BAHRIA UNIVERSITY KARACHI CAMPUS

Department of Software Engineering

COURSE: CEL 220
COMPUTER ARCHITECTURE & LOGIC DESIGN
PROJECT PROPOSAL
CLASS: BSE – 3B (FALL - 2023)

Project Title

Group Members

S. No.	Name	Enrollment #
01	Rehab Khaleel	02-131222-108
02	Asad Ali	02-131222-125
03	Hassaan Gul Rao	02-131222-114

Submitted to:

Course Instructor: Engr. Ramsha Mashood
Lab Instructor: Engr. Noor us Sabah

Date: 04/12/2023

Chance - A Text Based Adventure Game

1. INTRODUCTION & BACKGROUND

Chance, a text-based adventure game has a lot of potential as it is written on low level language, in this case, MIPS assembly language. It provides the user with simple yet interactive game that keeps the user entertained. The game setup is simple with user input commands, decision points for quests and challenges. Although assembly language is not used in games now a days, it is not rare to find one too.

2. PROBLEM STATEMENT

1. **Educational Value:**
   The educational value of other games is less as compared to MIPS making it more difficult to understand the basics.
2. **Platform Dependent:**
   Some games are platform dependent which causes reliability issues.
3. **Poor Resource Optimization:**
   System resource optimization is not that good in other text-based adventure game.

3. PROPOSED SOLUTION

3.1. FEATURES OF THE PROJECT

1. **Educational Value:**
   The educational value of the game Chance is more than traditional games.
2. **Platform Independency:**
   The game developed is platform independent making it more reliable.
3. **Good Source optimization:**
   System Resource optimization is great as we are directly working with the CPU.

3.2. METHODOLOGY

- The project will be implemented in MIPS assembly language, utilizing low-level instructions for better optimization of system resources.
- The flow of logic using branching, conditional statements would provide an interactive gaming experience.

3.3. TECHNOLOGIES TO BE USED

The project will be developed using:

- Assembly Language: **MIPS**
- No graphical user interface is implied due to the low-level nature of assembly language.
- Software: **MARS**

4. PROJECT SCOPE

- The MIPS Assembly Text Based adventure game project is scoped to optimize the system resources making it a efficient and responsive way to interact with user. Due to the limitations of assembly language, graphical features may be minimal.
- The quest narrative would be quite simple yet entertaining with multiple quests and potential expansion for additional quests.

5. PROJECT ABSTRACT

1. Game Setup:

- The game starts with an introduction, setting the scene and presenting the player with a scenario or quest.
- Information about the game world, characters, and objectives is provided through text.

2. User Input:

- Players interact with the game by entering text commands.
- Common commands may include actions like "look," "go," "take," and "use" or by providing choice number.

3. Game State and Logic:

- The game maintains an internal state that represents the current situation, locations, and inventory.
- Logic is implemented to handle player input, update the game state, and determine the outcome of actions.

4. Decision Points:

- Players encounter decision points where they must choose between different actions or paths.
- These decisions affect the storyline, introducing branching narratives and multiple possible outcomes.

5. Quests and Challenges:

- Players embark on quests, solve puzzles, or face challenges that require thoughtful decisions.
- Progression often involves exploring different locations, interacting with characters, and gathering information.

6. Game Over and Endings:

- Failure to make the right decisions or solve challenges may lead to a game over.
- Different endings are possible based on the player's choices throughout the game.

7. MIPS Assembly Implementation:

- The game logic, user interface, and input/output handling are implemented using MIPS assembly language in Mars stimulator.

8. Limited Graphics:

- Graphics are limited in a text-based adventure game on MIPS, and the focus is on textual descriptions and user choices.

9. Educational Purpose:

- Developing a text-based adventure game in MIPS can be an educational exercise, reinforcing concepts of assembly language programming, logic design, and system interaction.

6. MODULE DISTRIBUTION

1. **Main Module:**

Contains the main program logic, initializes the game, and manages the overall flow of the game.

2. **Input Module:**

Handles user input processing, including parsing commands and validating input.

3. **Output Module:**

Manages the display of text and descriptions of the game world.

4. **Game State Module:**

Manages the internal state of the game, including player location, inventory, and quest progress.

5. **Quest Module:**

Defines the quests, dialogues, and decision points in the game. Implements the logic for branching storylines based on player choices.

6. **Location Module:**

Defines the different locations or areas in the game world. Contains descriptions, characters, and items specific to each location.

7. **Character Module:**

Defines the characters in the game, including their attributes, dialogues, and interactions with the player.

8. **Item Module:**

Manages the items that the player can interact with or collect in the game. Includes logic for using, combining, or discarding items.

9. **Event Module:**

Handles special events or triggers in the game, such as scripted sequences, puzzles, or encounters.

10. **Utility Module:**

Contains utility functions that can be reused throughout the code, such as string manipulation or random number generation.

7. REFERENCES

https://www.makeuseof.com/tag/browser-text-based-games/

Teacher's Signatures: ___________________________

Remarks: ____

____

____