**Covert Protocol**

**Project Report**

Yash Ahuja

Ryan Cram

Neil Hoggatt

Peter Trang

**Executive Summary**

Covert Protocol is a real-time, squad-based, tactical, side-scroller in which the player will take control of group of miscreants and outsiders known as the Members of Blacklist who, with the help of a mysterious Informant, seek to overthrow a tyrannical government.

In order to successfully play the game the player will utilize the various abilities of the squad members. With the Hacker, the player will be able hack mechanical enemies, and thus gain strategic advantages in combat, or hack parts of the environment (including doors and traps) to bypass an encounter altogether. With the Medic, the player will be able to heal their squad members, mitigating damage and preventing squad members from dying, making the combat more manageable. With the Scout, the player will be able to sneak past enemies to activate traps and disable defenses, giving an alternative to charging through a level guns blazing. With the Soldier, the player will be able to force their way forward by utilizing the soldier’s high damage output coupled with his ability to shield his allies/himself, increasing everyones survivability. The greatest challenge will not be using the abilities of a single squad member, but rather it will be combining the abilities of each squad member to overcome obstacles and achieve the victory condition of a given level.

The game is broken down into multiple levels. Each level consists of a building and/or series of rooms, objectives (the goal of the level), enemies/trap/obstacles (which impede level completion), an aesthetic (the visuals that help define the environment of the level), and the area exit(s). The area exit is where the player will go after they complete the final objective of a level. Reaching the exit can be an objective in itself. Some areas have multiple exit points, and it is up to the player to decide on the route to take. Each route has a unique outcome and a unique set of challenges/objectives.The standard gameplay is further enhanced by the HUD and other interface elements.

Additionally, the concept of quantum key distribution is introduced to the player, indirectly, via the BB84 protocol. The concept presents itself through a mini-game/event that occurs when the player hacks into objects critical for level completion.

**Introduction**

Initially, the project was intended to be an educational game which focused on delivering the concept of Quantum Key Distribution directly to the player. The project would, as an additional requirement, be a submission into the Microsoft Imagine Cup competition. The idea was later adjusted to be a stealth focused game where the player would guide a squad through puzzle-like levels, and the levels required the player to hack the environment (which would open/close doors, reroute dangers, unlock a safe, etc). The player would be a skilled hacker, and would be the one handling all of the important hacking scenarios.The setting of the game was futuristic, so the hacking would be represented by a puzzle related to the concept of Quantum Key Distribution.

The project had several requirements.

1. Implement Quantum Key Distribution in such a fashion that the player should learn how it basically functions.
2. Implement at least one level, which showcased major mechanics.
3. Develop an interesting setting/story/environment to fit the proposed game concept.
4. Compete in all/any Imagine Cup competition and side-competitions.

**User Guide**

**Controls**

***Main Menu***

When in the main menu of the game (***page 7***), the user maneuvers it by using the “up”, “down”, and “return”/”enter” keys. The “up” and “down” keys will move between the options found in the main menu. Pressing “enter” will select the current option of the menu.

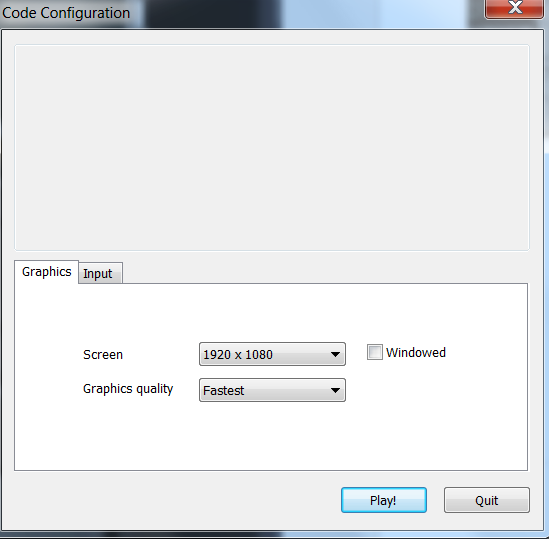
***In-Game***

The mouse will be the primary means of input. There are a few keyboard shortcuts.

* Mouse
  + Left Click:
    - Normal Click: If an object is selectable, left clicking it will highlight that object, Your squad members will always be selectable. If a unit is selected, and its under the player’s control, the player, it can be issued commands. Will unselect all previous objects if nothing or something unselectable is click on.
    - Hold: Will create a drag box. When the left mouse button is released, it will highlight and select all selectable objects inside of its area. Those objects can be issued commands, just like an individual object.
  + Right Click
    - Normal: Commands all selected units to use their default interaction with the targeted location as the argument. Normally, this commands the selected units to move to the clicked location. If there is an interactable target at the location, it is possible that a special interaction can be triggered. The most notable example of this is “fire”/”attack”, which makes the selected units attack the target, instead of using their default interactions.
    - Hold: Works as normal click, except the default interaction isn’t used. Instead, all the possible interactions possible at the mouse location are display in a menu (shown below). The menu takes into account the location, and whether or not an interactable object was at the location. Some standard interactions include “Examine”, “move”, squad abilities, etc.
  + Scroll wheel: Zooms in and out of the the game view. Increases the amount of the map that is visible, or makes it possible to focus on a particular area.
* Keyboard
  + ‘1’ Squad (Soldier) Shield ability shortcut
  + ‘2’ Squad (Medic) Heal ability shortcut
  + ‘3’ Squad (Scout) Stealth ability shortcut
  + “Space bar” Locks/Unlocks the camera. If the camera is locked, it will not move from its current location. When the camera is unlocked, it will be possible to move the current focus of the camera.

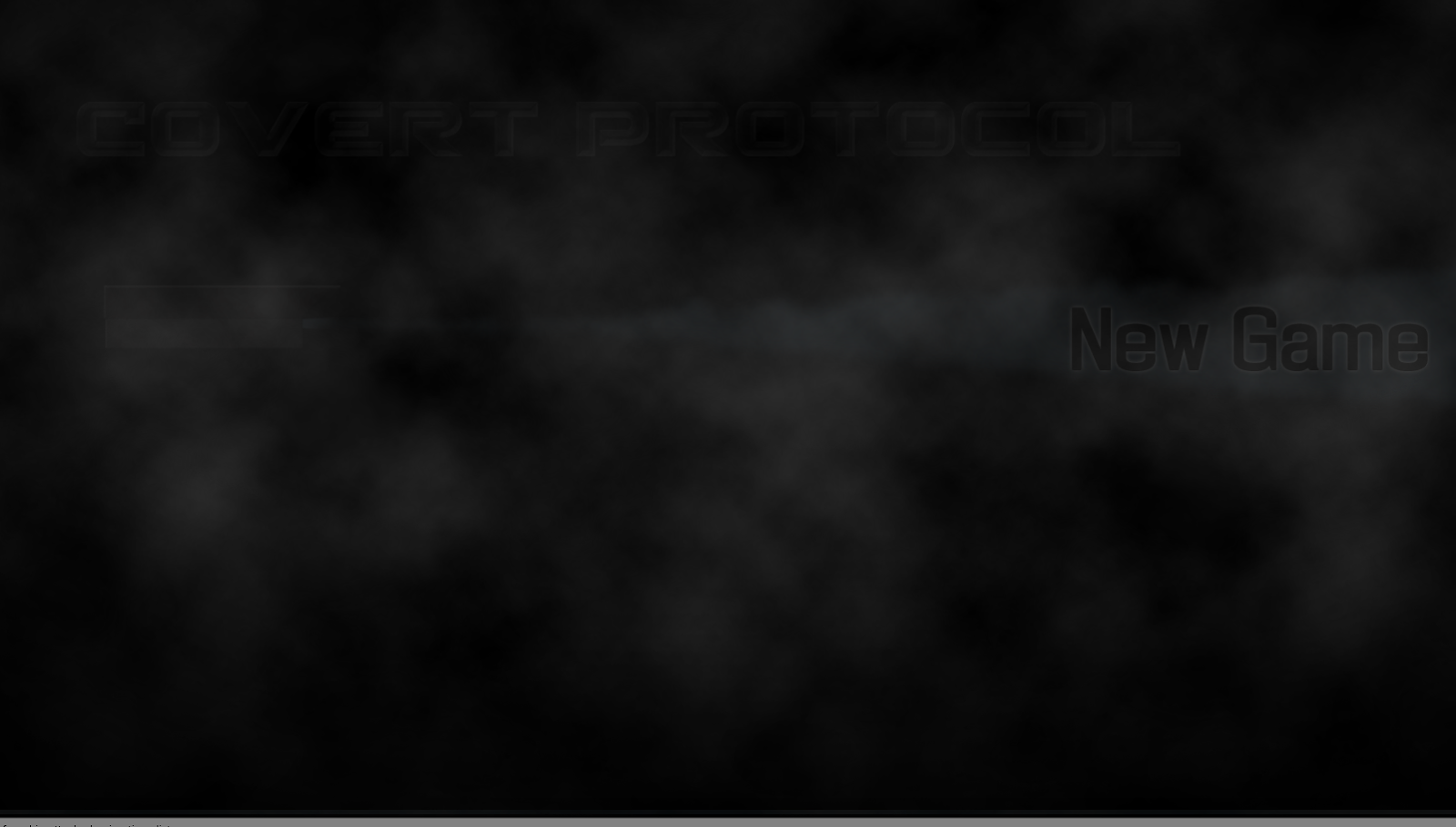
**Starting the Game**

When the user starts the game, they will first see this launcher open. It opens with all Unity created projects by default. The user just needs to select “Play”, and the actual game will launch.



***Main Menu***

Below is the Main Menu. It has several options, including: “New Game”, “Minigame”, and “Exit”.

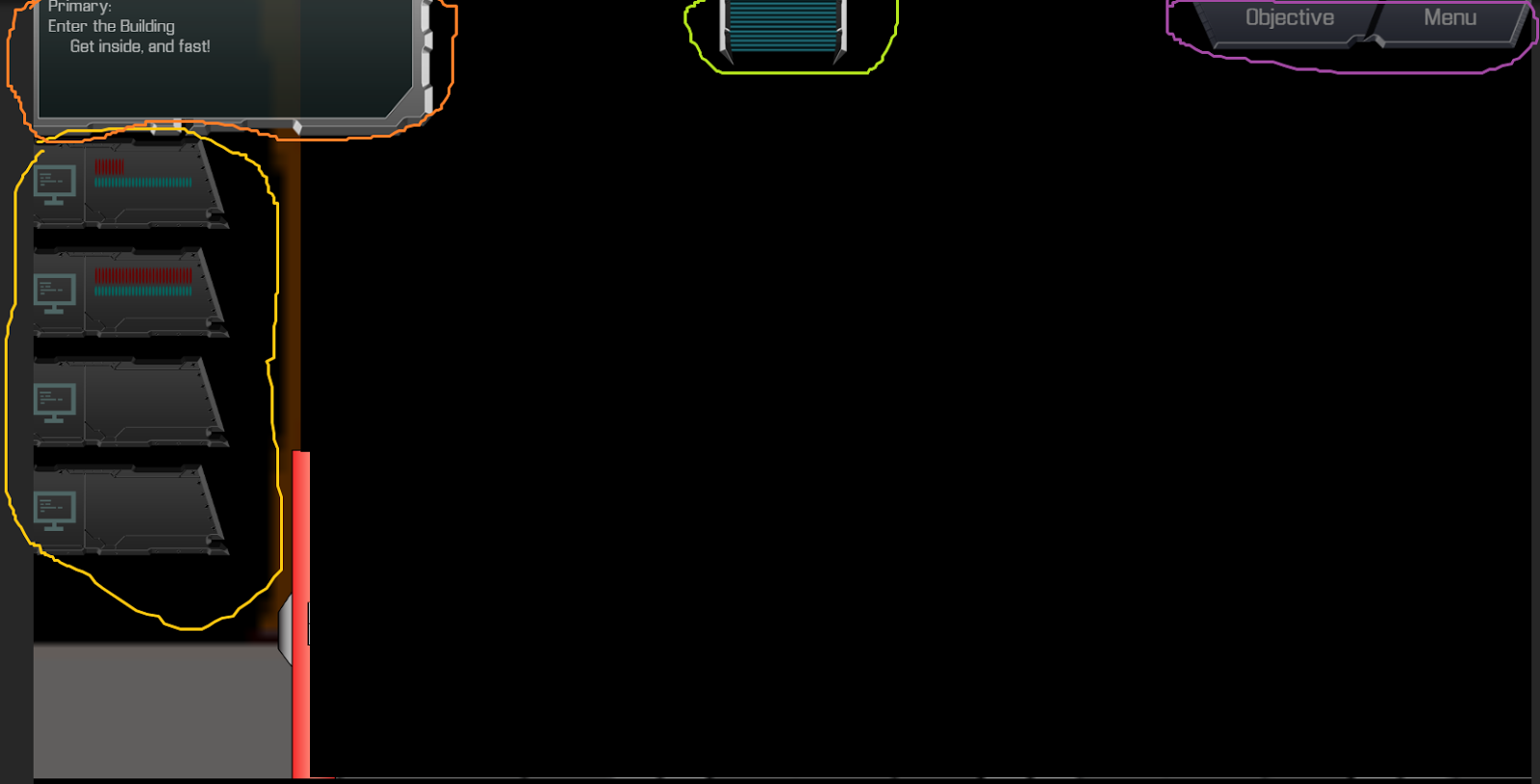


The New Game option is displayed if the user has not played before, and Continue if the use has played before. For presentation purposes, this will only be a “New Game” option. The “Minigame” option will open an instance of the minigame so that it may be played without going through the game.The final option in the menu is “Exit”, and it will terminate the game.

To start the main part of the game, the player will need to select “New Game”. The game will load the first level of the game.

***In-Game UI***

When playing through a level, there will be several interfaces that the player will need to acquaint themselves with, and these are the HUD, Interaction Menu (Radial Menu), Objective Menu, and the Pause Menu.



**HUD**

The HUD consists of several sections: the Squad Member information bars, the Simplified Objective Display, the Alert status, and the Objective/Menu buttons.The squad information bars (yellow section) give the player convenient information regarding the squad member health, energy (system which regulates ability usage), and shield (temporary health provided by the soldier). The Simplified Objective Display (gold/orange section) gives a simple overview pertaining to the current objectives. The lime green section changes color depending on the enemies awareness of the player, giving the player a heads-up to help them safety of the squad. The pink section contains a menu and objective button.

The objective button open a more detailed overview related to the player’s current objectives, and the menu button opens a pause menu which allows the player to 1) Restart the level, 2) return to the Main Menu, 3) Exit the game, and 4) open an Options Menu (to adjust in-game settings). Below is the Pause Menu.



*I*

**Interaction Menu (Radial Menu)**

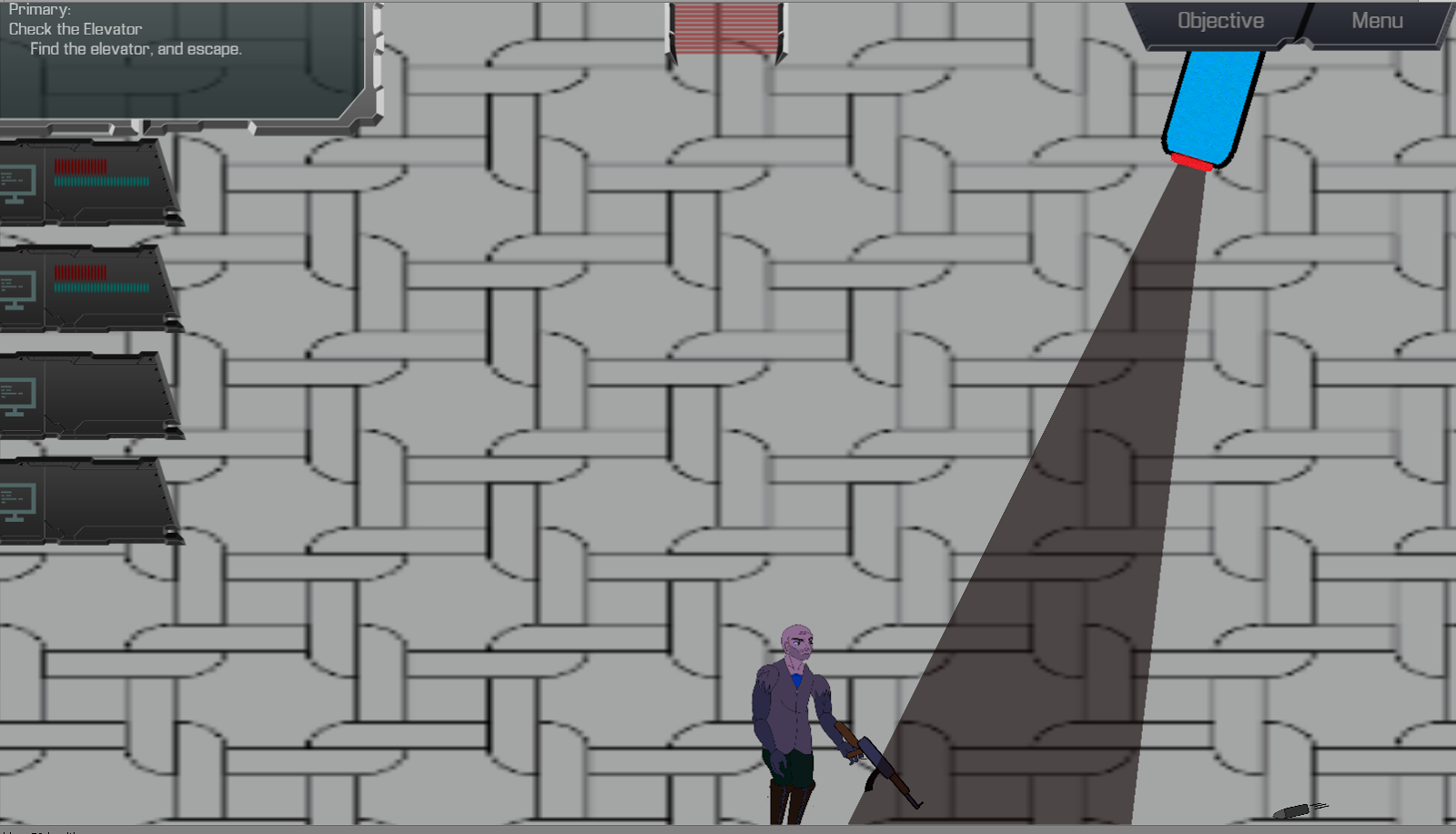
An important aspect of the game is interactions Interactions are essentially the things you squad members can do to each other (i.e. Heal), to their enemies (i.e. Attack), and the environment (i.e. Hack or Open). Whenever the user right clicks, they will trigger either a default interaction or the interaction menu to open. The default interaction is what the squad member will always perform when the player does a short right click. It differs depending on which unit is selected, and it is also depends on what the target is. For example, normally, a squad member will move to the clicked location when a default interaction is triggered. However, if an enemy happens to be the target, then the squad member will instead attack. If the player wishes to not use the default interaction, and instead decide use another interaction that is possible given the clicked location, and the selected units, then the player can open the Interaction Menu. The interaction menu is can be triggered by holding down right click until it opens.



All of the options in the menu, displayed above, are the possible interactions that can be performed given where the player clicked and the units currently selected. Each option has an icon associated with it, and when a icon is hovered over with the mouse, text associated with the icon is displayed in the center of the radial menu.

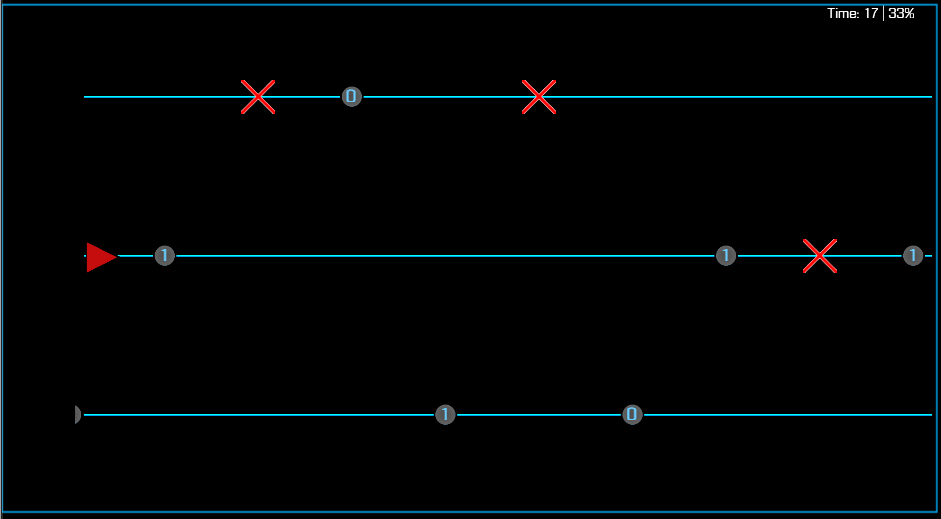
**Alert**

The alert area of the HUD changes its color depending on whether or not the enemy is aware of the player or not. In the previous images, the enemy was not aware. In the following image, the alert changes to red, giving the player a clear visual indicator.



***Minigame*** (Bitrunner):

*How to Play*



The goal of the minigame is to collect a certain percentage of the bits spawned (the circular objects with 1’s and 0’s) while avoiding the obstacles (the red ‘X’s). The default amount of bits that need to be collected is 85%.

Before playing, it is worthing noting the values in the upper right corner:



The first one tells you how much time is remaining, and the second value tells you the percentage of bits that you managed to collect so far.

*Controls*

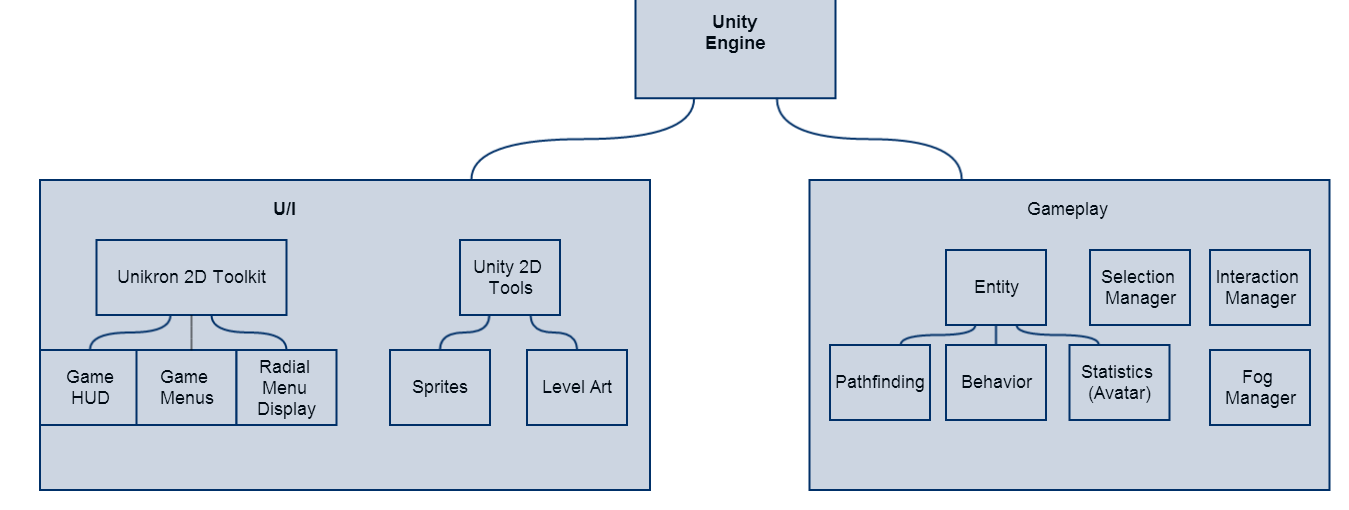
When playing the minigame, the primary controls is the Up arrow key and the Down arrow key. They are used to move the player (the red triangle below) between the lanes.

The mouse will be used when the minigame is not active. The user can left click the “Start Game” button in the main menu to start the game (below):

or left click the screen after a minigame is complete to exit the screen (below):



**Architecture and Design**



Unity is a component-based game engine. Objects in the game are referred to as Game Objects, and each of Game Object can have multiple components. A component helps define the behavior for a Game Object. Some components are built-in components, and the rest are user-defined scripts.

**Conclusion**

The team had three primary issues throughout the project, and was not able to complete one of the core requirements. The first was the competition process of the Imagine Cup (mentioned above), the second was merging with Unity, and the final one was the level creation process. The core requirement the team was “Compete in all/any Imagine Cup competition and side-competitions”, and it is related to the first issue.

The first issue related was the Imagine Cup competition process. The team did compete in every side competition, following our requirement. However, the competition information for the main competition was inconsistent. The inconsistent information created overhead and it was difficult for us to adequately compete. Eventually, we devoted our full focus to the development of the game.

The second issue was merging Unity scenes and scripts. A scene, in Unity, is the a level with all the game objects (walls, squad members, pathing graph, enemies HUD, etc) positioned in the Unity world space. The problem the team had with merging was merging the scene files. There was no tool that allowed the group to merge the scenes visually. The only approach was manually merging the scene files, which was time consuming. Due to the issues caused by the scene merging, the team had to limit who worked on a scene a given time. This solved the scene merging issue. Scene merging was not the only merging issue. The C# scripts used to implement the game features would occasionally be reverted, ignored, deleted, and/or re added due to an occasional git/Unity related issues. While the problems were always correctable, the overhead required to correct these rare problems is worthy of note.

The third issue was bottleneck found within the level creation process. In order for a scene to be completed, it had to pass through a series of modifications.’only one member of the group could modify that scene at a time. Sometimes, the script issue also occurred, drawing out the process further. Unlike the previous issues, this one could not be solved.

Overall, the team managed to deliver on the remaining three core requirements. The team was able to properly integrate the Quantum Key Distribution concept into the game through an indirect fashion. The approach taken by the team was the introduction the concept through certain hacking events that occurred on each level. It made the idea both engaging and connected to the main game. The team was able to implement two levels, not just one. Each level also is unique in its objective, challenge, and outcome. The team was able to present an interesting setting/environment that suited the game concept. The idea was explained in the executive summary.