

(Phase one)
Software Requirement
Specification
(SRS)

ID	Name	Email	Mobile
20120247	Omar Hatem Abdel Kader	dantesprada@hotmail.com	01091188062
20120237	Abdullah Hussein Sayed	abdullah.mantawi@gmail.com	01154036519
20120243	Ali Mohamed Bastawy	alibasta2012@yahoo.com	01157526851

Version 1.2

Document purpose and audience ::

Purpose ::

This document describes in details the system in terms of functional requirements , non-functional requirements and some basic concepts including UML use case

Audience ::

This document is directed to our T.A (teacher assistant) which in this case is considered as client and to

1.Introduction

- 1.1. Purpose
- 1.2. Scope
- 1.3. Definitions, acronyms and abbreviations

2.Requirements

2.1. Functional Requirements

- 2.1.1. Starting The game
- 2.1.2. Stopping the game
- 2.1.3. Establish game server
- 2.1.4. Connect to game's server
- 2.1.5. Snake Movement
- 2.1.6. Score

2.2. Non-functional Requirements

- 2.2.1. Usability
- 2.2.2. Reliability
- 2.2.3. Performance
- 2.2.4. Implementation
- 2.2.5. Interface
- 2.2.6. Legal

3.System Model

3.1 Scenario

- 3.1.1 Use Case Model
- 3.1.2 Actors
- 3.1.3 New Game
- 3.1.4 Snake's Movement

4. Ownership Report

1 Introduction

1.1 Purpose ::

The purpose of this SRS is to establish the requirements for " 2D Snake Server Game" project and to identify the context in which the software will operate with respect to specific operating systems, users.

Generally, the SRS may be inspected by any reader with some mathematical/programming skills and interests. Required knowledge would be good mathematical/programming skills, and good understanding of computer software and hardware in general. The actual intended audience of this SRS is the project group itself and TAs.

Careful review and understanding of the Software Requirement Specification (SRS) will ensure that the requirements outlined are correct and that the subsequent end product will lead to the fulfillment of the requirements .

1.2 Scope ::

Our goal in the "2D Snake Server Game " project is to produce a playable multiplayer game using a computer as server in which other players can play with each other not exceeding four players

1.3 Definitions, acronyms and abbreviations ::

Please refer to Appendix A for definitions of terms, acronyms, and abbreviations specific to the information presented in this SRS.

2 Requirements ::

2.1 Functional Requirements ::

The "2D Snake Server Game" supports only one type of user which is the player . The "2D Snake Server Game " tasks include starting the game , control the snake movement , establishment and joining a server for multiplayer mode .

2.1.1 Starting The game ::

From the interface of the game or the main menu the player / user will be able to access "New Game " button in which he can start playing the game .

2.1.2 Stopping The game ::

The player can exit the game by pressing the "back" button after pressing that button the player score and number of fruit that the snake ate in this turn of the game then he will be able to return to the main menu

2.1.3 Establish game server ::

The server computer can establish a server for the game in which a maximum number of players is three can be connected to this server . " Make Server " button that make the above function is available at the main menu of the game .

2.1.4 Connect to game's server ::

The other players that have the game application on there devices can access the game server established by the server computer by pressing " Connect to server " button in the main menu

2.1.5 Snake Movement ::

The game simply gives the players the ability of controlling the snake movement by the arrow buttons in order to get more fruits and achieve a higher score . The player that have the higher score wins , if the snake touched itself or any other snake then the player loses

2.1.6 Score ::

By collecting more fruit the player score goes higher and the player with the higher score wins.

2.2 Non Functional Requirements ::

2.2.1 Usability ::

easy to play ::

The rules of the game and the controls are in the simplest form

Snake Movement ::

By using the direction arrows on the keyboard the player can easily control the snake

2.2.2 Reliability ::

Crash safe ::

The "2D Snake Server Game" should be crash free 95% of the game runtime

2.2.3 Performance ::

Short Response Time ::

The response for the game commands shouldn't take more than 2 seconds , establishing the server and joining the server takes a little bit longer about 20 seconds

2.2.4 Implementation ::

Programming Language ::

The "2D Snake Server Game" should be fully implemented in Java

2.2.5 Interface ::

Simple user interface ::

The "2D Snake Server Game" have a simple interface in which it have the buttons to new game , establish server , join server and exit game button . The game itself have simple GUI environment in which the snake moves and collect the fruit

2.2.6 Legal ::

Cheap game ::

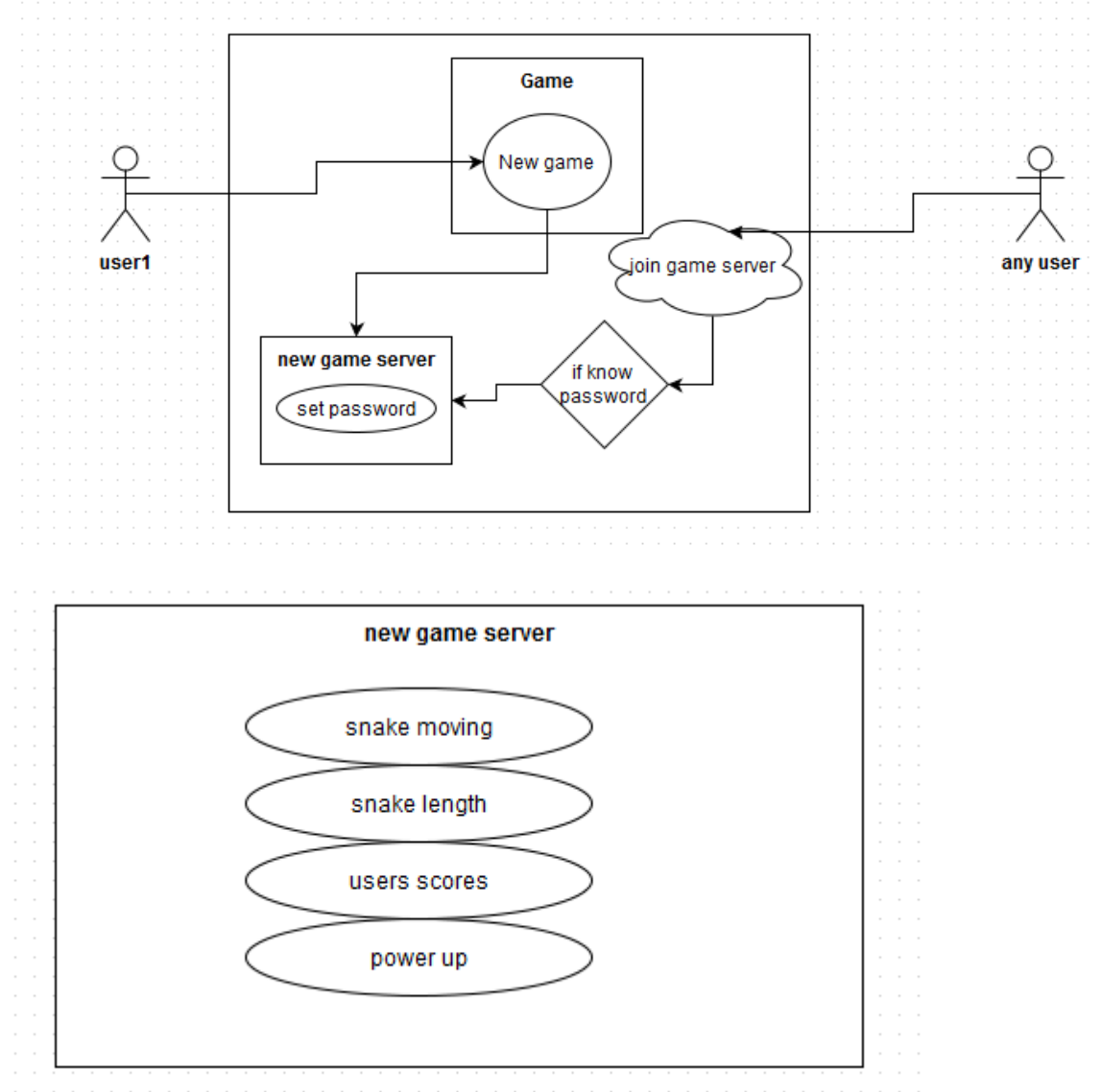
This game is totally established by student at the faculty of computer and information - Cairo university with no budget at all

3.System Model ::

3.1 Scenario ::

3.1.1 Use Case Model ::

The following diagram gives an overview of the identified use cases.



3.1.2 Actors ::

The player is our main actor but there are two kind of player in our system .

First type is the one the establish the server and set a password for this server .

Second type is the one that join the server and enter a password to access the server.

In the game play there is no difference between the second and first type.

3.1.3 New Game ::

Use Case	New game	
Actor	Player	
Precondition	The game just started	
Flow Of Events	Actor Steps	System Steps
	1 . the player starts The game by tapping The game icon	
		2. The game starts by showing the player the game menu
	3. by choosing new game the player either establish or connect a server with a password	

		4. the game establish or join the server that can only handle three players then the game play starts
--	--	---

3.1.2 Snake' Movement ::

Use Case	Snake Movement	
Actor	Player	
Precondition	The game is already started and the player started playing	
Flow Of Events	Actor Steps	System Steps
	1. the player simply controls the movement of the snake with arrow buttons	
		2. the game takes the movement commands given by the player and apply it 3. If the player hit itself or another player the game ends

	3. the player collect fruits by controlling the snake's movement	
		4. the game increment the score and the length of the snake when the player collect fruit

4.Ownership Report ::

In the following table we show the owners of each part in this document .

Item	Owner
Introduction , functional and non-functional requirements	Omar Hatem Abdel kader
UML use cases	Abdullah Hussein Sayed
Scenarios	Ali Mohamed Bastawy

For TA Use Only

Authorized Signature: _____

Date : _____

Appendix A ::

Establish server :

The game creates some sort of link in which other players can join to play together

Join server ::

It means joining the link that the game made to player with other player

Snake ::

Of course here we don't mean a real snake we mean a virtual snake that player play with it

Fruit ::

It virtual figures that the snake should collect to gain more length and higher score