Name: Upal Kundu Mark _____/50

1. Brief introduction __/3

Feature: MVP and Player Control

Our game is an escape room where the player must get out of all the rooms to win. As, TL-2: Software Architect, I will be creating 2 challenges in the game:

1 Puzzle Challenge

Players will be given a puzzle challenge to solve and match our team logo: Error 404. The player will be given a certain number of chances to solve the puzzle and then when they solve, they move to the next challenge.

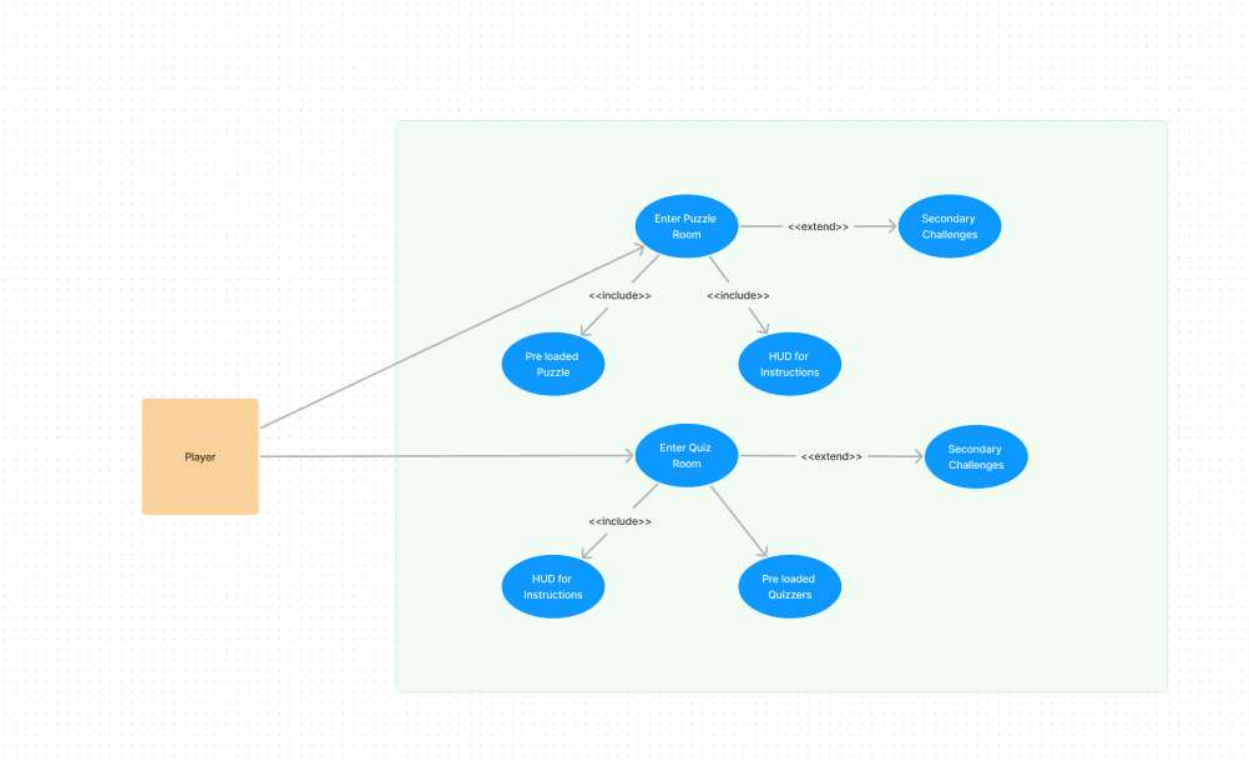
2 Quiz Challenge

Players will be given a certain number of quiz questions with the difficulty level increasing with each question. The questions will be AI-generated with GPT-4 API. Players have to answer all the questions before they can move to the next level.

Besides the main features, I will work on integration and overall architecture design for player control, level specification and the overall user experience.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Puzzle Game

Summary: The player enters the puzzle game and starts solving to match our logo. If they fail, they will be randomly selected for several chances.

Actors: Player

Preconditions: Puzzle scene is loaded

Basic sequence:

Step 1: Come to puzzle scene

Step 2: HUD for instructions on how to solve the puzzle.

Step 3: Accept player input and move puzzle components.

Step 4: If the player finishes the puzzle within the given chances, load the congratulations message and move on.

Exceptions:

Step 1: Puzzle Screen is not loaded [Retry loading the screen]

Step 2: Invalid input from user [Handle through Input validation and limiting controls]

Post conditions: Player moves on to the next quest

Priority: 1*

ID: C01

Name: Quiz Challenge

Summary: Players find and move to the quiz room where they are asked a random number of questions. Players try to get the best answer and are evaluated based on their answers. After finishing the quiz game, they move to the next challenge.

Actors: Player

Preconditions: GPT-4 API and Quiz Screen are loaded

Basic sequence:

Step 1: Enter Quiz room

Step 2: HUD for instructions on how to answer the quizzes.

Step 3: Accept player answers for each quiz question and evaluate on the backend

Step 4: If the player finishes the challenge, let them move to the next challenge

Exceptions:

Step 1: GPT-4 API is not loaded [Internet and Connectivity Issue]

Step 2: Player inputs send different signals to backend [Implement Input validation and limit controls]

Postconditions: The player moves on to the next quest

Priority: 2*

ID: C02

Name: Architecture Integration

Summary: My work is to ensure the whole game architecture is flawless and loads each screen/component. For faster load, implement threading, memory, and other controls. Also, make sure input validation is there so the player can't overload/ blank the infrastructure.

Actors: Computer and Connectivity

Preconditions: Basic game architecture is implemented for each screen/component.

Basic sequence:

Step 1: Test each component.

Step 2: Come up with plans for faster loading of screens and components.

Step 3: Test load and connectivity

Step 4: Research Unity loading and connectivity to come up with plans to improve our architecture.

Exceptions:

Step 1: The game does not load due to system configuration.

Step 2: Some graphics/ components do not perform as expected

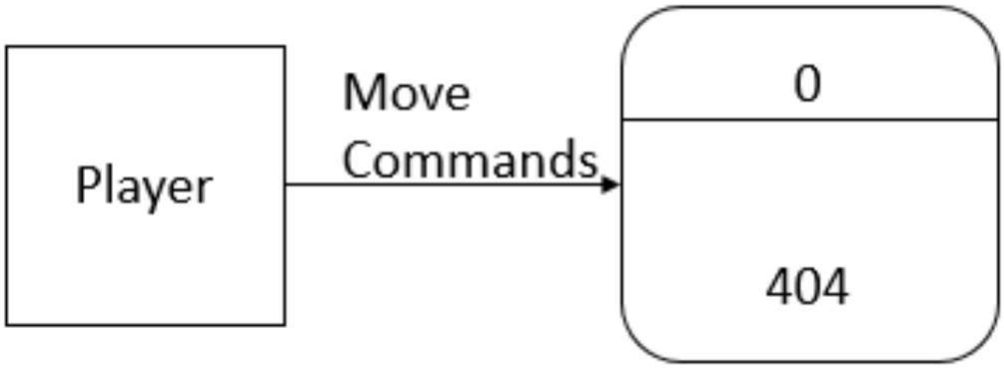
Postconditions: Smooth user experience and loading

Priority: 3*

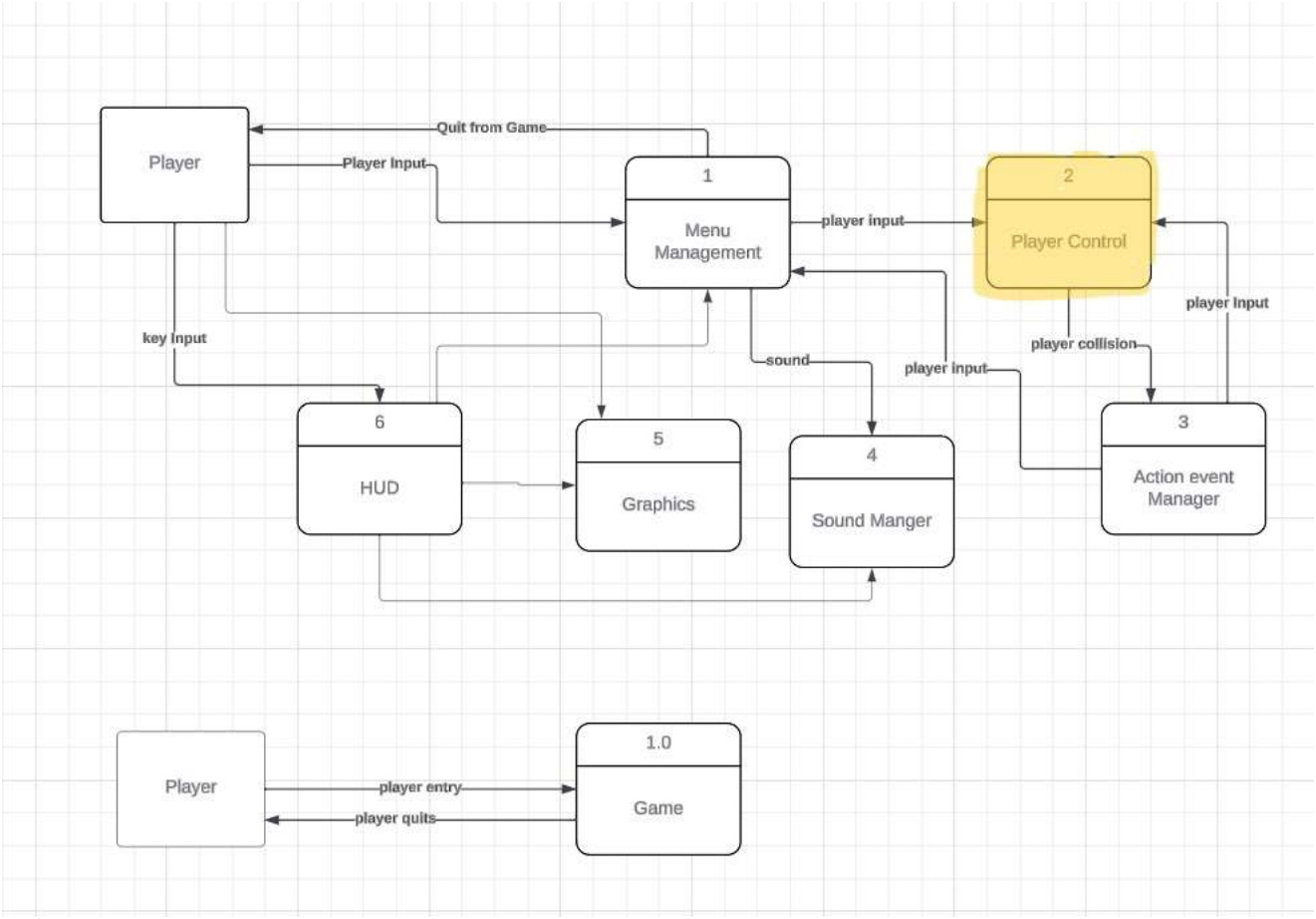
ID: C03

3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

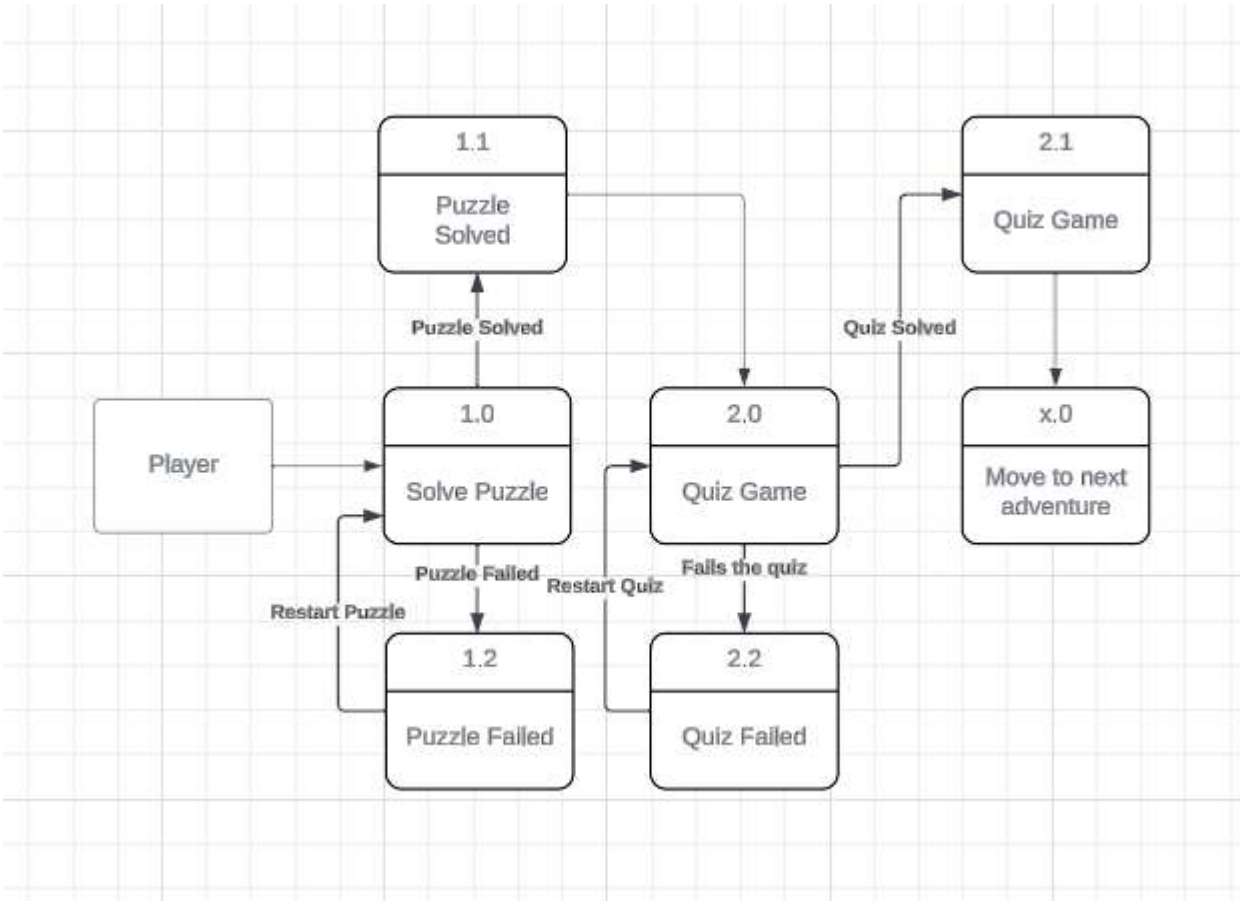
Data Flow Diagrams



Level 0 Diagram



Level 1 Diagram (Player Control and Game)



Process Descriptions

Puzzle Room

Enter Puzzle Room

HUD for instructions

Load puzzle screen

IF Puzzle Solved

Move to the main room to find a new challenge

ELSE

Move back to the puzzle screen

Quiz Room

Enter Quiz Room

HUD for instructions

Quiz Questions Generated in Backend

IF QUIZ questions solved

Move to the main room to find a new challenge

ELSE

Move Back to QUIZ room to get more questions

4. Acceptance Tests _____9

Automated QUIZ Question Generation

Run the GPT-4 API 100 times sending question and answer output to a file.

Function Generate Question()

The output file will have the following characteristics:

{

Question: AI-generated Question

}

Check for acceptable length and difficulty of questions.

Automated Quiz answer generation

Run the GPT-4 API 100 times generating question and automated answer output

Function QuestionEvaluator();

The output file will have the following characteristics:

{

Question: AI-generated Question

Sample Answer: Automated Answer

Evaluation: AI Evaluation for answers to questions

}

Automated Puzzle solver

Run puzzle solver backend script and output successful and unsuccessful attempts

Function TryPuzzle();

Output Structure

{

Attempt1: { Move: right

Solve_percentage: 10%},

Attempt2: {Move: left,

Solve_percentage: 20%},

…….

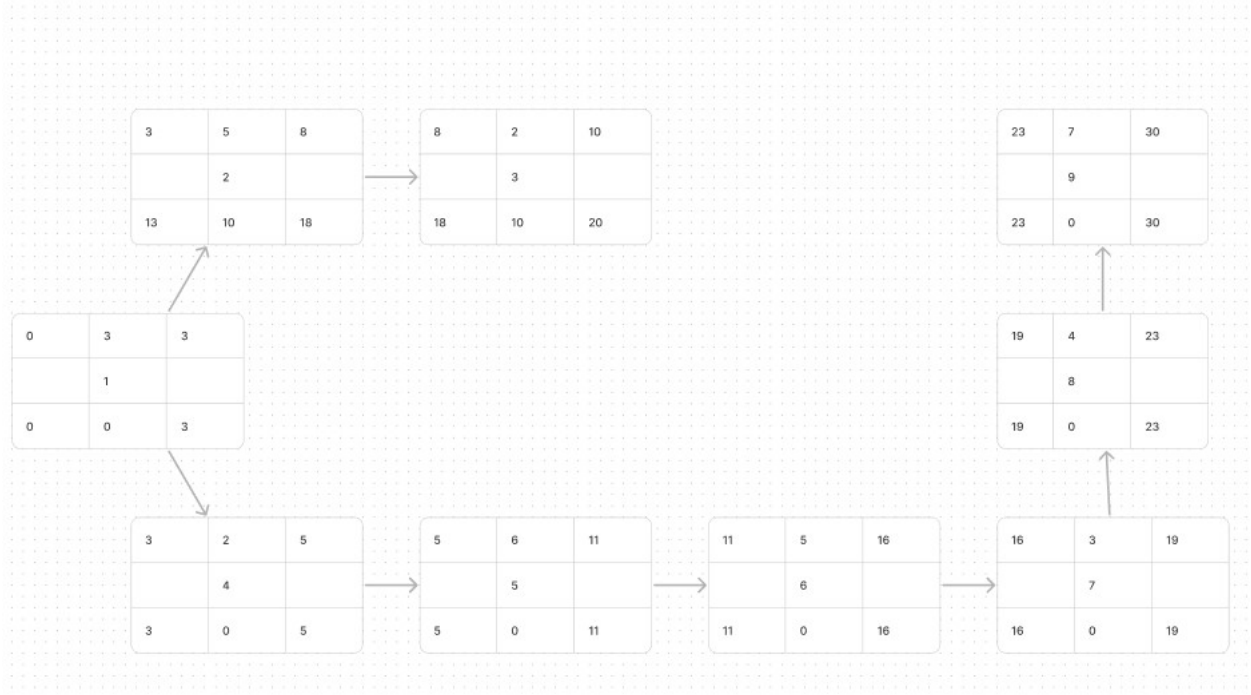
}

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Analysis	3	-
2. Puzzle Screen Design and movement	5	1
3. Test Puzzle Solver	2	2
4. GPT-4 API setup	2	1
5. Quiz Generation and Answer Screen	6	4
6. Quiz Evaluation Backend Setup and Test	5	5
7. Integration of screens	3	6
8. System Requirements Check	4	7
9. Integration with Other's games	7	8

Pert diagram



Gantt timeline

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	Indicator: Red (Slack Time), Green (Work Hours)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2	1. Requirements Analysis																														
3	2. Puzzle Screen Design and movement				1																										
4	3. Test Puzzle Solver									2																					
5	4. GPT-4 API setup				1																										
6	5. Quiz Generation and Answer Screen						4																								
7	6. Quiz Evaluation Backend Setup and Test												5																		
8	7. Integration of screens																	6													
9	8. System Requirements Check																				7										
10	9. Integration with Other's games																								8						