

SOFTWARE REQUIREMENT SPECIFICATION

Version 2.0

"SNAKEZZZ..."

(A Game created for Course Project)

CS 101: Computer Programming and Utilization

IIT BOMAY

Team Members:

Utkarsh Gautam (Team Leader)

Arushi Bansal

Surbhi Sahu

Anay Tripathi

Date:

19/10/2014

CONTENTS

1.0 Introduction

- 1.1 Purpose
- 1.2 Scope and Utility
- 1.3 References
- 1.4 Overview

2.0 Overall Description

- 2.1 Perspective
- 2.2 Features-
 - Mazes
 - Levels
 - Movement
 - Food
 - Collisions
- 2.3 Operating Environment
- 2.4 User Classes and Characteristics
- 2.5 Use Cases

3.0 Requirements

- 3.1 Functional Requirements
- 3.2 User Interface Requirements
- 3.3 Hardware Interface Requirements
- 3.4 Software Interface Requirements

INTRODUCTION

1.1 Purpose

The purpose of this document is to present a detailed description of the game “Snakezzz...”. The game is commonly found in cell phones and computers. This document will explain the various features, requirements and interfaces of the game created by us, how it will be played and react to external stimuli.

1.2 Scope and Utility

The game is a one player game which puts the user’s reflexes to test. The player moves the snake in a maze via keyboard commands in the four directions-up, down, left & right. The aim is to eat the food appearing periodically on the screen to gain points while avoiding collisions with the walls of the maze and itself. Every time the snake consumes food it grows longer thereby increasing the game’s difficulty with time and adding to its dynamics and maintaining interest.

1.4 References

- 1.Lecture slides on graphics by Abhiram G. Ranade
- 2.Book: An introduction to programming through C++ *by* Abhiram G. Ranade
(Special reference –Chapter 20 section 20.4)

1.4 Overview of Document

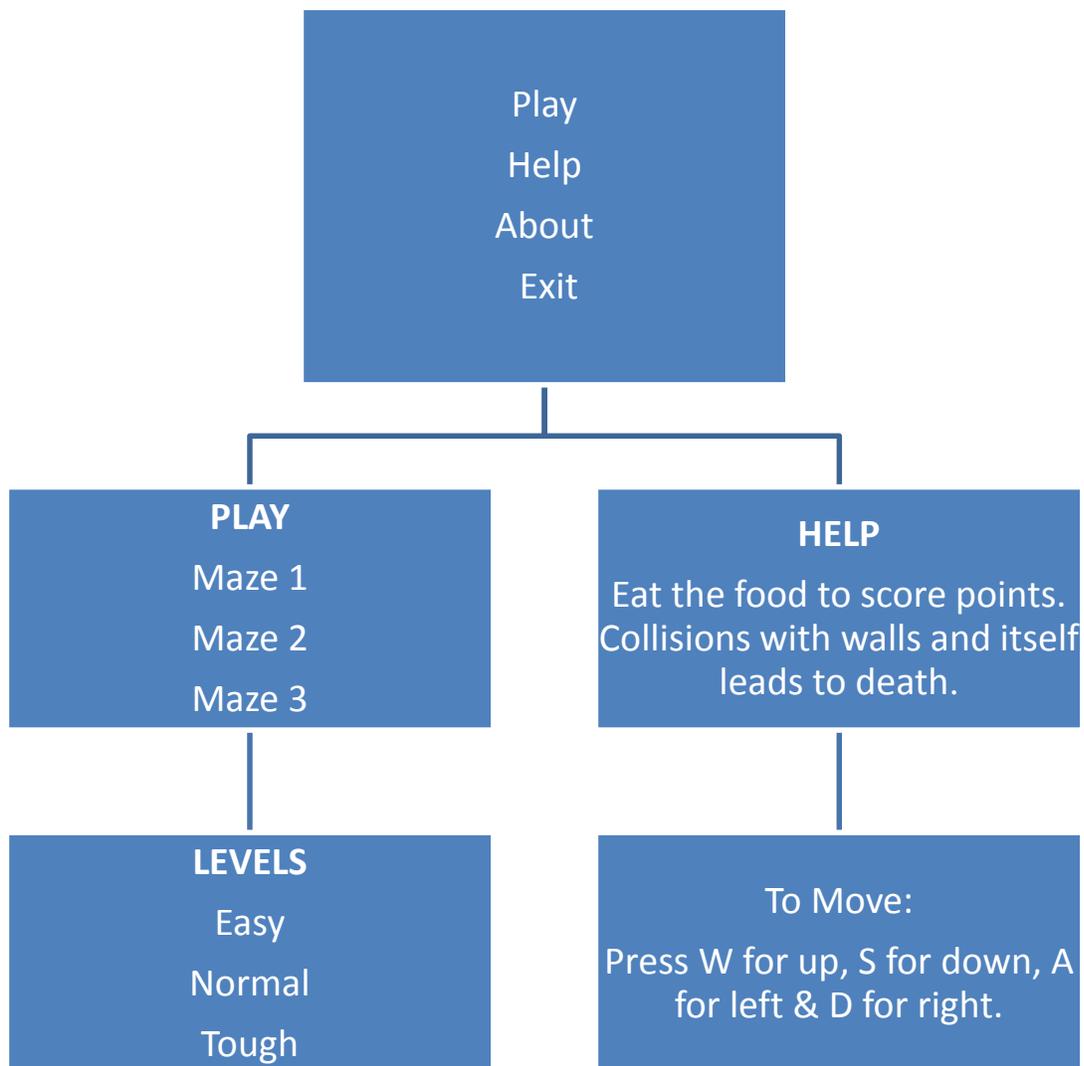
The introduction identifies the product developed i.e. the game “Snakezzz...”, and its main objective.

Next, the overall description explains the working of the game in layman terms. This part is targeted for all readers of the SRS document.

The requirements portion is mainly for software developers and explains in greater technical detail what are the functions, UIs, hardware and software requirements for working of the game. Detailed description of algorithms of various functions is given in the project report. A user manual is also available alongside.

OVERALL DESCRIPTION

2.1 Perspective



2.2 Features

Mazes-The user gets to choose from predefined mazes.

Levels-Higher the level, greater is the difficulty; as speed of snake increases with the level.

Movement-The snake can be moved in 4 directions-up, down, left and right.

Food- Food appears for the snake periodically at random positions in the maze.

Collisions-Collision of snakes head with walls of the maze or its own body leads to death and end of game.

Scores-Points are earned every time food is eaten.

2.3 Operating environment

The game is designed to be played on laptops to run it. The laptop must have a gcc compiler pre-installed to run the program.

2.4 User Classes and Characteristics

The game is for recreational purposes and may be played by people of all ages and professions. It has no limitations and no special skills are required to play it. It is in fact quite simple to play yet with sufficient challenge that keeps it interesting.

2.5 Use Cases

The user can choose to play, get help, know about the developer or exit. As the user chooses to play he gets the option to choose the maze and level he wants to play. While playing he can move the snake up, down, left or right using the keys w, s, a or d resp. A detailed user manual has also been enclosed with this.

REQUIREMENTS

3.1 Software Interface Requirement

The game has been created using Code Blocks IDE:: and GCC Compiler with integrated simplecpp library. More libraries are included for time/speed variations.

3.3 Hardware Interface Requirements

The user will interact with the game app via a keyboard and mouse of any computer..

3.3 Functional Requirements

Several functions need to be created.

Movement-The snake is an array of circle elements whose coordinates are stored in an nx2 array(n-length of snake). Circular queue is used and coordinates of last element are upgraded to be at the front of the first element in order to move the snake. The snake can be moved in 4 directions up, down , left and right.

Food- Food appears for the snake periodically at random positions in the maze. The randomization is achieved via random function: rand().

Checks-a)Death: Collision of snakes head with walls of the maze or its own body leads to death and end of game. For this a check function is created which on collision will end the game declaring "Game Over".

b)Eating Food: Another check function will test food consumption. This function gives bonus score and increases the length of the snake.

Distance- This function was created to check the position of mouse click during the initial navigation through various options like play/help/choose a maze/level etc.

Scores-Points earned for food consumption are incremented and returned at the end of the game.

3.4 User Interface Requirements

The game is based on graphical user interface and requires a simple laptop/desktop with windows, linux or any other gui, keyboard, mouse and monitor. It may also be played on touch screen laptops.

End of SRS