==================================================================

Student Name：彭敬樺 Assignment：3

Student ID：0416106

Student Email: wilbert.phen@gmail.com

**Claim: I worked on my own. ( YES / NO )**

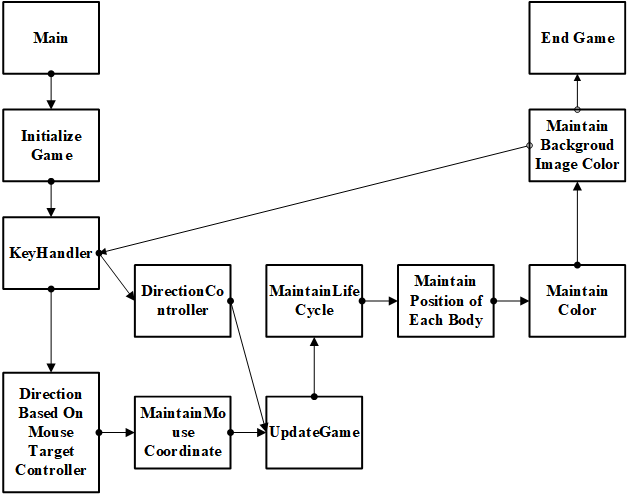
**Introduction**

Word Count: 115 words

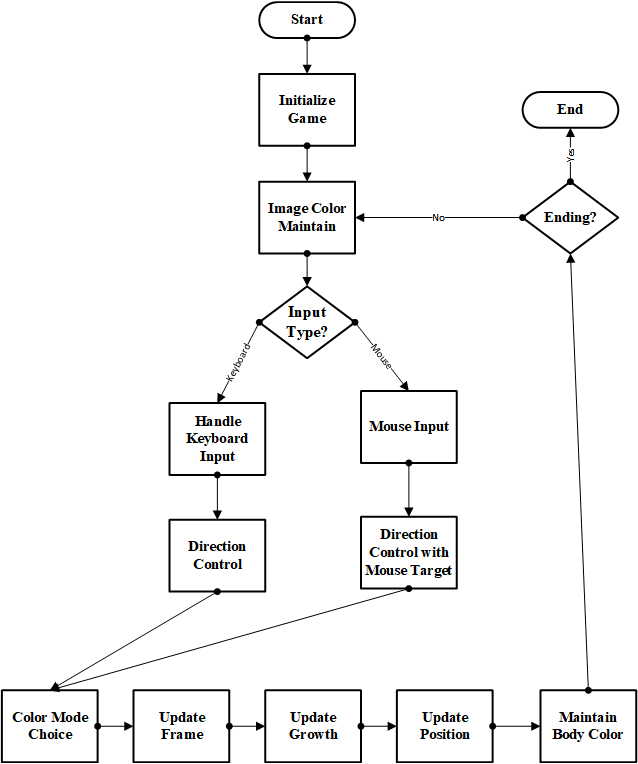
Assignment 3 consists of program to simulate the snake game. In the game, user will be able to control the snake directly by pressing key on the keyboard and many more key to represent the function implemented in the program. The default maximum number of snake’s body length is 1024.

Information of the author will be displayed before and after the gameplay. A message box will be used as pop up to display some parts of the available instructions. When the gameplay starts, user will see two separate windows, one is used to display status and value of the wanted information, the other window is the main display to play the game and interact directly.

**Structure Chart**

****

**Flowchart**

****

**System Architecture**

Word Count: 126 words

This program consists of 53 procedures inside the assembly main code. The C main program to execute and retrieve wanted information will call most of the written procedure. Many procedures are actually parts of bigger procedure that combine the result with the other procedure or even to assist its own calculation.

A few important procedure that will be called by the C main program is asm\_getStudentInfoString, AskForInputInitialization, asm\_HandleKey, initGame, updateGame, asm\_computeObjPositionX, asm\_computeObjPositionY, asm\_getObjectColor, asm\_SetImageInfo, asm\_GetImageColour, and asm\_GetNumofObjects. The function provided by each function is quite literally as the description written as its function name.

Due to the huge number of line and procedure number, procedure naming is set to be familiar and easy to guess its purpose so that to reduce confusion in the coding process.

**The Approach**

Word Count: 352 words

Writing a simple game is not as easy as what the name stated. Using assembly language, program could become more complicated and entangled. To prevent and reduce abnormality in the result, the code is separated into smaller function that will start running from the main function.

Default value and initialization value for some crucial variable is first saved into the constant variable that is positioned on the top of the code. This way the resulting program will be easier to be changed, corrected or improved in the future.

Data declaration is next, in this part, most variable is declared as global to be shared throughout the function without the need to send it by value. This approach is hoped to have good benefit in the performance of the result.

The function “updateGame” is used to maintain position of every parts of the snake body by some constraint set or changed in the other responsible function.

The function “asm\_HandleKey” is used to call the function that should be running if certain key is pressed. The respecting function will change the value of variable to change its constraint or rule to update the game.

The function “initGame” is responsible of initializing the game before the starts of the game. Many variable needs to have appropriate value, such as the color of snake’s head and the given input from the user to set the value of speed and life cycle of the snake.

The function “asm\_getImageColour” will return the value of RGB to the caller as indication of color for the requested coordinate. The color will be extracted from the value of array initialized in the function “asm\_SetImageInfo”.

The function “saving\_all\_data” and “loading\_all\_data” is used to implement the save and load function if “v” and “l” are pressed respectively. This function will write or read the current value in the important variable into a temporary variable that is responsible for recovery of the data if load is called.

The function “reset\_all\_data” is then used to initialized all the data into the starting value so that the program could redraw the snake without causing strange behavior.

**Discussion/ Experiments**

Word Count: 209 words

In this assignment, I have learnt more about how object can be interpreted as moving on the screen. A moving object is just about a frame being shown rapidly to create the illusion that something is changing its position. The problem in implementing this assignment is to take how many times the snake needs to be updated for the information to be sent to the screen. If the 2 of them is not synchronized well, then strange phenomenon will happen such as teleporting snake, randomly colored snake’s head, etc.

In my experience, the most time consuming time of writing a program is searching for bug. It is better to think thoroughly how to implement the wanted result before the coding process begun rather than fixing the problem after it has happened.

The next problem is that the snake coordinate and mouse coordinate is not entirely the same. They are connected through the value of Scale Factor provided in the program. Initially there are confusing of why the snake will not try the mouse in the clicked coordinate, when I finally realized that the range value for them are not the same after printing the value of the mouse position and snake’s head position and compare it side by side.

**Conclusion**

Word Count: 141 words

This assignment provides a good way to memorize and know more detailed function of each instruction in assembly language. By separating the usage and calculation that is different into different procedure will be beneficial for the progress of program making. Many procedure can be called and re-used many times in the program. By creating a more universal usage procedure, this can simplify the other parts of the program.

Next thing to pay attention to is the byte number for each data type and the leap of byte number needed to reach wanted data. Mistake in calculating the wanted byte or address can be fatal because this mistake is hard to be debugged, especially within a big or long program.

Some function can implicitly alter the value of register. Therefore, understanding every parts of the instruction and register can be very crucial.