# Software Requirements and Design Document

# For

# Group <3>

Version 1.0

**Authors**:
Grace B
Elias D
Jason G
Miguel M
Santiago R

## 1. Overview (5 points)

The purpose of the project is to develop a first person parkour game called Owl's Omen. The game revolves around the plot of the main character, Nim, who is an acrobat in a bankrupt circus. Nim is approached by a pair of wealthy siblings about smuggling goods into the poorer area of the city to help out those in need. As the game progresses, these smuggling missions get riskier and more challenging. The system is designed around this plot and incorporates features to mimic acrobatic movements. The movement system allows for traditional WASD movement, with jumps. It also allows for launching off of any surface and pulling towards any surface (to/from a specific pointed-to location in the surface), as well as "recovery jumps" that can be done mid-air to regain push/pull charges at the cost of mid-air directional adjustments for some time.

There are surfaces that affect movement by: increasing player speed (unidirectional and multidirectional), slowing down player speed (multidirectional), and providing the player with a jump boost given that they are already jumping. The game also includes a rudimentary dialogue system, which allows the main character to interact with NPCs. This game also includes a functioning enemy AI that can track and follow the main character. This enemy AI can not only track the main player, but it can also patrol points spread around the map and also has a character design fitting the world and some animation. The game also features a HUD with various UI elements that include (but are not limited to) a boost cooldown meter, push/pull charge capacity, distance-to-goal indicator, and a functional minimap.

The game also includes a Main Menu that will load the game (any scene specified), open settings(same scene, new panel), select a level, and quit the game (ends application). The settings menu allows users to change technical options like the volume, resolution, graphics, and toggle fullscreen. There is also a Pause Menu that pauses the game and allows the user to resume, return to the menu, save and load, or quit the game. These menus play music and have sound effects when clicking buttons. When transitioning from them there is a loading screen that fades from black.

## 2. Functional Requirements (10 points)

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just* *<u>what</u>*). *You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. Player movement system (Priority: High) (in the following, 'x' is used to mean an arbitrary value exposed in the inspector to be tuned when balancing the game, each time x is mention it is a different arbitrary value)
    a. Slow down game speed to 0.5 while SHIFT key is held.
    b. Movement abilities while on ground:
        i. Move forwards/backwards/left/right at speed x based on first-person view on using W/S/A/D keys
        ii. Press space to jump into the air by x amount

        iii. Click on a point at a surface that is no more than x meters away from the player to receive a push in the opposite direction they are facing with force x or towards where they are facing with force x.
            1. Limited by x amount of charges, which are shared between pushes/pulls
            2. Controls: Push down LMB to aim a push off a surface, release to execute the push. Push down RMB to aim a pull onto a surface, release to execute the pull. If the player presses RMB while LMB is pressed down, the force is changed to match RMB, and vice-versa.
        iv. When the player lands, their charges are reset.
    c. Movement abilities while on air
        i. Same as ground i, but wasd movement has slight smoothing applied such that it is easier to land on smaller platforms
        ii. Pressing space instead triggers a “leap” angled in the current movement direction, until the player touches the ground or uses another boost, their wasd movement has a lower effect than normal, such that they lose a significant amount of control and must instead plan before leaping.
            1. Doing this has the benefit of gaining some distance, but primarily it resets the charges the player has
            2. This is on a cooldown
        iii. Same as on-ground.
    d. Ground momentum conservation
        i. Some airspeed (if not all) will be retained by the player upon landing on the ground given that they do not take any hard turns (anything higher than x degrees over y time) or stop moving (holding down whichever wasd key corresponds to their current momentum vector)

2. First person arm animation: (Priority: Medium)
    a. The staff model that the player holds sways with acceleration to give feedback on current speed to the player
    b. The staff model also angles itself when looking down or up by an amount equal to the degree the player is looking down/up by
    c. The staff model should spin to point towards the current surface whenever the player is aiming
    d. Staff is a diegetic UI, meaning that the information normally displayed in the UI overlay is instead displayed “in-world” as parts of the staff.
        i. Charge counter is at the tip of it, lunge meter is a charging segment of it
    e. Headbob
        i. Gets more intense the faster the player is moving on the ground
3. General VFX: (Priority: Low)
    a. Boosts to/from places, colorful explosions
    b. Leaps, some air effect around the screen to show the speed
    c. High speed, a general-purpose way to indicate the user is moving over x speed, could just be increased screenshake.

4. Enemy: (Priority: High)
    a. Enemy attack animation to signal enemy attacking the player
    b. Enemy now goes back to patrolling when the player gets too far away
    c. Enemy now deals a damage decay, where the player will constantly lose health based on the distance of the enemy every X seconds.

d. The closer the player gets to the enemy, the more damage the enemy will deal to the player. The farther away the enemy is the less damage the player takes every X seconds.

5. Damage Animation: (Priority: Low)
   a. Red color overlay shown when player takes damage
   b. Red overlay quickly fades as player takes damage
   c. Red overlay stays when the player gets below X amount of health.

6. Surfaces Affecting Player Movement (Priority: High)
   a. Current features that are already implemented
      i. Speed boost (unidirectional and multidirectional)
      ii. Jump boost (applied to players that are already jumping and a separate jump boost for players that are not already jumping)
      iii. Slow down (unidirectional multidirectional)
   b. Features that are currently being worked on
      i. Jump boost for players that are not already jumping

7. Dialogue System (Priority: Low)
   a. Current features that are already implemented
      i. Canvas that displays an arbitrary amount of dialogue
      ii. Player can speed through dialogue
      iii. Player can skip to next chunk of dialogue
      iv. Player can respond to dialogue via button mechanism
      v. Typewriter effect when displaying text
      vi. Pausing for a specific amount of time
         1. The amount of time paused depends on the punctuation
   b. Features that are currently being worked on
      i. Fixing the button bug where the button shows up too early
      ii. Improving the UI of the canvas
         1. Making the aesthetics of the dialogue match the gameplay more
8. General Audio: (Priority: High)
   a. Current features that are already implemented
      i. Menu Sound effects when pressing buttons
      ii. Menu Music plays
   b. Features that are currently being worked on
      i. In-game audio and sound effects
      ii. Music during gameplay

9. Save/Load: (Priority: High)
   a. Current features that are already implemented
      i. Basic information like player health, position, and level is saved and loaded
   b. Features that are currently being worked on
      i. Implement this data into in-game effects like loading levels and the position of a player in that level.
10. Scene Transitions: (Priority: Low)
    a. Current features that are already implemented
       i. Simple Loading animation is played in between scenes
11. HUD Elements: (Priority: Low)
    a. Current features that are already implemented

        i. Functional minimap that follows player movement + rotation, renders simple map layout for terrain objects, and renders game objects as icons.
        ii. Waypoint marker that indicates distance from an object and clamps to the sides of the screen. Now works with multiple waypoints.
        iii. Objective list that indicates number of objectives interacted with.
    b. Features that are currently being worked on
        i. Improving UI elements as needed.

12. Teleportation (Priority: Medium)
    a. Transport a player from one location (the teleport) to a predefined target location (the teleportation target).
    b. Requires a character controller for the player, gameobject for the teleport, and gameobject for the teleportation target.
    c. On collision (when the player enters the teleport), the player's location is changed to that of the teleportation target.
    d. This feature is completely implemented.
13. Level Selector (Priority: Medium)
    a. Select any level from a menu on the main menu.

## 3. Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

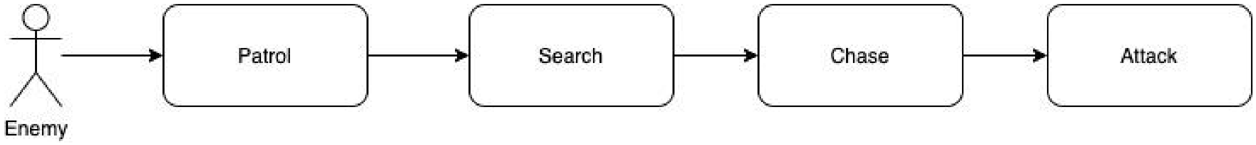
Runs at a minimum of 60fps on any normal gaming rig.

Inputs are responsive in that any input will cancel any current action that would conflict with it (for example, clicking somewhere to boost off a surface should not lock player into a "launching off the ground" animation that is impossible to cancel, any other boost should cancel the animation of the current one.) In other words, inputs over animations / visuals--ideally both in harmony (VFX that "feel right" even with twitchy inputs).

Surfaces that affect motion "feel" natural to the player. One of the most important parts of implementing these features is making sure that the boosts, jumps, or slow downs do not feel jarring or unnatural to the player. To test this, I went through each feature and experimented around with the numbers until I found a setting that "felt" right. A lot of this is pure experimentation and to a certain degree, is subjective.

## 4. Use Case Diagram (10 points)

*This section presents the **use case diagram** and the **textual descriptions** of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*
***Textual descriptions of use cases**: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*



Use Case descriptions

- Patrol: The enemy will be patrolling from different set points in the map
- Search: The enemy will be scanning each object in view to see if it's a player
- Chase Player: Once the player is seen within the enemy's vision the enemy will chase the player
- Attack Player: Once the enemy is within a certain distance of the player the enemy will attack using an attack animation.

## 5. Class Diagram and/or Sequence Diagrams (15 points)

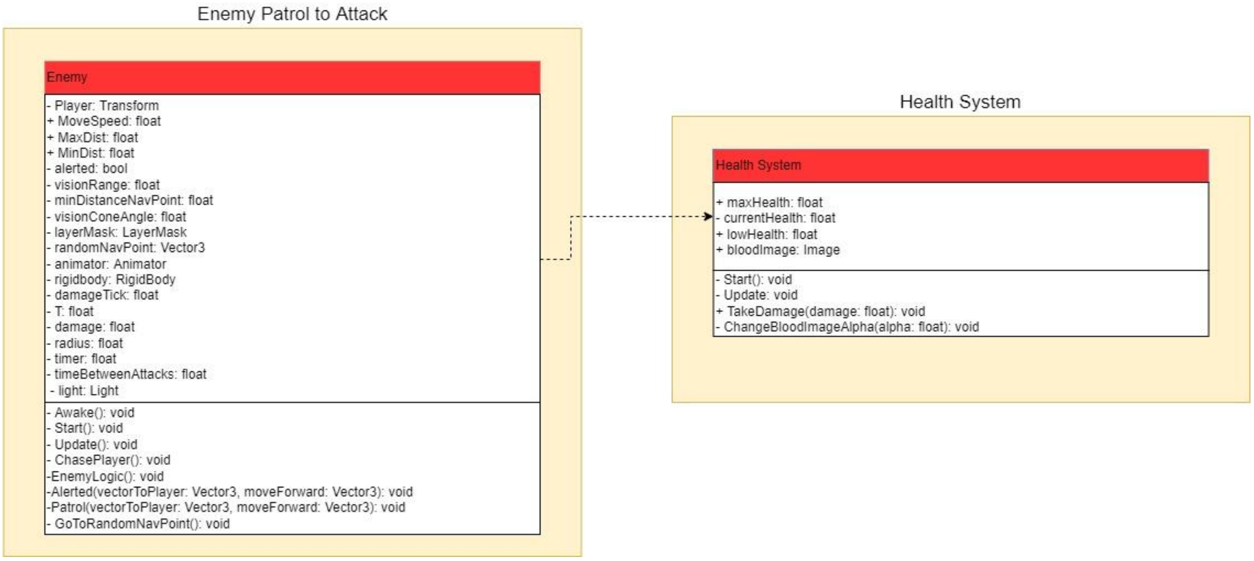
*This section presents a high-level overview of the anticipated system architecture using a **class diagram** and/or **sequence diagrams**.*

*If the main **paradigm** used in your project is **Object Oriented** (i.e., you have classes or something that acts similar to classes in your system), then draw the **Class Diagram of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main **paradigm** in your system is **not** **Object Oriented** (i.e., you **do not** have classes or anything similar to classes in your system) then only draw **Sequence Diagrams**, **but for all the use cases of your system.** In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams** show the **fundamental objects/classes** that must be modeled with the system to satisfy its requirements and **the relationships** between them. Each class rectangle on the diagram **must also include the attributes and the methods of the class** (they can be refined between increments). All the **relationships between classes and their multiplicity** must be shown on the class diagram.*

*A **Sequence Diagram** simply depicts **interaction between objects** (or **functions -** in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*



## 6. Operating Environment (5 points)

The software is built on Unity 2021.1.1f1 to operate in Windows (primarily), MacOS, and Linux systems.

## 7. Assumptions and Dependencies (5 points)

TextMeshPro
Other Unity Store packs (primarily for visual assets and FX).
Audio (Jalandhar by Kevin MacLeod | Download and Stream on Chosic, hammer hit anvil blacksmith 01 sound effect | | sound effects free download (freesoundeffect.net), Desert Caravan by Aaron Kenny | Download and Stream on Chosic)
Image (desert, City Wallpapers HD / Desktop and Mobile Backgrounds (wallup.net))