

# **Elabyrinth Game Application**

## **Software Requirement Specification (SRS) Document**

### **Sprint 1 Implementation**

**Project Timeline: 21.10.2022 to 27.10.2022**

# INDEX

1. Introduction	
1.1 Purpose	4
1.2 Intended audience	4
1.3 Intended use	4
1.4 Scope	4
2. Overall description	5
2.1 Assumptions and dependency	5
3. System feature and requirements	5
3.1 Functionality	5
3.1.1 Start Menu	5
3.1.2 Select Maze	5
3.1.3 Play Game	5
3.2 System requirement	6
3.2.1 Tools to be used	6
3.3 System feature	6
4. DataFlow Diagram	
4.1 DFD level 0	8
4.2 DFD level 1	9

## **1. Introduction: -**

The Software Requirement Specification (SRS) helps us to understand the entire software and give us the idea of how and what our application is going to do. The whole SRS document contains the overview description about our application, purpose, scope, tools which are used in our application and also basic overview of those tools. The focus of our SRS is to give minute details about our Labyrinth Game with the help of problem statements and examples like stories.

**1.1 Purpose:** -The purpose of this document is to show the requirements for the Labyrinth application, in which players have to pass the given maze successfully in defined constraints.

**1.2 Intended Audience:** - This application is intended to be played by kids.

**1.3 Intended Use:** -

- Kids (Pre school and nursery kids)

Since this a general-Purpose Software any one can access it.

**1.4 Scope:-** This project focuses on creating the development of an eLabyrinth application. Which takes the player name and shows how many constraints(time/moves) are remaining to win the game successfully. We record the number of moves played by each players and the player with minimum number of moves wins the game. The probability of winning the game is the ration of zero's to the size of matrix.

## **2. Overall Description: -**

A Labyrinth is a tree-based maze in the Royal House of England where the royals often enjoyed themselves with their servants who would get in and find their way out of a labyrinth maze.

Even today unsuspected tourists who visit these get lost and one can often hear cries for help. Leading us to this project where we feel the computer Linux based version will be highly safer! An electronic version based on logical ICs and EPROMs was created by Elektor a Dutch based magazine in 1984. A reloaded version based on a microcontroller was also done in 2020. We will implement it based on a Linux Console.

## **2.1 Assumptions and Dependency: -**

- System should have Ubuntu Linux installed.
- System should have either 4GB or more RAM.
- The service is used preferably on a playstations for kids.
- For youngsters to have a fun time at game zones in malls.

## **3.System Features and Requirements: -**

### **3.1 Functionality: -**

**3.1.1 BG\_01-> Start Menu:** This is the first main menu level function that provides flexibility to enable the user to do either select the game or exit the application.

**3.1.2 BG\_02-> Select Maze:** This function is the sub menu and start point of the game where the user is asked to select the level of difficulty of the maze he wants to play.

**3.1.3 BG\_03-> Play Maze:** This function is used to play the maze game.

### **3.2 System Requirements: -**

#### **3.2.1. Tools to be used:**

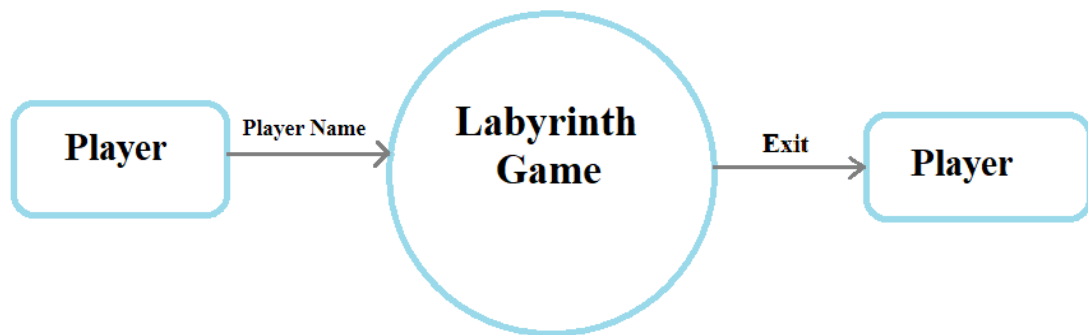
- 2-D Arrays and Data Structures
- C File Handling
- C Language
- System Programming

### **3.3 System Features: -**

- Supportability: The system is easy to use.
- Design Constraints: The system is built using only C language.
- Usability: The maze game is make for gaming lovers who like to solve challenges.
- Reliability & Availability: The system is available 24/7 that is whenever the user would like to use the system, they can use it up to its functionalities.
- Performance: The system will work on the user's terminal.

#### 4. DataFlow Diagram:

##### 4.1 DFD Level 0 -



#### 4.1 DFD Level 1 -

