

# **Rogue AI!**

**Final Year Project – Mid Report**

**Session 2020-2021**

A project submitted in partial fulfilment of the  
COMSATS University Degree  
of  
BS in Computer Science / Software Engineering (CUI)



Department of Computer Science  
COMSATS University Islamabad, Lahore Campus

20 May 2021

## Evaluation

Project Group Members (To be filled by students)				(To be filled by supervisor)
Sr.#	Reg. #	Student Name	*Signature	Obtained Marks (Total Marks: 10)
(i)	FA17-BSE-011	Muhammad Humza Butt		
(ii)	FA17-BSE-071	Maha Muzammil Ejaz		
(iii)	FA17-BSE-038	Sheikh Umer Saeed		

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others.

Supervisor Name: ASMARA SAFDAR

Supervisor Signature: \_\_\_\_\_

Remarks (if any): \_\_\_\_\_

---

---

## **Abstract**

Rogue AI! will be an FPS puzzle game in which players are required to solve different logical puzzle to move forward and defeat robotic enemies. It is designed to test player's intelligence and logic building skills and solving hidden puzzles all while fighting robotic enemies in a challenging environment.

# Contents

Rogue AI!.....	1
1    Introduction.....	6
2    Goals and Objectives.....	6
3    Problem Statement .....	6
4    Assumptions and Constraints.....	7
4.1    Assumptions .....	7
4.1.1    Product Delivery .....	7
4.2    Constraints .....	7
4.2.1    Performance .....	7
5    Motivation and Scope .....	7
6    Game Scope .....	7
7    Requirement Analysis.....	8
7.1    Literature Review .....	8
7.2    Stakeholders (Actors) .....	<b>Error! Bookmark not defined.</b>
7.3    Game / Software Development Life Cycle(SDLC).....	8
7.4    Requirement Elicitation.....	12
7.4.1    Functional Requirements .....	12
7.4.2    Non-Functional Requirements .....	15
7.5    Use Case Description.....	16
7.5.1    Launch Game .....	16
7.5.2    Playable Character .....	17
7.6    Use Case Diagram .....	18
7.6.1    Launch Game Use Case Diagram .....	18
7.6.2    Playable Character Use Case Diagram .....	19
8    System Design.....	20
8.1    Work Breakdown System (WBS).....	20

8.2	Launch Game Activity Diagram.....	21
8.3	In Game Activity Diagram .....	22
8.4	Game Update Activity Diagram .....	23
8.5	Sequence Diagram – Main Menu & In Game Activity .....	24
8.6	Sequence Diagram – Update Game.....	24
8.7	Gantt Chart .....	25
9	System Testing .....	25
9.1	Test Case: .....	26
9.1.1	Test Case Game Installation.....	26
9.1.2	Test Case Game Launching .....	26
9.1.3	Test Case Sounds .....	27
9.1.4	Test Case User Interface .....	27
9.1.5	Test case Performance.....	27
9.1.6	Test Case Puzzle Solving.....	28
9.1.7	Test Case Settings .....	28
9.1.8	Test Case Competition with the boss .....	29
9.2	Unity Engine Testing:.....	29
9.2.1	Game Installer Files .....	29
9.2.2	Models and their Animation.....	29
9.2.3	Player Game Object .....	29
10	Conclusion.....	29
10.1	Problem Faced and Lessons Learned .....	29
10.2	Project Summary .....	30
11	Future Work .....	30
12	References .....	30

# 1 Introduction

Rogue AI aims to be a warfare of human vs robots with the chain of puzzles to be solved to achieve the goal and a sci-fi story line. The game will help the player to make warfare strategies by intelligent decisions against the enemy and solving logical puzzles.

It will be a level-based game where difficulty increases after each level. At the end level, the player has to compete with a robotic army and boss to recover the chip that can cause end of human race.

Such games are played by vast majority of people in recent years regardless of age, gender or profession. Intense games like *Quantum Conundrum*, *Half-life 2* and *Portal* are a huge success.

## Character Types:

- |                         |   |
|-------------------------|---|
| 1) <b>Player</b>        | The player has to solve puzzles set by robots and fight with them in order to recover microchip.                                      |
| 2) <b>Enemy</b>         | Enemies are the AI robots who can follow and attack.  |
| 3) <b>Alien Animals</b> | These are alien life forms that can be dangerous or healthy. Killing the right one can provide player with health ups and vice versa. |
| 4) <b>Boss</b>          | It will be found on the last level. The player has to fight and kill boss to get the chip.  |

# 2 Goals and Objectives

- To improve the logical skills of the player
- A deployable product to be launched in the market
- To defeat the enemies and to recover the chip
- To make logical skills of the person a part of his survival instincts by making him/her undergo various challenges.

# 3 Problem Statement

In today's world, games have actually become the mode of only having fun, due to which a person wastes lots of time on playing such games. People tend to play games when they feel like they mentally have to relax themselves which basically becomes an addiction in future.

Moreover, in today's world, people tend to shift more towards casual and battle games, which basically isn't of any help to the user. Our game is designed in such a way that would be

beneficial to the player. Our game is basically a mix of fun and learning. The game is specifically designed in a way that a user will be able to build his logical sense by solving logical puzzle along with having fun.

## 4 Assumptions and Constraints

### 4.1 Assumptions

#### 4.1.1 Product Delivery

Product shall meet all deadlines and shall deliver what is intended to.

### 4.2 Constraints

#### 4.2.1 Performance

As desktop has boundless performance in terms of processing and graphics memory, Rogue AI! will be kept performance insensitive and will be able to run on majority sort of desktop devices.

## 5 Motivation and Scope

The core motivation for this game is to persuade the user to think logically in adrenaline conditions. It will put the user in a situation to make faster and logical decisions for his survival. In a nutshell, this game will serve the purpose of providing fun along with polishing the spatial skills of the user.

We are developing this game as our Final Year Project along with considering the industry demand to hit the game market. The specifications are mentioned in the goals and objective section.

## 6 Game Scope

According to J. Schell, in his work “The Art of Game Design: A book of Lenses” the scope of a game is defined by splitting into four categories: Mechanics, Story Line, Aesthetics and Technology.

Rogue AI! consists of moderate level of Mechanical needs as it is a puzzle solving game along with being a FPS game. The right answer to the logical puzzles shall decide whether or not the player shall be promoted to next phase or not. The Mechanical aspect is only required for all levels i.e., when the player has to fight and kill the robots.

In terms of storyline, such games need very minimal storyline which is enough to support the character’s look and personality. Rogue AI! has a mixed timeline as it has a Player, enemy, Alien Animals and Boss to play with as your character. The goal for each player is quite different in our game.

Aesthetics for the game matters as proper feedback is to be provided to the user with respect to visuals and sounds. Character and level design are a useful element in providing the engaging feel to the game and can contribute to the feel of the game. Rogue AI! however uses medium level of game aesthetics and has a moderately need of mechanical needs.

Rogue AI! will use 3D technique to visualize characters and environment and it will be a 3D game with side scrolling perspective of battle arena.

Therefore, Rogue AI! scores following in terms of game scope

- Mechanical Needs = Medium
- Story Needs = Medium
- Aesthetics = Medium
- Technology = Low

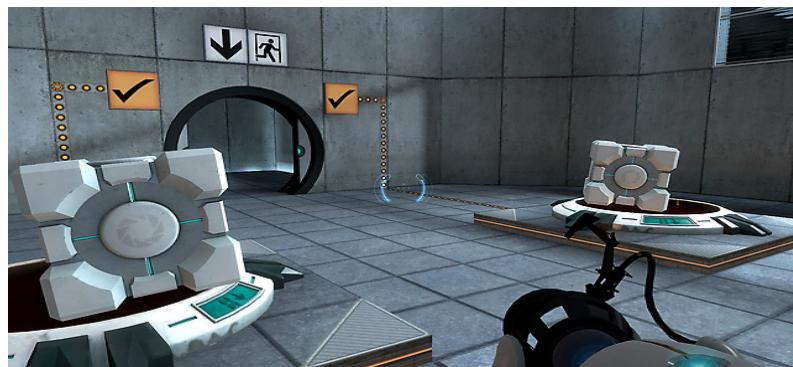
## 7 Requirement Analysis

### 7.1 Literature Review

Many fps puzzle games have been a huge success. An example is of a very famous game Portal.

#### 7.1.1 Portal

This is a 2007 game which was developed by Valve. It contains puzzles which the player has to solve by teleporting the character and then by simplifying the objects by using “Aperture Science Handled Device”, which is a device which creates intermediate parts in between two plane. Chell, who is a character has been challenged by AI known as GLaDOS (Genetic Life from and Disk Operating System) in order to complete the puzzle in Science Centre. This is done by using a gun, in return when the puzzles will be completed a cake is then offered. (1)



*Fig 1 – Portal*

### 7.1.2 Quantum Conundrum

Quantum Conundrum is a puzzle-platformer that has been viewed from a first-person perspective. As an opposing boy, the athlete can run and jump, participate in various modes, and raise light objects. A player may die by falling into a toxic liquid, endless holes, or falling from a very high altitude, and if he is attacked by harmful lasers; this will restart the player at the beginning of the puzzle. The purpose of each room is to reach its exit door, or it may be necessary to activate some switch or other devices before exiting. (2)



*Fig 2 – Quantum Conundrum*

### 7.1.3 Portal 2

This game is basically a puzzle game. The player is to be replaced by Chell in single player, as one of the two robots which are Atlas and P-Body either in a collaborative campaign or as an icon of human in community developed puzzles. The characters explore with nature. The characters suffer moderate damage but will eventually die if they suffer from injury again. No penalty is faced when falling on solid surface, but if he falls into endless holes, it will kill the character at that very moment. The motive of the campaign is to explore the Science Laboratory, which is basically a complex, flexible mechanized mechanics. Majority of the game takes place in modular test rooms with well-defined entrance and exit doors while a certain amount takes place behind the scenes where the purpose is less clear. (3)



*Fig 3 – Portal 2*

#### 7.1.4 Half Life 2

It is a single player FPS game in which players control Gordon Freeman. It is similar to half-life, especially the health systems, weapons and periodic physics puzzles. The new source engine and advanced graphics are an exception though. The game is initiated without any particular thing, but slowly and steadily the player builds up his arsenal during the game. Great effort is made in order to make exploration rewarding and enjoyable even though the game is linear in nature.

Detailed Simulation of Physics have been used as part of the new features. A few parts of the game involve driving cars. Environmental puzzles are also introduced with temporary mechanical devices which revolves around the players ability to pick, move and place objects. The solution to this is to include the physical properties of objects for instance shape, weight and strength. For example, “Route Kanal”, who is a player is required to place cinder blocks on a temporary ramp so that he could proceed to the wall. Player can also build stairs with blocks, so this proves that puzzle can be solved in different ways.

(4)



*Fig 4 – Half life 2*

## 7.2 Game / Software Development Life Cycle(SDLC)

For development basis of Rogue AI! I have selected the Agile Development model as there are no defined requirements for me to follow other than the baseline of the Game Design. It is natural for developers to experiment with the game and come up with the most workable output. Following are the main principles that have shown to be beneficial and viable for our development.

- Giving Less Importance to documentation and more on developing the actual product by giving high priority to quality
- We have to work on contemporaneous parts of the game in order to make them fit together.
- This model shall enable to create prototypes with primitive shapes and to test the feasibility of the game system
- In Agile, success isn't measured through milestones and deadline, but it is measure through product quality and stakeholder's satisfaction.
- As Agile is able to respond to changes, we were able to adapt to the frequently changing modules due experimentation of improvisation.

## 7.3 Requirement Elicitation

### 7.3.1 Functional Requirements

#### FR-01: Scripting requirement

Requirement No.	Description
<b>FR01-01</b>	The game must be implemented in C#
<b>FR01-02</b>	The game must be developed in Unity Game Engine

#### FR-02: Display, control and Audio

Requirement No.	Description
<b>FR02-01</b>	The game must be controlled with a keyboard and mouse
<b>FR02-02</b>	The game must be played on desktop with support for dynamic resolutions
<b>FR02-03</b>	The game must feature music and sound effects needed for gameplay

#### FR-03: Operating System

Requirement No.	Description
<b>FR03-01</b>	The game must be playable on Windows

#### FR-04: Levels and content

Requirement No.	Description
<b>FR04-01</b>	The game must provide hints to user for solving logical puzzles
<b>FR04-02</b>	The game must have a new map for each level

#### **FR-05: 3-Dimensional platformer**

Requirement No.	Description
<b>FR05-01</b>	The game must be viewed in a 1 <sup>st</sup> person perspective
<b>FR05-02</b>	The environment and characters shall be 3-dimensional in nature
<b>FR05-03</b>	Character's Health shall be visible within the game

#### **FR-06: Main Menu**

Requirement No.	Description
<b>FR06-01</b>	The game must provide options to view controls
<b>FR06-02</b>	The game must provide options to select game level

#### **FR-07: Single player**

Requirement No.	Description
<b>FR07-01</b>	Game shall be single player
<b>FR07-02</b>	Game shall be FPS

#### **FR-08: Logical Puzzle**

Requirement No.	Description
<b>FR08-01</b>	Solving logical puzzles must be sub objective of game
<b>FR08-02</b>	The game shall provide a few hints to solve the puzzle
<b>FR08-03</b>	Puzzle difficulty shall increase with each level
<b>FR08-04</b>	Logical points must be added w.r.t time taken to solve puzzle and complexity of the given puzzle

#### **FR-09: Hunt Animals**

Requirement No.	Description
<b>FR09-01</b>	Player must gain health after hunting energy providing alien animals
<b>FR09-02</b>	Player must loss health after hunting energy reducing alien animals

#### **FR-10: Fight Robots**

Requirement No.	Description
<b>FR09-01</b>	The player shall be able to shoot robotic enemies.
<b>R09-02</b>	Killing score must be increase if player kills robotic enemy.
<b>FR09-03</b>	Killing robotic enemies must be sub objective of game.

### **FR-11: Game Objectives**

Requirement No.	Description
<b>FR10-01</b>	All sub objectives must be completed to reach main objective
<b>FR10-02</b>	Objectives must be do-able and achievable

### **7.3.2 Non-Functional Requirements**

#### **NFR01: Performance**

Requirement No.	Description
<b>NFR01-01</b>	Normal game response time should be less than 1 second during gameplay
<b>NFR01-02</b>	The minimum frame rate should be at least 30 frames per second. Smoother gameplay will lead in higher frames per second

#### **NFR02: Usability**

Requirement No.	Description
<b>NFR02-01</b>	Average user should be able to get familiar with all the controls easily within 3 minutes
<b>NFR02-02</b>	Novice user should be able to get familiar with combat controls easily in less than 5 minutes

#### **NFR03: Maintainability**

Requirement No.	Description
<b>NFR03-01</b>	The source code of the game must be readable and maintainable

#### **NFR04: Platform**

Requirement No.	Description
<b>NFR04-01</b>	Game has to run on a window's-based platform

#### **NFR05: Resource Required**

Requirement No.	Description
<b>NFR05-01</b>	User must have a window's desktop with 4 GB of ram, 128 GB of storage and core i5 processor

#### **NFR06: Quality Graphics**

Requirement No.	Description
<b>NFR06-01</b>	The game must have good quality 3D graphics

#### **NFR07: User Interface**

Requirement No.	Description
<b>NFR06-01</b>	User Interface has to be clean and compatible with desktop screens

### **7.4 Use Case Description**

#### **7.4.1 Launch Game**

Use case ID: 001	Use case Name: Main Menu
Priority	High
Actors:	User
Use Case Summary	It will enable the User to select different options from main menu
Pre-condition:	The game is launched
Normal Course of Events	Alternate Path

1. The use-case begins when the user first launches the game	
2. The Main menu shall be exhibited	
3. User can chooses any many options from the menu.	
<b>Post Conditions</b>	
Main menu options are available and selectable.	

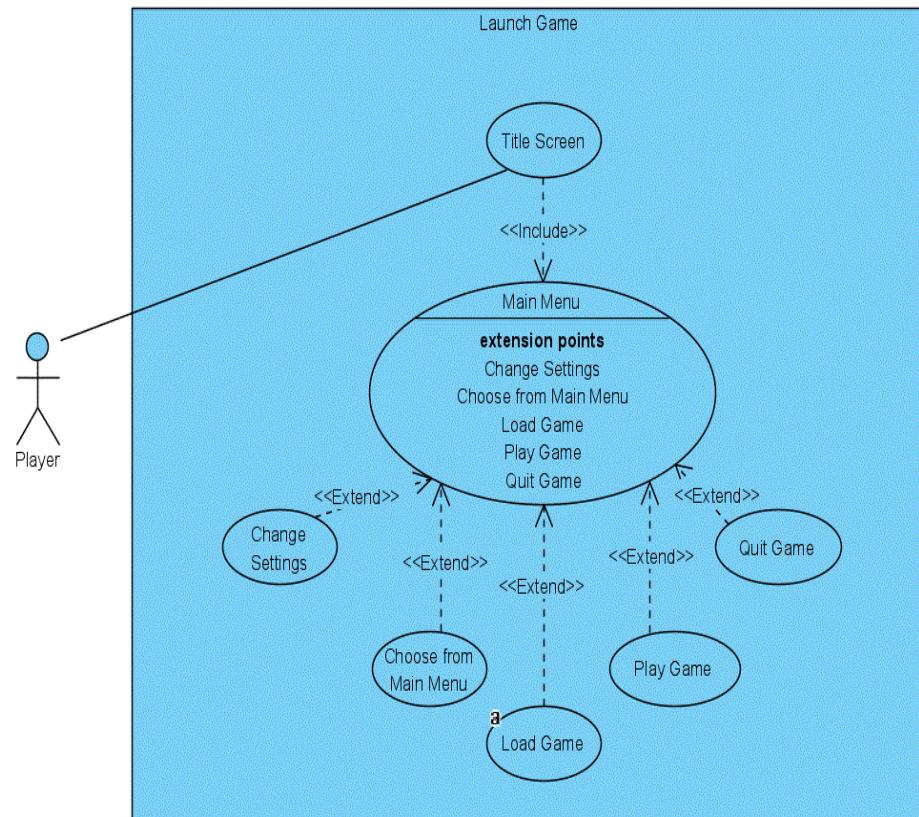
#### 7.4.2 Playable Character

Use case ID: 001	Use case Name: Playable Character
Priority	High
Actors:	User
Use Case Summary	It shall enable the User to control the playable character
Pre-condition:	The User have to first select the play-game option
<b>Normal Course of Events</b>	
1. The use-case begins when the user chooses the start game	
2. Map loads	
3. User solves puzzle	3(a) User may enter the wrong answer for the puzzle
4. This use case ends	
<b>Post Conditions</b>	

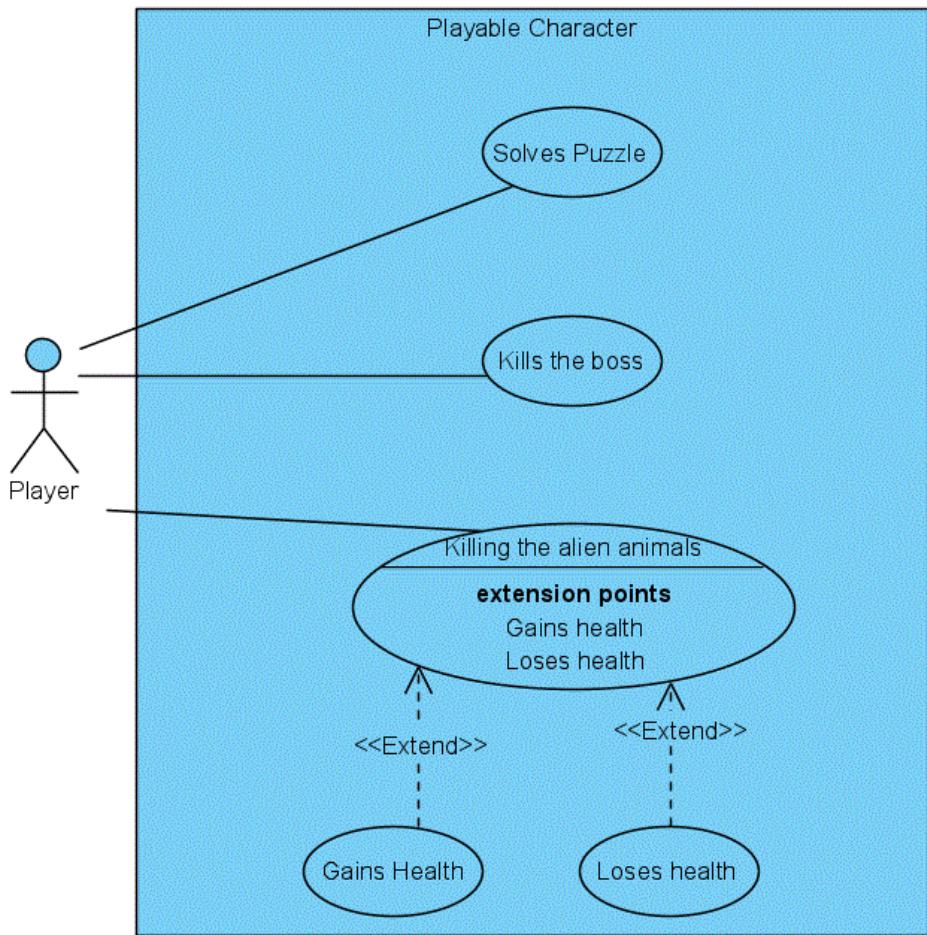
The user solves the puzzle until it the level is completed.

## 7.5 Use Case Diagram

### 7.5.1 Launch Game Use Case Diagram

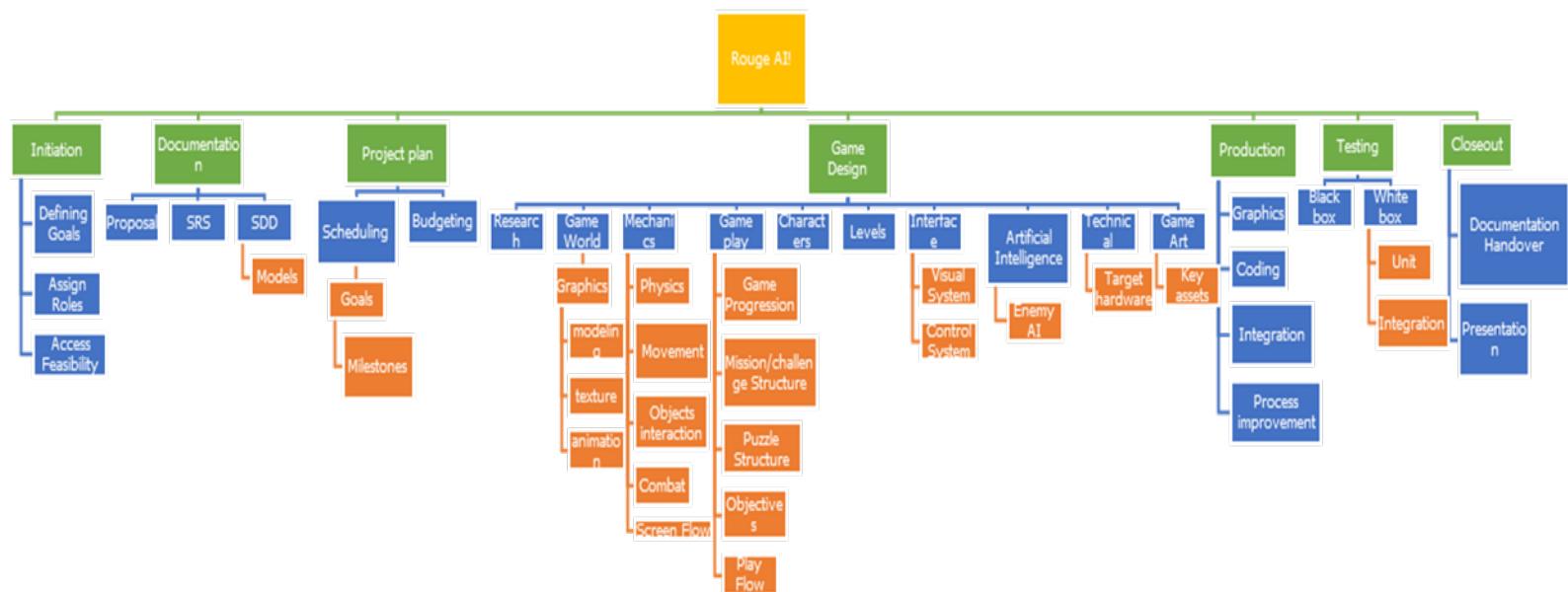


### 7.5.2 Playable Character Use Case Diagram

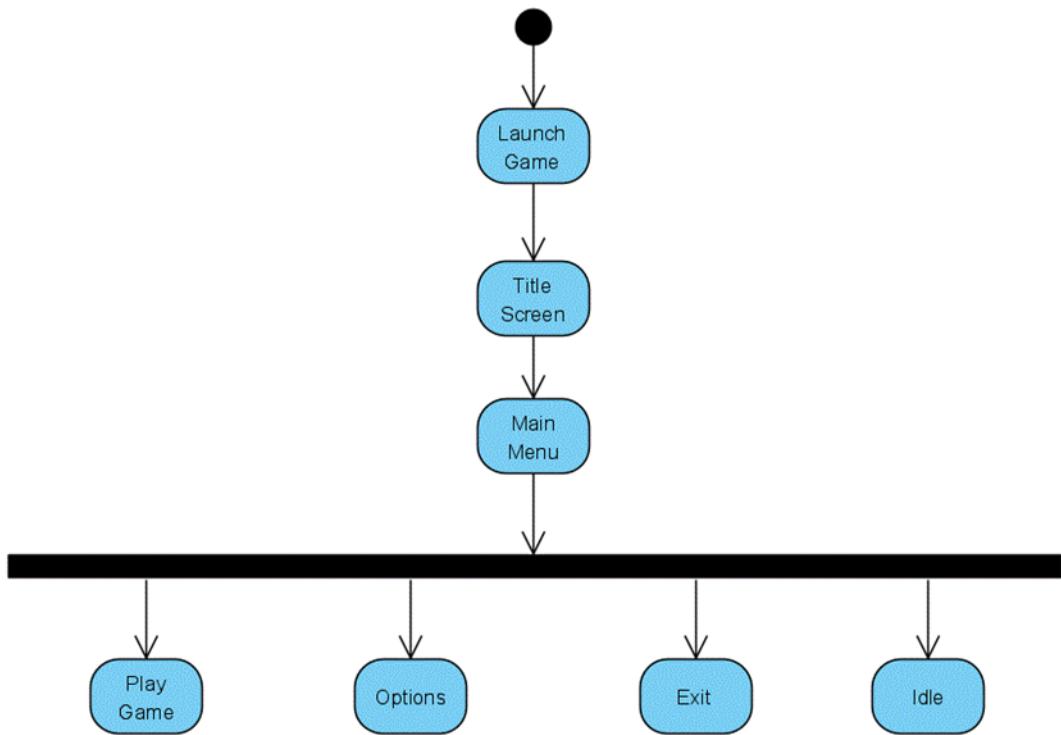


# 8 System Design

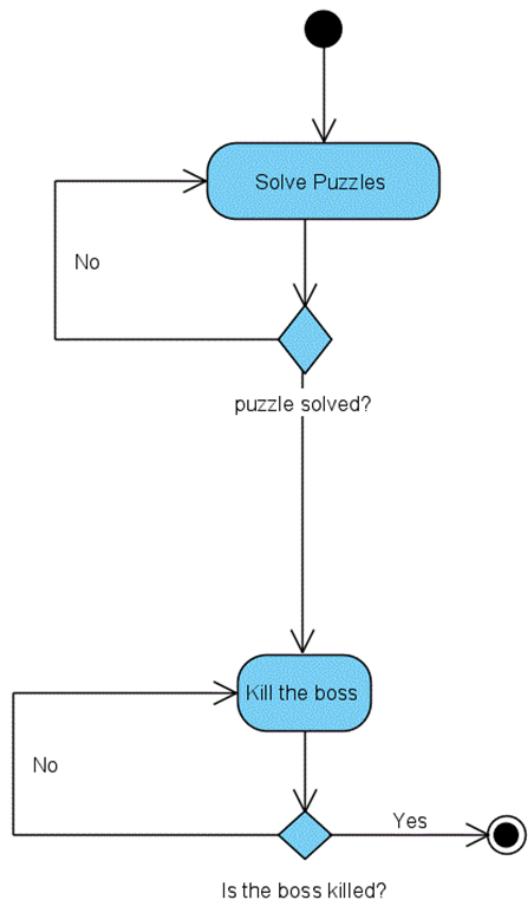
## 8.1 Work Breakdown System (WBS)



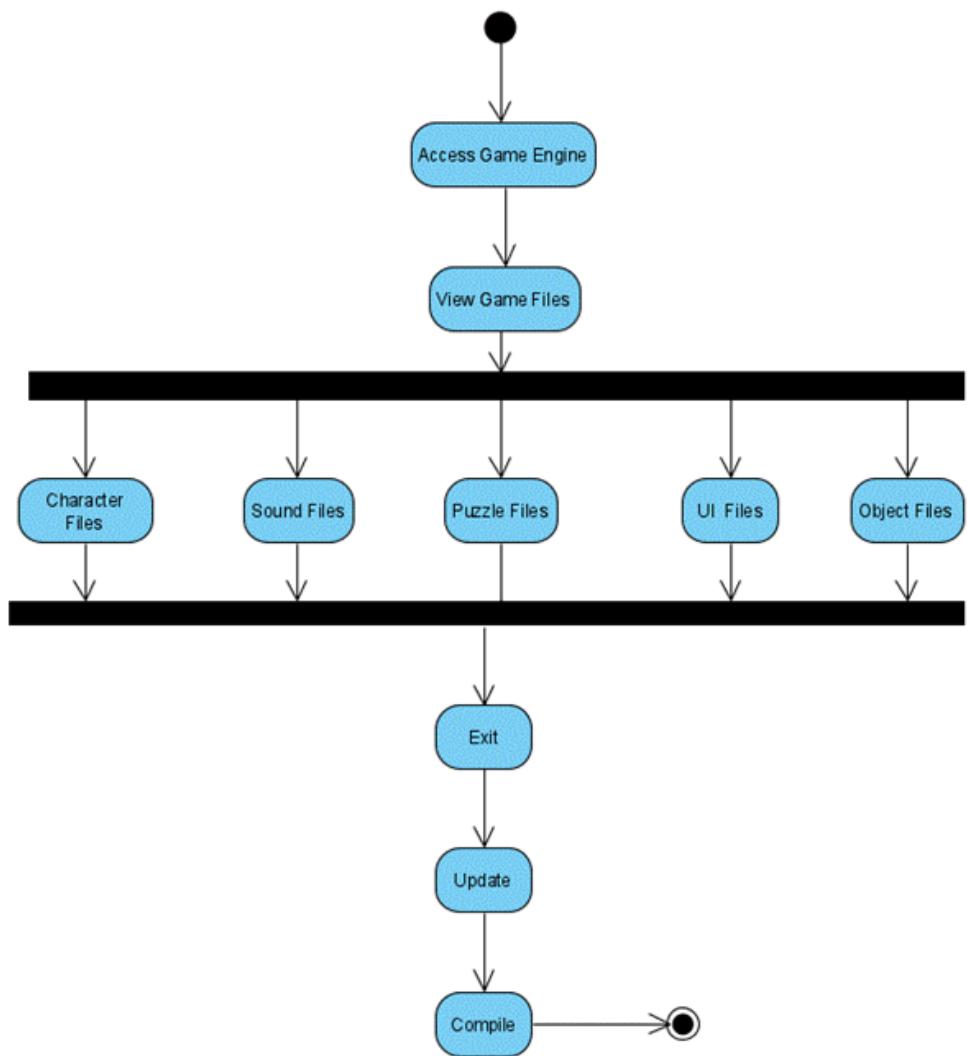
## 8.2 Launch Game Activity Diagram



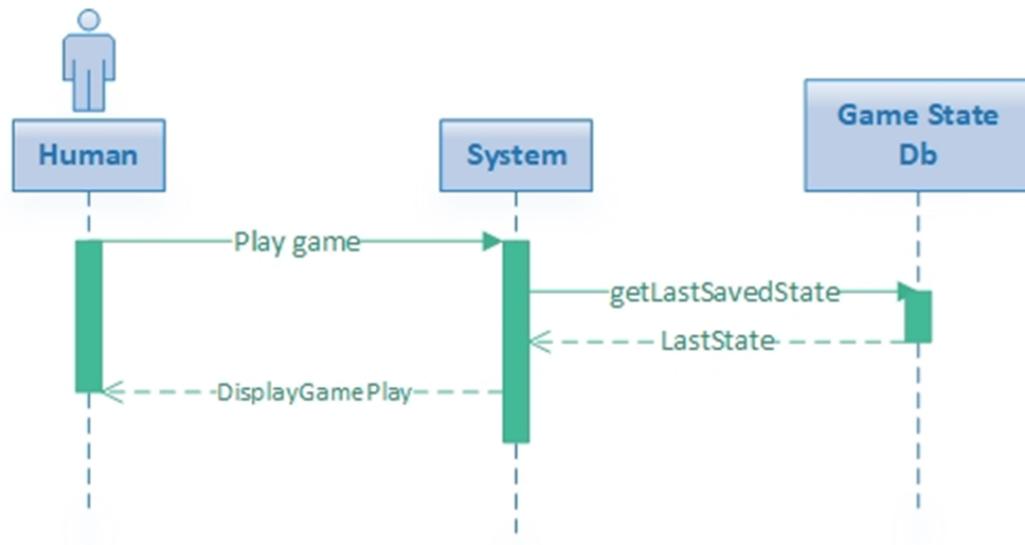
### 8.3 In Game Activity Diagram



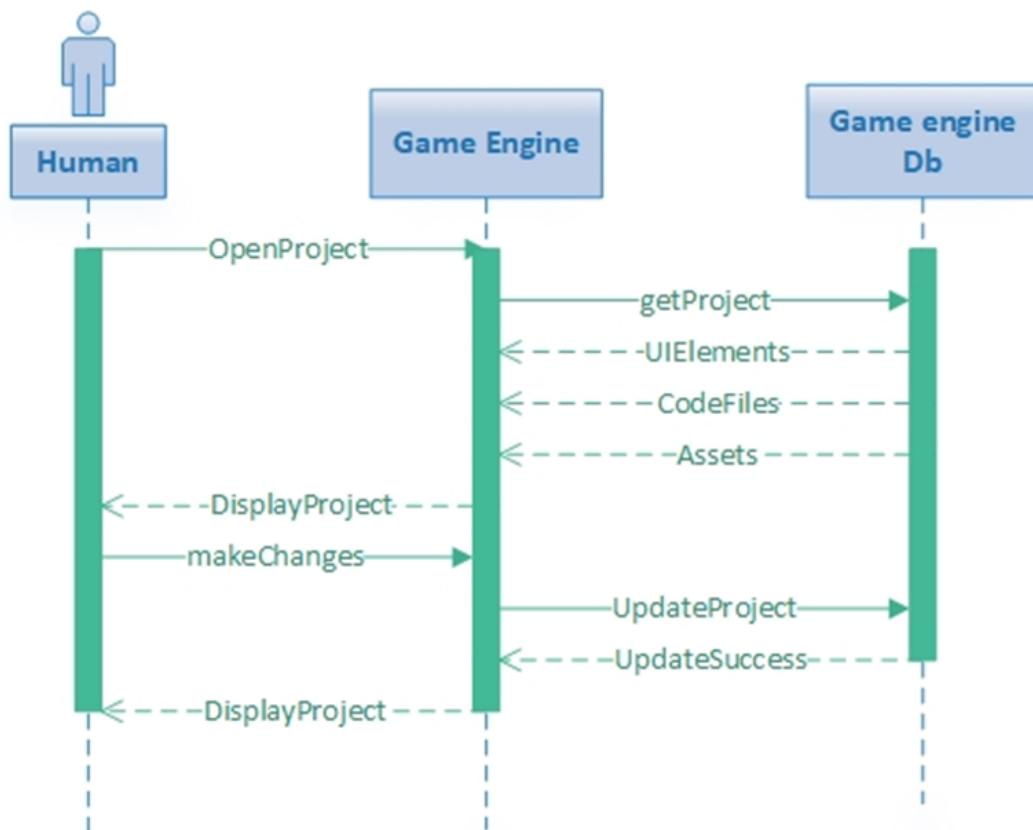
## 8.4 Game Update Activity Diagram



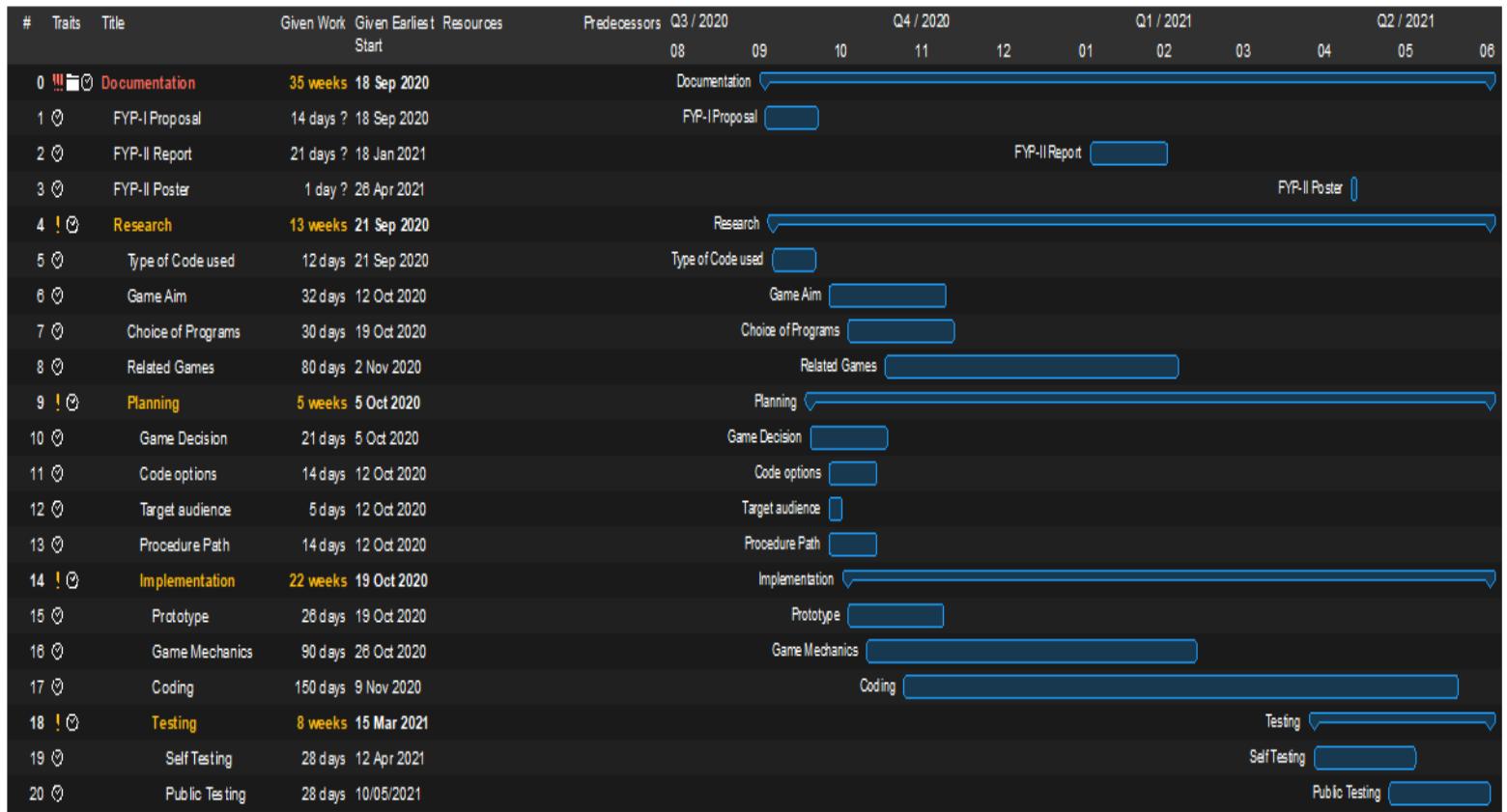
## 8.5 Sequence Diagram – Main Menu & In Game Activity



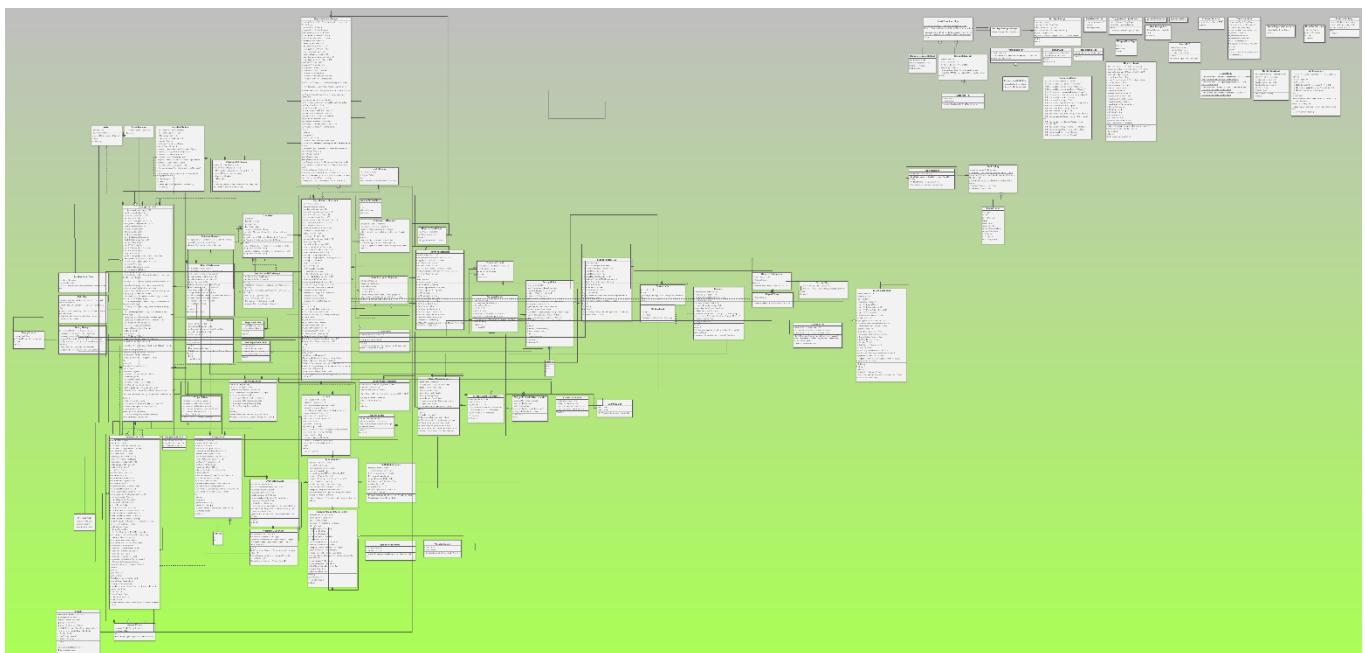
## 8.6 Sequence Diagram – Update Game



## 8.7 Gantt Chart



## 8.8 Class Diagram



## 9 System Testing

### 9.1 Test Case:

### 9.2 Unity Engine Testing:

#### 9.1.1 Test Case Game Installation

Purpose	Test “Game Installation”
Setup	Desktop
Step 1	Unity Setup is to be Downloaded
Step 2	Install Unity
Step 3	Installer file installed successfully
<b>Result</b>	<b>Installed</b>

#### 9.1.2 Test Case Game Launching

Purpose	Test “Game Launching”
Setup	Desktop
Steps	<ul style="list-style-type: none"><li>• Does the game launch as it's supposed to be?</li><li>• Game displays menu<ul style="list-style-type: none"><li>○ Play Game</li><li>○ Weapon Select</li><li>○ Settings</li><li>○ Exit</li></ul></li></ul>
<b>Result</b>	<b>Game launched and menu displayed</b>

### 9.1.3 Test Case Sounds

Purpose	Test “Sounds”
Setup	Desktop
Verification	<ul style="list-style-type: none"> <li>• Check ON/OFF Sound and Music</li> <li>• If the sound effects are synchronized with actions in the game.</li> </ul>
Result	<b>All sounds and music work as supposed to be.</b>

### 9.1.4 Test Case User Interface

Purpose	Test “User Interface”
Setup	Desktop
Verification	<ul style="list-style-type: none"> <li>• Check in Landscape orientation</li> <li>• Check for screen title</li> <li>• Font displayed properly (colour, size etc.)</li> <li>• Check if buttons are displayed properly</li> <li>• Check If camera follows the player as it should be.</li> </ul>
Result	<b>User interface works as it should to be</b>

### 9.1.5 Test case Performance

Purpose	Test “Performance”
Setup	Desktop
Verification	<ul style="list-style-type: none"> <li>• Make sure that any action is not taking considerable amount of time.</li> </ul>

	<ul style="list-style-type: none"> <li>• Check if gameplay is smooth</li> <li>• Check if the loading time is satisfactory.</li> <li>• Check if all assets are loading properly.</li> </ul>
<b>Result</b>	<b>Game performance is optimized</b>

#### 9.1.6 Test Case Puzzle Solving

Purpose	Test “Puzzle Solving”
Setup	Desktop
Verification	<ul style="list-style-type: none"> <li>• Check if player solves all the puzzle or not</li> <li>• Check if the player is rewarded accordingly.</li> </ul>
<b>Result</b>	<b>Puzzle count registered properly</b>

#### 9.1.7 Test Case Settings

Purpose	Test “Settings”
Setup	Desktop
Verification	<ul style="list-style-type: none"> <li>• Check Settings menu is displayed properly.</li> <li>• Check if the settings menu is working properly.</li> <li>• Check if the settings menu is displaying all the changeable content.</li> </ul>
<b>Result</b>	<b>Settings menu is displayed and functions properly</b>

### 9.1.8 Test Case Competition with the boss

Purpose	Test “Play Match”
Setup	Desktop
Verification	<ul style="list-style-type: none"><li>• Check if player attack the boss properly or not.</li><li>• Check if player is rewarded accordingly.</li></ul>
Result	<b>Player competes with the boss.</b>

## 9.2 Unity Engine Testing:

### 9.2.1 Game Installer Files

- Make installable builds with no errors.
- Game does not misbehave according to the design.
- Resource files of the game are installed.

### 9.2.2 Models and their Animation

- Character Models are displayed as made.
- Character Animations plays animated.
- Textures render as made.
- HUD shown as designed.

### 9.2.3 Player Game Object

- Can solve puzzles.
- Can kill the boss.
- Can kill the right Alien Animal and improve its health.

## 10 Conclusion

### 10.1 Problem Faced and Lessons Learned

Game development is basically a team work. Every person has its own role when it come to the development process but as per the current situation we weren't able to clear out our details for a really long time. Though we had several meetings but it takes time for an individual to understand the entire concept through online sessions. Face to face interaction easily solves a lot of ambiguities which rose for us this Semester. Documentation, Research, Planning, Implementation and Testing together make a game. So it indeed was difficult for us somehow.

Lessons learned is that we need to invest ourselves in Game development as it is helps to learn concurrency as well as it helps to learn about performance and user experience.

## 10.2 Project Summary

Rogue AI! is a game that is developed on Unity Engine for Desktop. It is being developed by keeping the constraints and limitations of the devices in view. It is basically developed in order to improve the logical thinking of the player as the player is to solve logical puzzles. The first four levels are crossed by solving the logical puzzle. The last level is to be solved by killing the boss enemy. In between the game, player has the option of regaining his health by killing the right Alien Animal, this shall boost his health.

## 11 Future Work

We have some plans of taking this game to new levels by adding some secret assets after achieving the currently planned milestone. We are looking forward to add wonders to it.

Future extension of this work may include:

- New designs
- Puzzle scenarios
- More players and enemies
- More Levels
- Interplanetary
- Reward schemes
- Daily Login Bonuses
- Different techniques of gaining health

## References

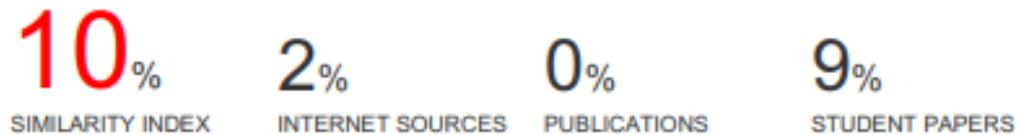
1. **Portal game play:**
  - <https://www.youtube.com/watch?v=XUp9arduWTg>
2. **Portal Wikipedia:**
  - [https://en.wikipedia.org/wiki/Portal\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Portal_(video_game))
3. **Portal 2 Wikipedia:**
  - [https://en.wikipedia.org/wiki/Portal\\_2](https://en.wikipedia.org/wiki/Portal_2)
4. **Half-life 2 Wikipedia:**
  - [https://en.wikipedia.org/wiki/Half-Life\\_2](https://en.wikipedia.org/wiki/Half-Life_2)
5. **Quantum Conundrum Wikipedia:**
  - [https://en.wikipedia.org/wiki/Quantum\\_Conundrum](https://en.wikipedia.org/wiki/Quantum_Conundrum)
6. **PUBG Mobile Wikipedia:**
  - [https://en.wikipedia.org/wiki/PlayerUnknown%27s\\_Battlegrounds](https://en.wikipedia.org/wiki/PlayerUnknown%27s_Battlegrounds)
7. **COD Mobile Wikipedia:**
  - [https://en.wikipedia.org/wiki/Call\\_of\\_Duty:\\_Mobile](https://en.wikipedia.org/wiki/Call_of_Duty:_Mobile)

# rouge ai

---

## ORIGINALITY REPORT

---



## PRIMARY SOURCES

---

1	Submitted to Higher Education Commission Pakistan	7%
2	www.coursehero.com	1%
3	Submitted to University of Nottingham	1%
4	Submitted to University of Greenwich	<1%
5	Submitted to Flinders University	<1%
6	en.wikipedia.org	<1%
7	www.ict-redesign.eu	<1%
8	Tom Tyler. "Difficulties", Parallax, 2020	<1%