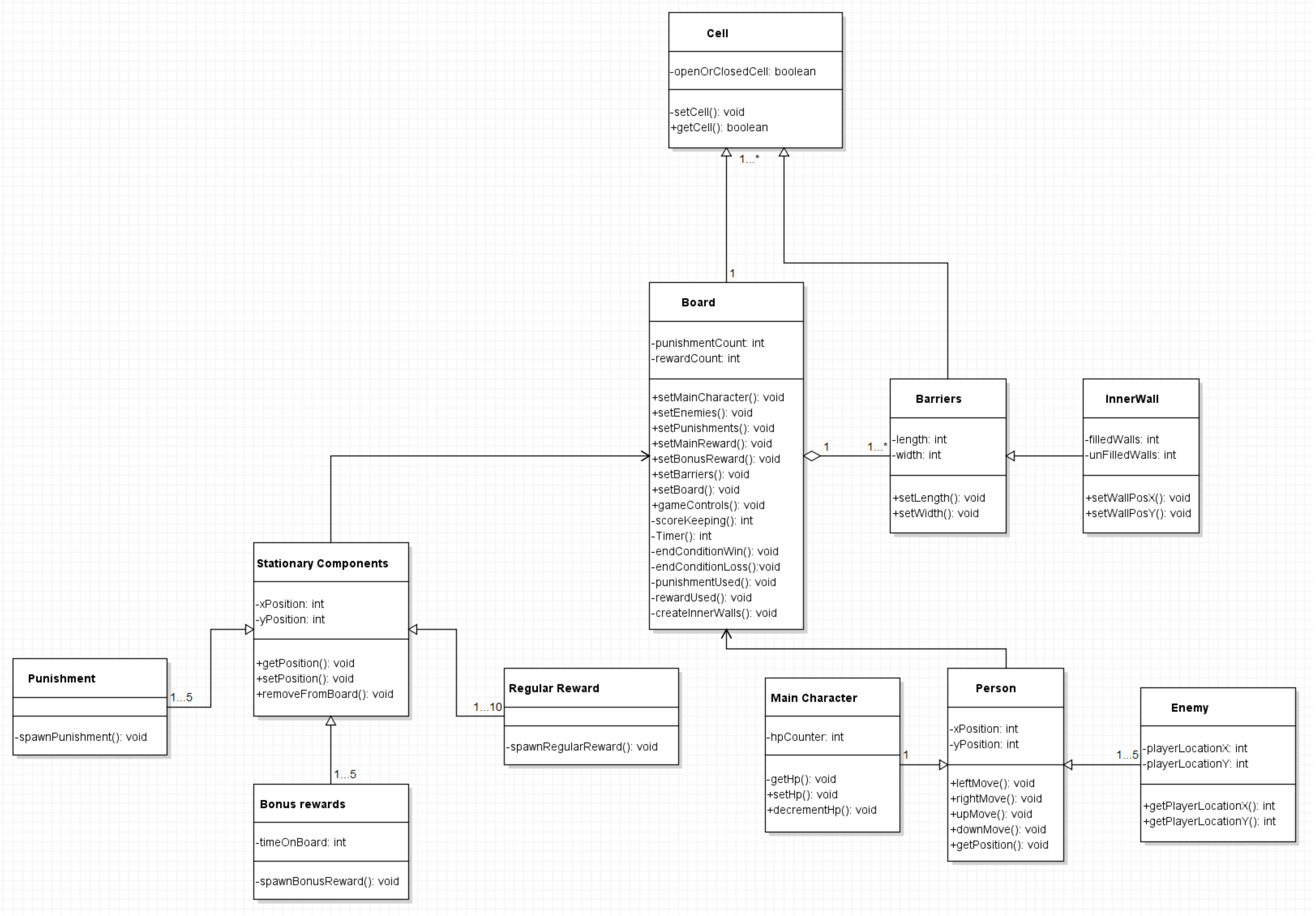


- When the thief meets the police, the player will lose the game and the screen displays “GAME OVER!”. Player can press the “Retry” button or exit the game.





Part 4: UML Diagram



Note: Parts were divided equally between team members. At the end, we combined everything into one PDF and submitted everything as one.