School of Health and Society
Department Design and Computer Science
Kristianstad University

Course: DA120B

Program: Software Development

Authors: Yihui Wang, Paul Takem, Zhanyao Zhang, Micheal Okoro

Feeding the Snake

Table of Contents

1. Introduction

Several snake games exist, but all represents their origin from the normal “Snake game” concepts where the player manoeuvres a line which grows in length, with the line itself being a primary obstacle. The concept originated in the 1976 arcade game Blockade [1] The beauty of the game was that it was a slightly different experience each time you played. [2] Our game is based on this game and adds much more features than it.

We create a movable snake, with food and enemies on the screen, and there are some special apples which could give the snake more strength such as speeding up, the ability to kill the enemies and being invisible to the enemies. The player uses the arrow keys to provide directions of movements. When the snake eats an apple, it increases in length, and the goal of our game is to make the snake as long as possible in a limited time. The game over when the snake meets enemies or time is up.

The screenshot of our game is graphically illustrated in Fig.1 below, given a view on how the game interface is.

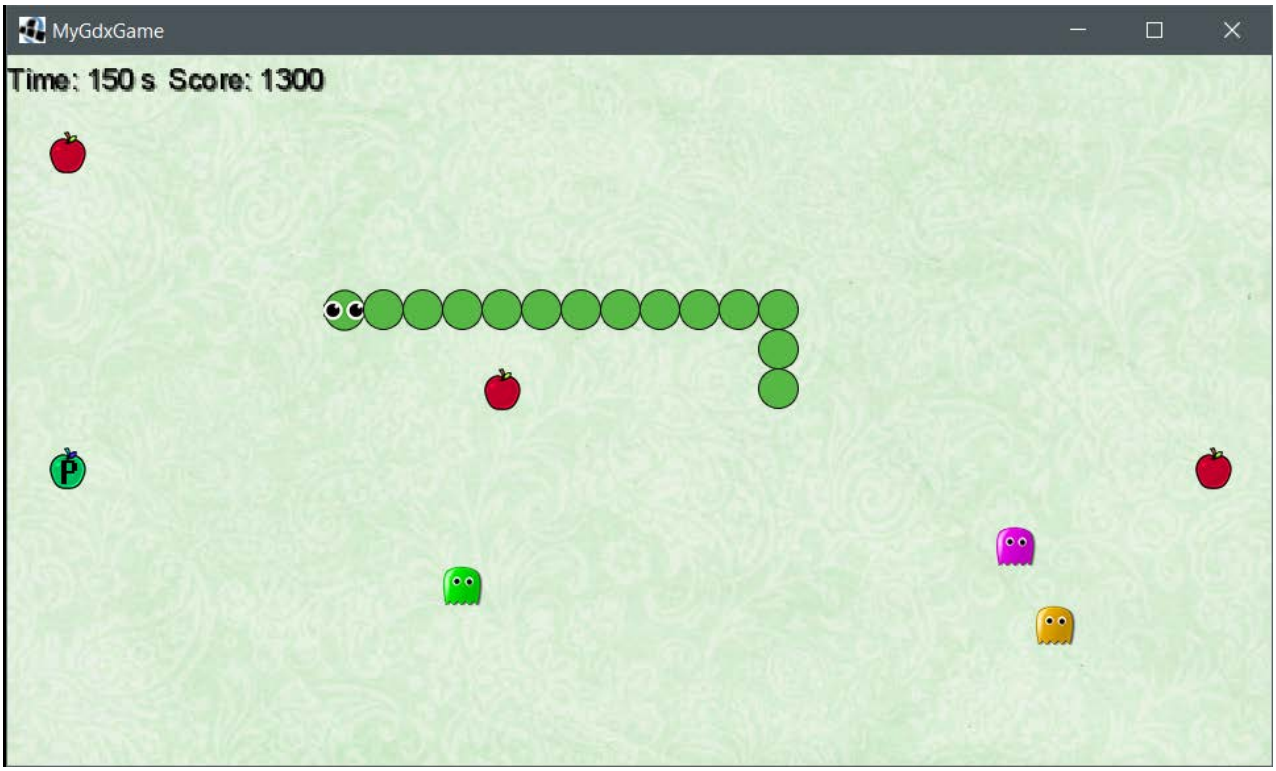

Fig.1: The screenshot of the game

1.1 Restriction

Some restrictions are encountered here in the process of designing and coding of the game.

- The using of snake block makes the code more readable and understandable
- The game width and height variables hold the dimensions of the game area.
- When the snake meets an enemy, it will end the game and then the user can restart again.
- The snake could kill an enemy when it is speeding up which costs the length of the body.
- Special apples help to make the snake have more abilities such as speeding up, the ability to kill the enemies and being invisible to the enemies.

1.2 Enhancements

We use the extra features like the creature of moving enemies, which the snake sees it as an obstacle to its survival. The player can kill the enemy to avoid being killed Another extra feature is the interface design, and different action has different sound effect during the game.

2. Requirements

Table 1 - Requirements

Req. No	Req. Name:	Req. Description	EMH	AMH	Prio.
1	Add a snake head	Create head and add picture	7hrs	8hrs	1
2	Make snake head move	Make it move in one direction	8hrs	10hrs	1
3	Make snake head controllable	Change the direction with keyboard	6hrs	7hrs	1
4	Add a body to snake head	Create body and add picture	8hrs	15hrs	2
5	Make the body parts longer	Increase the length of body	10hrs	12hrs	2
6	Add food	Create food and add picture	6hrs	5hrs	3
7	Make snake longer when eating food	Increase the length of body when eating food	8hrs	10hrs	4
8	Make food continuously	Make food appear one by one	6hrs	4hrs	5
9	Place more apples	Add more food at the same time	4hrs	3hrs	6
10	Add enemy	Create enemy and add picture	6hrs	8hrs	7
11	Restart the game when meet enemy	Remove all the body and restart the game	6hrs	4hrs	8
12	Make enemy move randomly	Make enemy move in randomly direction	4hrs	6hrs	9
13	Press space to speed up	Make snake move faster when pressing the space	10hrs	15hrs	10
14	Speed up at the cost of the length of body parts	Decrease the length of body when speeding up	8hrs	10hrs	10
15	Kill the enemy when the snake is speeding up	Remove the enemy met by the faster snake	10hrs	13hrs	11
16	add special food that have different effects	(1) speed up for a few seconds without losing body (2) won't die in few seconds	10hrs	8hrs	12
17	Make special food disappear after a few seconds	Remove special food when it is not eaten by snake in a few seconds	8hrs	6hrs	13
18	Add sounds	Add the sounds of each action	4hrs	3hrs	14
19	Improve the interface	Make the game more good-looking	4hrs	2hrs	14
20	Score board	Count the scores	8hrs	10hrs	15

Design and Implementation

3.1 Game Screen

The main class of the game. It is used to deal with the main function of the game. In this class, we use many variables to control different values, such as the state of game, the speed of the snake, the length of movement, the position and direction of the snake. Many Booleans are used to judge different conditions, such as whether the snake is dead, whether the snake has the abilities to speed up and kill enemies. The methods in this class control the whole concept of the game. These are used to check many different conditions during the game.

The Boolean “Speedup” and “disappear” are to deal with the S-apple and P-apple. If the snake eats a S-apple, the “Speedup” will be true, and the method SpeedUp() will do the speeding up effect. So does the P-apple.

3.2 Body part

Body part class is one of the most important class in this game, which is used to handle the length of the snake body. An array is used in this class to deal with the body part. When the snake eats an apple, a new body will be added to the array and will be showed on the screen. The UpdateBodyPosition() will always update the position of the body parts, which makes the body move following the snake head.

3.3 Apple

Apple class deals with the apple in the game. It includes the position and time of the apple, which decides when and where an apple should be placed. The TIME and timer decides how long the apples will appear on the screen.

3.4 Enemy

This is used to handle the enemies in the game. It includes the position, time, and direction of the enemies, which decides when and where an enemy should be placed and which direction an enemy should move.

The Fig.2 and Fig.3 below are the UML diagram of this game

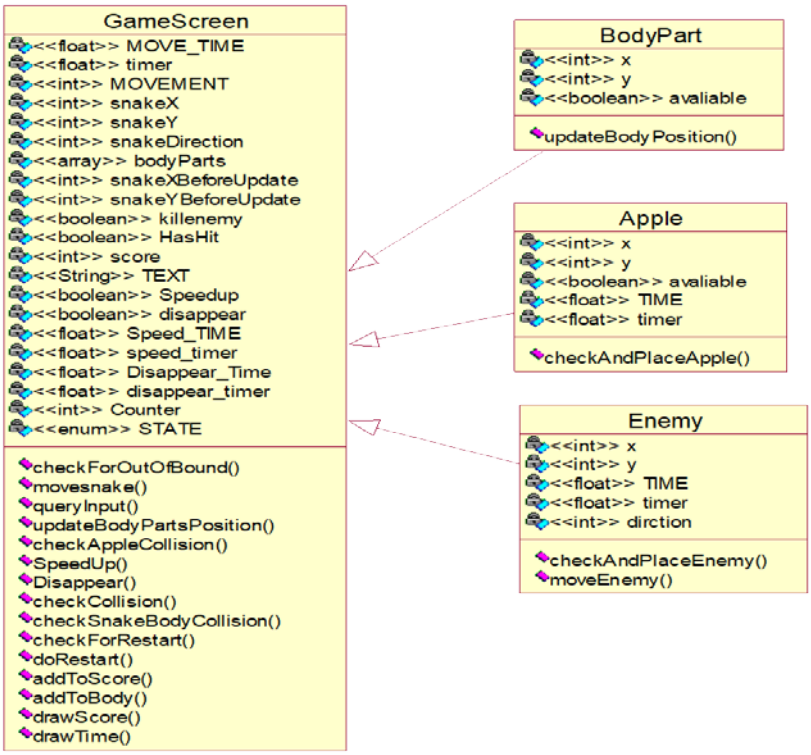

Fig.2: The UML diagram

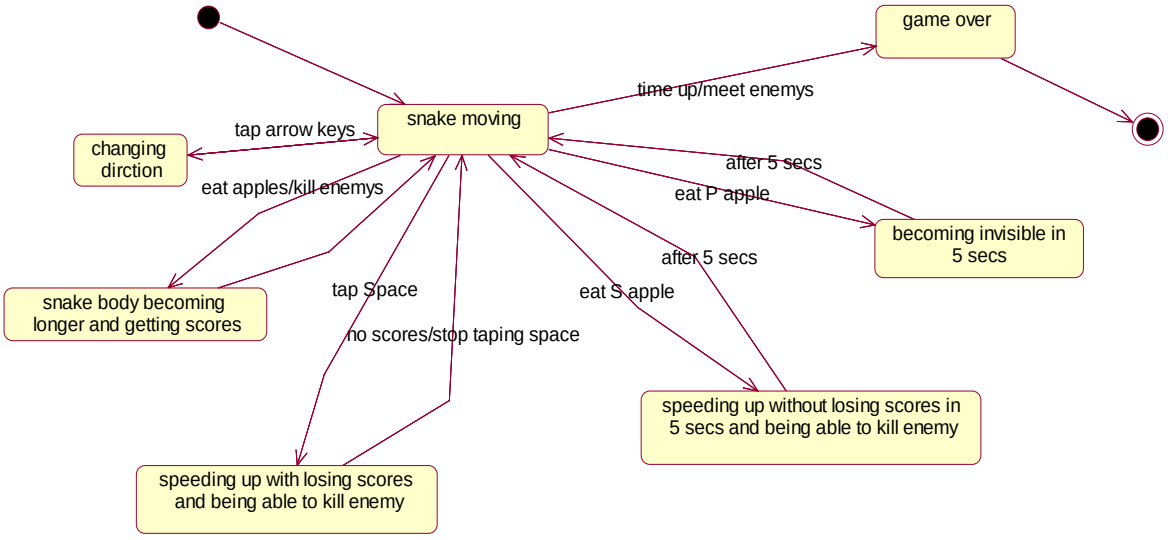

Fig.3: The UML statechart diagram

4. Test Results

Table 2 below contains the current status of implemented and tested requirements.

Table 2 - Test Results

Req. No	Req. Name:	Test Result
1	Add a snake head	PASSED
2	Make snake head move	PASSED
3	Make snake head controllable	PASSED
4	Add a body to snake head	PASSED
5	Make the body parts longer	PASSED
6	Add food	PASSED
7	Make snake longer when eating food	PASSED
8	Make food continuously	PASSED
9	Place more apples	PASSED
10	Add enemy	PASSED
11	Restart the game when meet enemy	PASSED
12	Make enemy move randomly	PASSED
13	Press space to speed up	PASSED
14	Speed up at the cost of the length of body parts	PASSED
15	Kill the enemy when the snake is speeding up	PASSED
16	add special food that have different effects	PASSED
17	Make special food disappear after a few seconds	PASSED
18	Add sounds	PASSED
19	Improve the interface	PASSED
20	Score Board	PASSED

5. Summary and Conclusion

5.1 Weekly Progress

5.1.1 Week 1

Here, we were more concern with the introduction of our game, and to see which concept of the snake game will suit the requirements which could work on. We develop a plan and decided to create some features of the snake game and how it could be implemented to corresponds to what we wanted and what we desired.

5.1.2 Week 2

In this week, our plan manifested as we all agreed and started with some possible codes, which accelerated and progressed in the game. We could produce a worm like snake head and make it possible to resemble a snake creature, as a result of this stage our ideas grow faster and we continue with the codes until a snake head was develop.

5.1.3 Week 3

For this week, we rebuild our frame and fix some bugs, all our members learn more knowledge about our game. Based on what we do last week, we add more requirements into the game and make the

game more interesting. And last week, we make it possible that, the game cannot end until you quit the window. Now we add enemies and game will be over if the player knocks on it by accident. It's the most important step what we did this week.

5.1.4 Week 4

Week 4, we have re-arranged the whole concept of our game and add some requirements which we make the game more interesting and understanding. The frame has been built with some fix bugs of all our members in the group, we add more apples food to the snake, and enemies in respective positions. We have also summaries the game with a UML class diagram to illustrate the implementation in a way that, a java programmer will understand, and easy to learn the basic techniques and how it was assigned with the various codes to identify the concept of the game.

5.1.5 Week 5

Week 5, The modifying of the UML Class diagram for our game, and we have completed the various requirements for the week. Speeding up of the game by pressing the shift key and, the background to make the game more and more beautiful, sound in the background a very good bright color. This is to make the game more attractive and interesting to people who may have interest after watching the game. Special food has also been added to make the snake gets energy to withstand its enemies, and the food appears and disappear at a regular interval making it very easy for anyone who may be of interest to work with such a game of this type. In addition to our final week requirements, we also produced a very good interface that suit the background of the desktop and attract the interest of people who could try to play the game after watching this kind of snake game. The last week of this meeting and it surrounding requirements has educate us in a very pleasant manner and making us to develop a good mindset, by encouraging us to further a better path ahead to be able to gathered more skills, ability, and knowledge wherever our capabilities, and capacity may enable us.

5.2 Difficulties and challenges

5.2.1 Adding the snake head

Adding the snake head is our first priority, and this requirement is quite difficult for us at the moment, because we don't have enough knowledge about the libgdx at that time. After reading some reference books and learning from the internet, we succeed in putting a snake head on the screen.

5.2.2 Making the body parts move follow the head

How to make the body parts move following the head is another difficulty. We could make only one body move following the head at first, and we discussed this problem for a long time. Finally, we solved this problem by the method UpdateBodyPosition(), which will always use the former position of head or the body, and use it as the new position as the later body. This method will always make all the body parts move following the head.

5.2.3 The collisions

There are many collisions in the game such as the snake head and apples, snake body and apples, snake body and enemies, and so on. To deal with this problem, we designed different methods to handle different situation.

5.2.4 Putting text on the screen

We don't know how to put text on the screen at first. After studying from the internet, we found we could use the Bitmap to show the text.

5.3 Correctness of time estimates

Correctness of time estimates is what we considered very important in our project, because if the timing is too less or too more it will affect the efficiency of the project. From week one to two, time estimates were not really good because we often estimated a short time for a requirement, but actually we used much more time than the time we estimated. As we proceeded, every member had more understanding about what we should do with the various assigned requirements and responsibilities, the time estimates became more and more accurate.

5.4 Priority decisions

Our priority, at first was to determine the basically structure of our snake game, it's to make the model of the snake and then consider about how to control the snake and make the game completely. We developed a snake head with the java libgdx and continued with the coding until we finally started to produce the body parts, making it possible for the snake to have food and add enemies to make the game more challenging. The randomly food and special function of the special apples is an idea we had after we already carried out the basically model of our game. And then what should be considered is how to make the game more attractively, and the background music and score table had been produced. It's just like a big tree, firstly we make the trunk of the tree and after that we add the branches and leaves ,so that let it become a "tree"..

5.5 Conclusion

Successfully, we basically met up with the goal of our wish and the necessary challenges and difficulties were overcome with what we were meant to do and through the process, we also achieved some goals that we didn't imagine to at the beginning. From these weeks work, we have already seen how our game will look, a basically completely snake game which meets the tentative plan we made at the beginning have been achieved. We have met a lot of difficulties through the process but fortunately all of the problems were solved. Everyone try our best to make our game more attractively and debug. All of us had no experience in making a game and these problems were like our teachers let us make a better software. Hopefully everyone can enjoy our game.

6 References

[1] Wikipedia: Snake (video game). (Available: <https://en.wikipedia.org/wiki/Snake_(video_game)>)

[2] James Cook (2015). LibGDX Game Development By Example, (Available: <https://www.geekbooks.me/book/view/libgdx-game-development-by-example>)

[3] Kilian Koeltzsch (2015). Game Development Newbie & Debugging Questions, (Available: <www.java-gaming.org>)

[4] Goran Locher (2015).Android LibGDX Game Development Masterclass, (Available: <https://www.udemy.com/libgdx-game-development-masterclass/?locale=de_DE>)