Plan and Game Description

Objective:
- Save as many hostages as possible and find the exit within the time limit.
- Score will be determined by the number of hostages and remaining time upon reaching exit.

Player:
- Move in 4 directions
- 1 life

Rewards:
Achievement messages will be displayed when the player gets reward.
- Regular:
    - Placed around the map behind locked doors.
    - Follows behind the player (stackable).
    - Grants a unique ability to the player when following the player around.
    - Increases player's hit box.
- Bonus:
    - Harder to obtain.
    - The player will have to go down a different path with more traps and enemies.
    - Optional, but will significant increase in the player's score.
    - Levels with bonus rewards will appear after the player obtains a certain amount of regular hostages.
- Keys:
    - Must be collected in order for the player to open a door to key rooms.

Enemies:
More enemies are released as remaining time decreases.
- Non-moving enemies:
    - Stationary traps
- Moving enemies:
    - Horizontal/vertical movement type
    - Lock on movement type (within certain range)
    - Projectile-type enemies that fire projectiles
    - Take hostage and use its skill type
- Punishment:
    - Takes time away from clock
    - Slow player movements
    - Hostage(s) return to locked door

Barriers:
- Walls:
    - Stationary & non-passable
    - Border along edge of the map
    - Structure for map layout
- Doors:
    - Stationary
    - Passable with condition met
    - non-passable for enemies.

Board:
- Top-down view, 2D stylized board.
- Start point is on the left of the map
- End point will be near the right of the map and triggers when a minimum number of hostages are obtained.
- Time remaining counter & hostage counter
- Player abilities and cooldowns

Schedule:
Our group will be applying the agile-unified process model, and we will be focusing on a continuous cycle of modelling, implementation, testing, deployment, configuration, and environmental management.

1. Complete version of UML diagram
2. Implement UML into code
3. Test and fix bugs/errors
4. Add additional functionality
5. Add to UML diagram
6. Repeat steps 3,4,5 until satisfactory