Design Document for Snake Game

Keyur Patel
Alex Guerrero
Shafeeq Rabbani

December 8, 2015

Contents

1 Revision History

Table 1: Revisions

Name	Date	Description
Shafeeq Rabbani	08/12/2015	Created Revision Table
Shafeeq Rabbani	08/12/2015	Project description added to Introduction
Shafeeq Rabbani	08/12/2015	Fixed section 6.1.1 (previously 5.1.1
Alex Guerreroi	08/12/2015	Edited document

2 Introduction

This project is about the redevelopment of the Snake game. The Snake game is a traditional computer game which involves the player controlling a 'snake' on the screen that swallows 'food' that randomly appears on the screen, growing in size every time it does so. The player should avoid hitting the snake into itself and the borders around its space as this would end the game.

The following documentation is intended elaborate how the design of the snake game is implemented. The document will also examine the details of the functional and non-functional requirements mentioned in the Software Requirements Specification. This document is intended for the following readers:

- Designers: The document can serve to ensure that all functional and non-functional requirements are met. Further, designers can also use this document to verify any discrepancies among different modules.

- New Project Members: This document can bring new team members up-to-date with the overview and structure of the game.

- Maintainers: This document can also used to understand the structure of the game and all of the modules within. It would then be the responsibility of the maintainers to update the Design document by mentioning any change they have made to it.

- Professor and TAs: As this document is being marked, the professors and TAs will have access to the document to determine if it was structured as intended. The document will also serve to give the Professor and the TAs an overview of the modules of the snake game.

The Snake game has been divided up into modules which hide information from other modules in the document in order to implement the Information Hiding principle of Software Engineering. Further, the modules only share the information amongst each other that is necessary. This is done to ensure the Low Coupling principle of Software Engineering. In order to read or modify values within different modules, there are getter and setter methods unique to the respective modules. This is done to implement the Encapsulation principle of Software Engineering.

This document consists of the Module Interface Specification (or MIS) which is intended for programmers who work to further develop the Snake Game.

The rest of this document consists of the GANTT and PERT chart meant to highlight the time frame for future deliverables for development of the Snake Game.

3 Anticipated and Unlikely Changes

There are two types of possible changes: Anticipated and Unlikely Changes. This section covers both of these changes.

3.1 Anticipated Changes

AC1: The type of food which the snake eats.

AC2: What must happen after the game is over.

AC3: More options available in the menu as game expands .e.g. save game, high scores, sound, difficulty etc.

AC4: The map of the game.

AC5: Different PowerUps will be available such as Intangibility (.i.e. ability to go through walls) as the game expands.

AC6: The characteristics of the snake .e.g. its color, the way it grows, the speed at which it moves etc.

AC7: Addition or modification of Keyboard and mouse commands as game is expanded in the future.

3.2 Unlikely Changes

These are design decisions that have to be changed after they were fixed in the software architecture state (in order to simplify the design). It wasn't intended that these decisions would have to be changed.

UC1: Input/Output devices (Input: Keyboard, Mouse Output: Screen).

UC2: The game will always be implemented in python using the Pygame library.

UC3: The goal of the game is to get the highest score possible.

4 Module Hierarchy

This section lists the modules in the Snake game. The modules listed below are leaves in the module hierarchy in the table below.

M1: Food Module

M2: GameOver Module

M3: MainMenu Module

M4: PlayMap Module

M5: PowerUp Module

M6: Snake Module

M7: Controller Module

The Hardware-Hiding Module is already implemented by the operating system and hence will not be reimplemented.

Level 1	**Level 2**
Hardware-Hiding Module	
Behaviour-Hiding Module	Food Module GameOver Module MainMenu Module PlayMap Module PowerUp Module Snake Module Controller Module
Software Decision Module	

Table 2: Module Hierarchy

5 Connection Between Requirements and Design

The table below highlights the connection between the system requirements(which are listed in the Software Requirements Specification) and the modules.

AC	Modules
AC1	M1
AC2	M2
AC3	M3
AC4	M4
AC5	M5
AC6	M6
AC7	M7

Table 3: Trace Between Anticipated Changes and Modules

6 Module Decomposition

6.1 Behaviour-Hiding Module

Secrets: The contents of the required behaviours.

Services: Includes programs that provide externally visible behaviour of the system as specified in the software requirements specification (SRS) documents. This module serves as a communication layer between the hardware-hiding module and the software decision module. The programs in this module will need to change if there are changes in the SRS.

6.1.1 Food (M1)

Secrets: The structure of the Food

Services: Generates and stores a Food object in a random position

Implemented By: SWHS

6.1.2 GameOver Module (M2)

Secrets: The format and structure of the Power up

Services: generates and stores a powerup object in a random position

Implemented By: SWHS

6.1.3 MainMenu Module (M3)

Secrets: The format of the game board

Services: updates state of the game board and returns objects to be displayed

Implemented By: SWHS

6.1.4 PlayMap Module (M4)

Secrets: The format and and state update redirection of the backend

Services: emulates the game window and different states and menus

Implemented By: SWHS

6.1.5 PowerUp Module (M5)

Secrets: The behaviour of the Snake object

Services: defines the points of the snake and has functions to manipulate snake

Implemented By: SWHS

6.1.6 Snake Module (M6)

Secrets: Attributes pertaining to the snake such as its length.

Services: Determines the speed the snake moves at, its length etc.

Implemented By: SWHS

6.1.7 Controller Module (M7)

Secrets: The algorithm for coordinating the running of the program.

Services: Provides the main program.

Implemented By: SWHS

7 Traceability Matrix

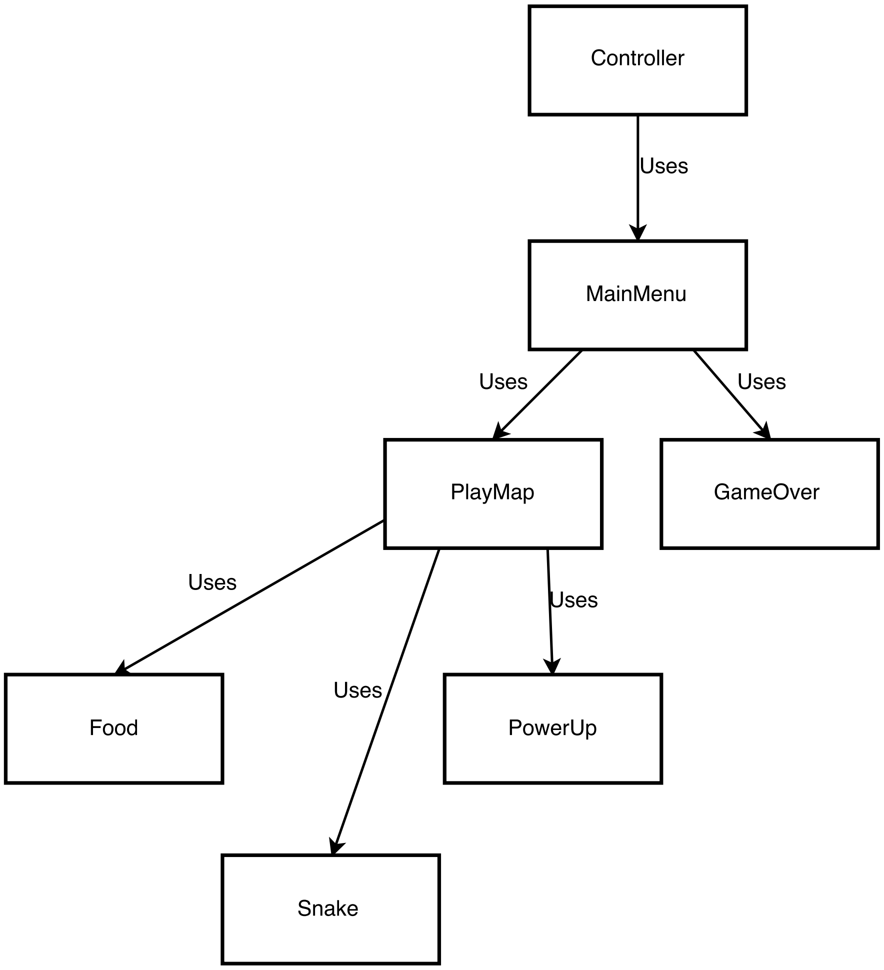
This section shows the traceability matrix between the modules and the requirements.

Req.	**Modules**
R1	M2
R2	M2, M8
R3	M2,M5,M6,M7
R4	M3, M6, M7
R5	M2
R6	M2,M5,M6,M7
R7	M2,M3,M4,M5,M6
R8	M2,M4, M5

Table 4: Trace Between Requirements and Modules

8 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided.


Controller
Uses
MainMenu
Uses
Uses
PlayMap
GameOver
Uses
Uses
Uses
Food
PowerUp
Snake

9 Course Schedule

In this section, a Gantt and PERT Chart scheduling the remainder of the semester are provided.

Snake

Nov 6, 2015

McMaster, CAS Department

Project manager	
Project dates	Nov 3, 2015 - Dec 8, 2015
Completion	0%
Tasks	28
Resources	3

Remake of the classic arcade game snake, in python.

Tasks

Name	Begin date	End date
Write Design Doc	11/3/15	11/6/15
Module Guide	11/3/15	11/4/15
MIS	11/5/15	11/6/15
Schedule	11/3/15	11/6/15
Design Doc Due	11/6/15	11/6/15
Implementation	11/7/15	11/11/15
Controller Module	11/7/15	11/9/15
View Module	11/7/15	11/9/15
Create Pause State	11/7/15	11/9/15
Refine Code	11/10/15	11/11/15
Unit Testing	11/12/15	11/15/15
Model Module	11/12/15	11/15/15
Controller Module	11/12/15	11/15/15
Viewer Module	11/12/15	11/15/15
System Testing	11/14/15	11/16/15
Rev 0 Demo	11/17/15	11/17/15
Revise SRS	11/17/15	11/20/15
Revise Test Plan	11/17/15	11/20/15
Revise MG	11/20/15	11/23/15
Revise MIS	11/20/15	11/23/15
Revise Implementation	11/19/15	11/23/15
Testing	11/19/15	11/26/15
Usability Test	11/24/15	11/26/15
Test Report Due	11/27/15	11/27/15
Prepare Final Demo	11/28/15	11/30/15
Final Demo	12/1/15	12/1/15

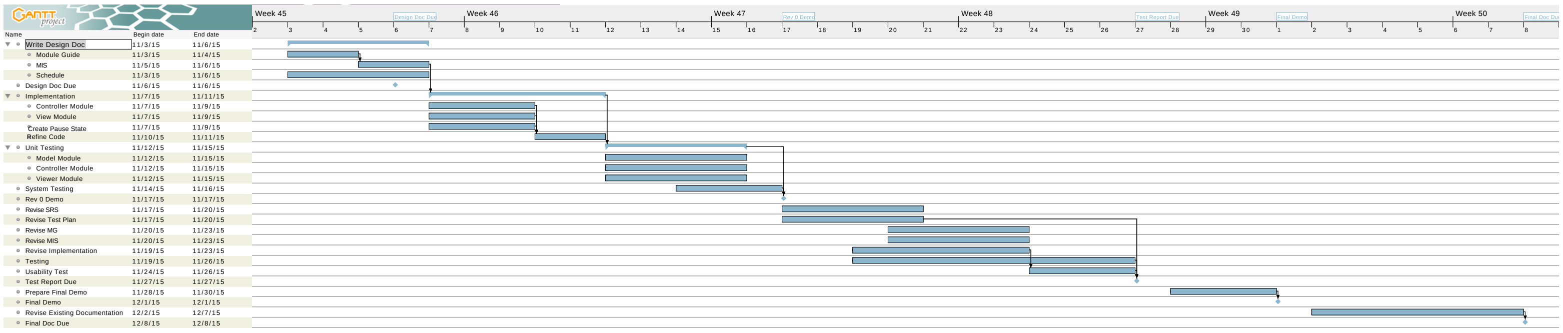
Tasks

Name	Begin date	End date
Revise Existing Documentation	12/2/15	12/7/15
Final Doc Due	12/8/15	12/8/15

Resources

Name	Default role
Alex Guerrero	undefined
Shafeeq Rabbani	undefined
Keyur Patel	undefined

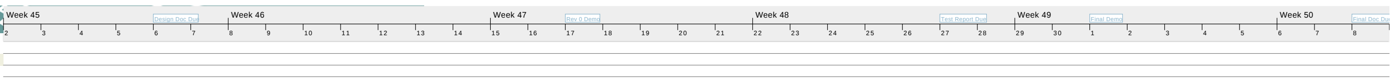
Gantt Chart



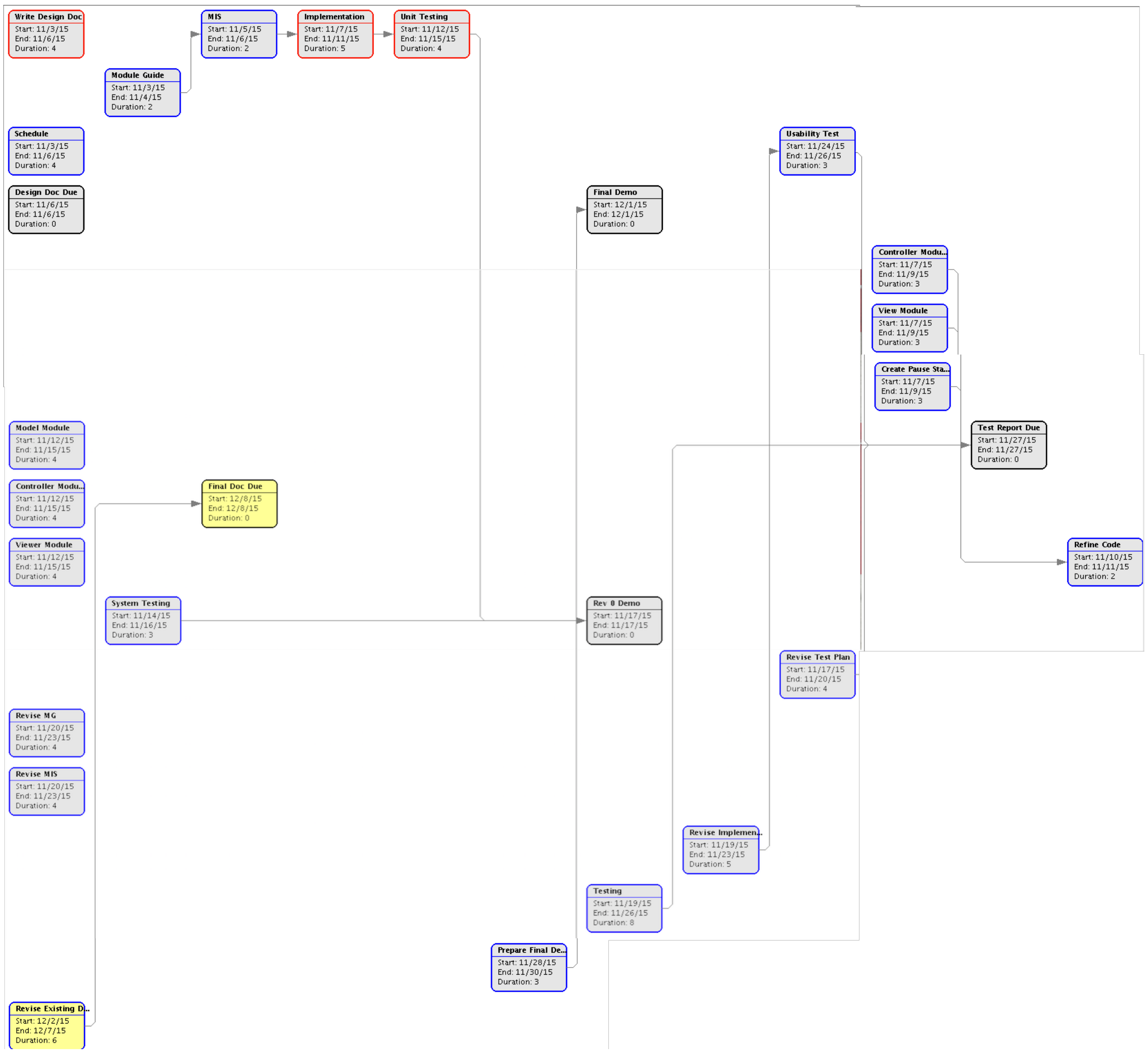
Name	Begin date	End date
Write Design Doc	11/3/15	11/6/15
Module Guide	11/3/15	11/4/15
MIS	11/5/15	11/6/15
Schedule	11/3/15	11/6/15
Design Doc Due	11/6/15	11/6/15
Implementation	11/7/15	11/11/15
Controller Module	11/7/15	11/9/15
View Module	11/7/15	11/9/15
Create Pause State	11/7/15	11/9/15
Refine Code	11/10/15	11/11/15
Unit Testing	11/12/15	11/15/15
Model Module	11/12/15	11/15/15
Controller Module	11/12/15	11/15/15
Viewer Module	11/12/15	11/15/15
System Testing	11/14/15	11/16/15
Rev 0 Demo	11/17/15	11/17/15
Revise SRS	11/17/15	11/20/15
Revise Test Plan	11/17/15	11/20/15
Revise MG	11/20/15	11/23/15
Revise MIS	11/20/15	11/23/15
Revise Implementation	11/19/15	11/23/15
Testing	11/19/15	11/26/15
Usability Test	11/24/15	11/26/15
Test Report Due	11/27/15	11/27/15
Prepare Final Demo	11/28/15	11/30/15
Final Demo	12/1/15	12/1/15
Revise Existing Documentation	12/2/15	12/7/15
Final Doc Due	12/8/15	12/8/15

Resources Chart

Name	Default role
Alex Guerrero	undefined
Shafeeq Rabbani	undefined
Keyur Patel	undefined



PERT Chart


Write Design Doc
Start: 11/3/15
End: 11/6/15
Duration: 4
Module Guide
Start: 11/3/15
End: 11/4/15
Duration: 2
MIS
Start: 11/5/15
End: 11/6/15
Duration: 2
Implementation
Start: 11/7/15
End: 11/11/15
Duration: 5
Unit Testing
Start: 11/12/15
End: 11/15/15
Duration: 4
Schedule
Start: 11/3/15
End: 11/6/15
Duration: 4
Design Doc Due
Start: 11/6/15
End: 11/6/15
Duration: 0
Usability Test
Start: 11/24/15
End: 11/26/15
Duration: 3
Final Demo
Start: 12/1/15
End: 12/1/15
Duration: 0
Controller Modu...
Start: 11/7/15
End: 11/9/15
Duration: 3
View Module
Start: 11/7/15
End: 11/9/15
Duration: 3
Create Pause Sta...
Start: 11/7/15
End: 11/9/15
Duration: 3
Test Report Due
Start: 11/27/15
End: 11/27/15
Duration: 0
Model Module
Start: 11/12/15
End: 11/15/15
Duration: 4
Controller Modu...
Start: 11/12/15
End: 11/15/15
Duration: 4
Final Doc Due
Start: 12/8/15
End: 12/8/15
Duration: 0
Viewer Module
Start: 11/12/15
End: 11/15/15
Duration: 4
Refine Code
Start: 11/10/15
End: 11/11/15
Duration: 2
System Testing
Start: 11/14/15
End: 11/16/15
Duration: 3
Rev 0 Demo
Start: 11/17/15
End: 11/17/15
Duration: 0
Revise Test Plan
Start: 11/17/15
End: 11/20/15
Duration: 4
Revise MG
Start: 11/20/15
End: 11/23/15
Duration: 4
Revise MIS
Start: 11/20/15
End: 11/23/15
Duration: 4
Revise Implemen...
Start: 11/19/15
End: 11/23/15
Duration: 5
Testing
Start: 11/19/15
End: 11/26/15
Duration: 8
Prepare Final De...
Start: 11/28/15
End: 11/30/15
Duration: 3
Revise Existing D...
Start: 12/2/15
End: 12/7/15
Duration: 6