**Assembly language and system programming**

Assignment \_\_\_\_02\_\_\_\_

Student Name:\_\_\_\_\_\_\_\_\_\_\_\_楊家安\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: \_\_\_\_\_\_\_\_\_\_\_0516021\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student email address:\_\_\_\_\_\_\_\_\_\_gxxygxxy@gmail.com\_\_\_\_\_\_\_\_\_

**[10%] Introduction [ at least 100 words]**

**WORD COUNT:\_\_\_\_\_\_\_202\_\_\_\_\_\_ [ Must be filled or zero score]**

This is homework 02.

In the menu, there are five options in this program that the user can choose from. There are the details of each option.

The first option is to change the color of the ship. There are three colors the user can choose from: blue(1), green(2), and yellow(3). To choose the color, press the corresponding key under the color block. Press other key won’t get any response. By the way, the default color is yellow.

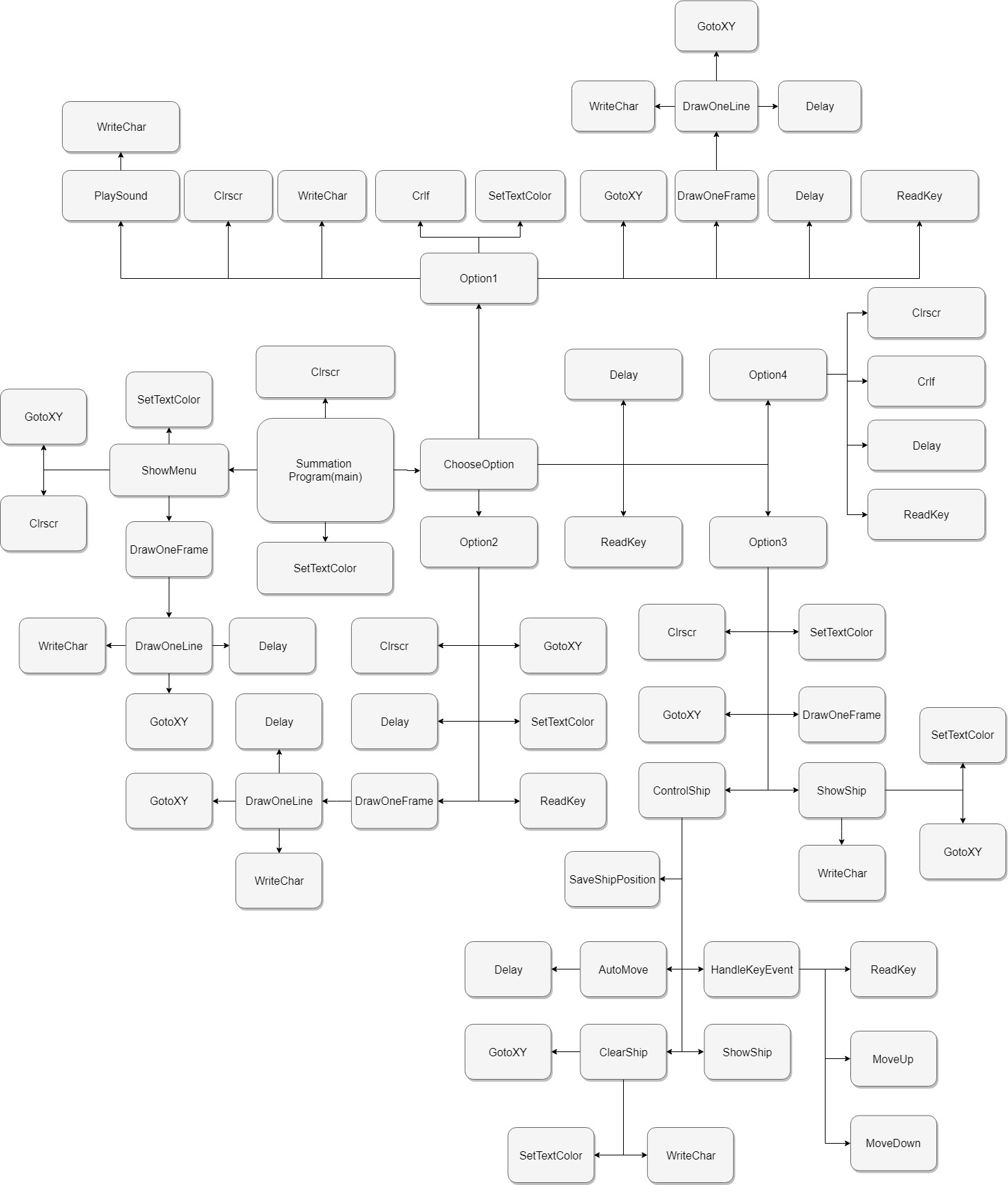
The second option is to show a frame in counter-clockwise starting from the upper left corner, which is in ship’s color. The delay in printing the char is 50 msec. After showing the frame, press any key to return to the menu.

The third option is to play a game. There is a frame whose color is the color of the spaceship. The spaceship will move automatically from left to right, and when it reaches the right side, it will return to the left side. The user can move the spaceship with ‘e’ and ‘c’ upward and downward, respectively. Press spacebar to go back to the menu.

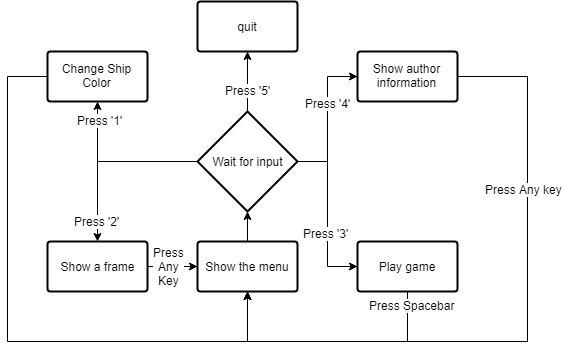
The fourth option is to show the author information.

The fifth option is to quit the program.

**[10%] System Chart** [at least 10 components]



[**10%] Flow Chart [**at least 10 components**]**



**[10%] System Architecture** [**at least 100 words]**

**WORD COUNT:\_\_\_\_\_\_\_180\_\_\_\_\_\_ [ Must be filled or zero score]**

There are some self-defined procedure in my program: ShowMenu, ChooseOption, Option1, Option2, Option3, Option4, DrawOneFrame, DrawOneLine, ShowShip, ClearShip, ControlShip, HandleKeyEvent, AutoMove, SaveShipPosition, MoveUp, MoveDown, PlaySound.

In the main, there are ShowMenu, ChooseOption, SetTextColor, Clrscr.

In the ShowMenu, there are Clrscr, GotoXY, SetTextColor, DrawOneFrame.

In the ChooseOption, there are Delay, ReadKey, Option1, Option2, Option3, Option4.

In the Option1, there are Clrscr, Writechar, Crlf, SetTextColor, GotoXY, DrawOneFrame, Delay, ReadKey, PlaySound.

In the Option2, there are Clrscr, GotoXY, SetTextColor, DrawOneFrame, Delay, ReadKey.

In the Option3, there are Clrscr, SetTextColor, GotoXY, DrawOneFrame, ShowShip, ControlShip.

In the Option4, there are Clrscr, Crlf, Delay, ReadKey.

In the DrawOneFrame, there is DrawOneLine.

In the DrawOneLine, there are WriteChar, GotoXY, Delay.

In the ShowShip, there are SetTextColor, GotoXY, WriteChar.

In the ClearShip, there are GotoXY, SetTextColor, WriteChar.

In the ControlShip, there are SaveShipPosition, AutoMove, HandleKeyEvent, ClearShip, ShowShip.

In the HandleKeyEvent, there are ReadKey, MoveUp, MoveDown.

In the AutoMove, there is Delay.

In the SaveShipPosition, there are no procedure.

In the MoveUp, there are no procedure.

In the MoveDown, there are no procedure.

In the PlaySound, there is WriteChar.

**[30%] The approach [ at least 300 words]**

**WORD COUNT:\_\_\_\_\_\_\_375\_\_\_\_\_\_ [ Must be filled or zero score]**

In the main, I first call the procedure ShowMenu to show the menu by mWrite. Then I call the procedure ChooseOption, to wait for the input of the option by ReadKey and Delay. After the input, according to the input to jump to the compatible label to show the detail of the option, by using cmp and je.

Option one:

First, to make all the messages well-aligned at middle top of the screen, I use lots of GotoXY and some spaces, and the UI looks friendly. Second, to show three rectangles(4x2) with different colors, I draw it by using DrawOneFrame easily, just adjust the width(4) and height(2). Three different rectangles are done in the same way, the only different part is the set the different color by SetTextColor. Third, use ReadKey, cmp and je to choose the corresponding color, meanwhile, save the color in variable ShipColor. By the way, to play the sound, I move 7(in ASCII, it means bell) into al and WriteChar.

Option two:

To draw the frame in counter-clockwise manner, starting from the upper left corner, just change the sequence of DrawOneLine in DrawOneFrame. I also save the ship color in variable ShipColor, and it’s drawn in ShipColor. To add the delay, there is a variable DrawDelay in procedure DrawOneLine, so it’s easily to control the delay. If I don’t want the delay, just change DrawDelay to 0 easily.

Option three:

Biggest part in the homework. The approach to drawing the frame is similar to the option two, just change DrawDelay to 0. To move to spaceship from left to right automatically, there is a procedure AutoMove, just increase the X-coordinate of the spaceship, and if it reaches the right side(ShipX=79), move the spaceship to the left side(ShipX=1) directly. To control the spaceship with ‘e’ and ‘c’, call HandleKeyEvent; in the procedure, there is ReadKey and cmp; to move the spaceship, just increase or decrease the ShipY, and if the spaceship reaches the top or bottom bound(ShipY=1 or ShipY=23), set ShipY to another bound position directly. Press Spacebar to go back to the menu by Readkey, cmp and je.

Option four:

Just use mWrite to show the message and Crlf to carriage return line feed.

Option five:

Directly use the instruction exit.

**[20%] Discussion/Experiments [ at least 200 words]**

**WORD COUNT:\_\_\_\_\_\_\_262\_\_\_\_\_\_ [ Must be filled or zero score]**

The process of doing homework 02 is smoother than doing homework 01. I think it’s because I have been familiar with the basic x86 assembly language grammar through the homework 01. Also, there are something similar to homework 01, such as choosing the option and drawing a frame.

Option one is easily to solve, just use a variable to store the current color of the spaceship.

Option two is also easy by changing the sequence of DrawOneLine in DrawOneFrame.

Option three is the new thing different from homework 01. That is, to move the spaceship. There is some skills in it. To make the spaceship looks moving, we have to clean the old position of the spaceship and draw the new position of the spaceship. To clean the old position, we must store the coordinate of it, that is, variable ShipX\_Old and ShipY\_Old.

Besides, there are several parts which have to draw the frame, such as showing the menu, option two and option three. Especially the option two, there must be a delay while drawing. But the other part of drawing the frame doesn’t need a delay. Originally I write two version of DrawOneFrame, one owns the delay, and the other doesn’t. That’s a direct easy way but I feel it’s not good. Later I think for a while if it’s possible to combine them in one version. That’s easy! Just declare a variable DrawDelay in it, and if we don’t need the delay, set DrawDelay to zero. It seems work. Then I think the procedure Delay can handle the 0 msec.

**[10%] Conclusion [ at least 100 words]**

**WORD COUNT:\_\_\_\_\_\_\_179\_\_\_\_\_\_ [ Must be filled or zero score]**

I have learnt how to move an object in x86 assembly, and use the key to control it. This is a very basic step in making a game. Almost every 2D game need to move the character with the keyboard on the screen, e.g., PAC-MAN, or a spaceship shooting game. And the UI of this homework is more beautiful than the last one, it really gives a looking of a game, not just a DOS view.

The homework makes me recall that I have made a small spaceship shooting game(I think spaceship game is always a good practicing for coding games) with Unity3D by C# about a year ago, and actually, at that time I refer to many resources on the Net, so I think the game I made before isn’t done on my own. It’s just a memory.

This homework is really a good practicing in coding x86 assembly. It is not hard, but it’s not very easy, either. Comparing to homework one, this homework is much easier. But I think my ability in coding could be faster.