Student Name: 李佳燕

Student ID: 0416039

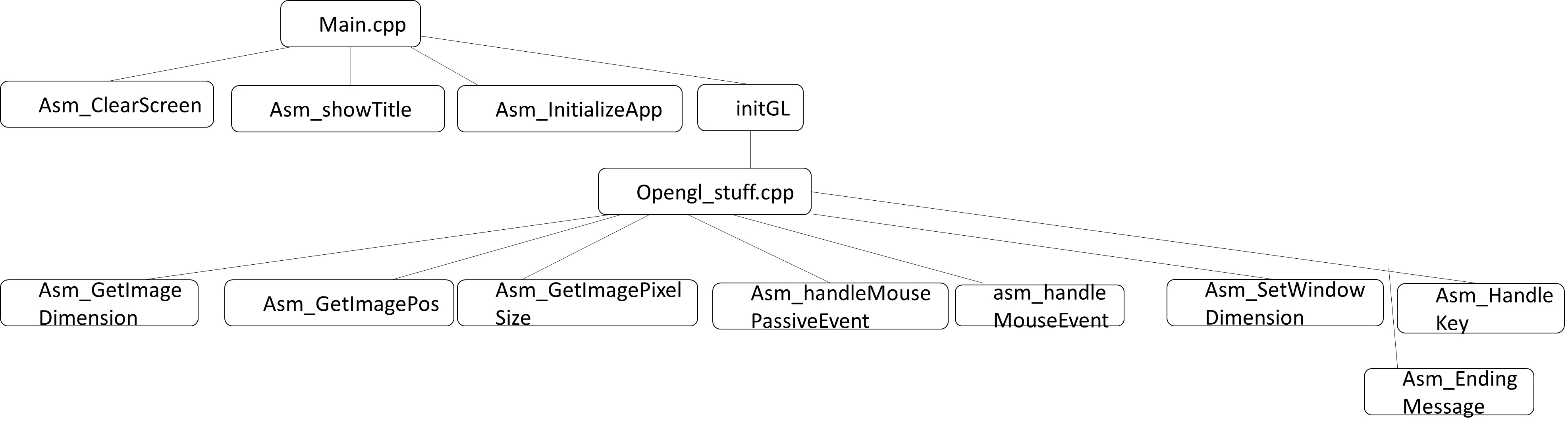
Student email address: girlannie123@gmail.com

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

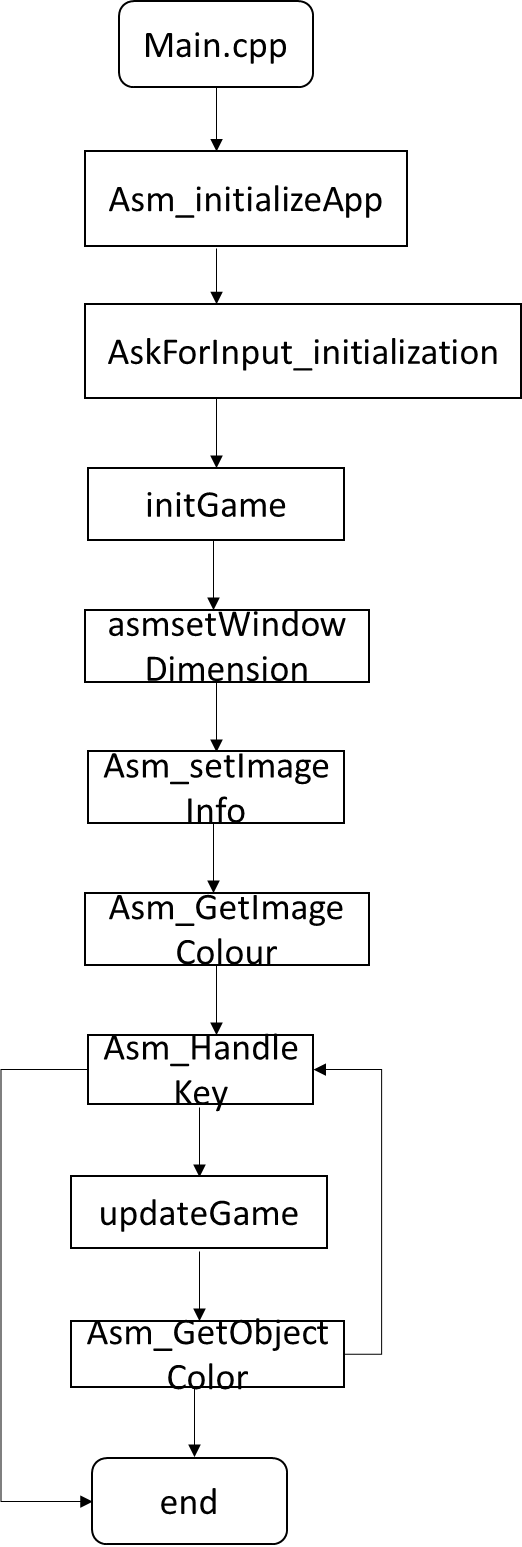
**WORD COUNT:\_\_\_\_\_\_110\_\_\_\_\_\_\_ [ Must be filled or zero score]**

The program is moving a lot of dots which is linking to a snake. User can input the space of the dots and the speed of moving first. Then during the game, user can change the direction, change the color to random or rainbow, or clear the dots. User also can ask the program that whether the program produce more dots or not. When user clear the dots, there is only one dots which is also the head of the dots. Also, user can store the status of the dots and load them into the program. There is also a picture in the background.

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



**[10%] Flow Chart**



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

**WORD COUNT:\_\_\_\_\_\_167\_\_\_\_\_\_\_ [ Must be filled or zero score]**

After showing the information, user has to input the space of the dots and the speed of moving. If user didn’t input anything, the default space is 25 and the default speed is 100. Then the program will enter the game part. Enter ‘w’, ‘a’, ’s’, ’d’ to move up, left, down, right. The default color of dots is red. Enter ‘r’ the change the color into random expect of head dot. Enter ‘p’ to change the color into the rainbow color. Enter ‘c’ to clear the dots expect the first one. Enter ‘v’ to save the status of the dots. Enter ’l’ to load the dots, which means change the status of the dots into the status we save lest time. User can also click the left mouse button to change the direction. Enter space bar to determine whether the dots have to grow or not. When the dots hits a boundary, the movement will change the direction. There is also the picture in the background.

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

**WORD COUNT:\_\_\_\_\_\_\_652\_\_\_\_\_\_ [ Must be filled or zero score]**

First, show my information at the title bar of the window, and show my information and the key usage instruction in a dialogue box. Just put the string we want to show in the dialogue into edx and put the string we want to show at the title bar of the window into ebx and call MsgBox can solve the problem. Then after show my information at the beginning in the console window, print a message to ask user input the speed and life cycle of the snake. Then start the game. Change the code in the asm\_handleKey to handle the dots. Use cmp to compare which command user input. Then call the function. If user enter ‘a’, ‘d’, ‘w’, ‘s’, call moveLeft, moveRight, moveUp, moveDown respectively. The four function is use to change the moveDirection. When user enter ‘p’, change the rainbow key to 1 and random key to 0. When user enter ‘r’, change the random key to 1 and rainbow key to 0. When user enter ‘c’, change clear key to 1. When user enter ‘v’, call save function, which is put all the status about dots into another parameters, When user enter ‘l’, call load function, which is put all the status which is in the other parameters into the dots status. When user enter space bar, change the space key (if space == 1, change to 0, else, change to 1). When user enter ESC, change quit key to 1, which is then show endingMessage and quit the game. When user push left mouse button, asm\_handleMouseEvent will change the mouse key into 1 and save the end point of X and Y into X\_m and Y\_m. Then go to the updateGame function. First, determined whether the dots is in mouse event or not. If it is in the mouse event, compare if first dot of X is at X\_m or not first, if it is not, move left/right to get the position. Then compare if the first dot of Y is at Y\_m or not, if it is not, move up/down to get the position. If both X and Y is at X\_m and Y\_m, change mouse key into 0. If it is not in mouse event, determined what is moveDirection is. If it is left, then sub the first dots of X. If it is right, add it, and so on. If the clear key is equal to 1, move numObjects into 1 and space key into 0. Then compare timer and grow of cycle. If equal, move timer to 0 and compare if numObjects is in MaxNumObjects or not. If not and space bar is equal to 1, give a random objColor into numObjects position of ObjColor array ( call Random32) and increase timer 1. If numObjects is equal to 1, move red color into ObjColor. Whatever the clear key, space key, mouse key is equal to 1 or not, we both have to change the position except the first dot. Just move the position of X, Y into the next dots position can solve the problem. Then write asm\_getObjectColor function. Determined whether the rainbow key equal to 1 or not. If not equal, get the position objID of ObjColor array. Give the value into r, increase 4, and give the value into g, and so on. If rainbow key is equal to 1, do the division on ObjID and get the remainder, which can determine which color we want to give r, g, b ( get the color from rainbow\_color array ). Then write asm\_SetImageInfo function. Just give the value into parameters which is relative to image respectively. Then write asm\_GetImageColour function. Get the position in the mimageptr0/mimageptr1 array (use imageIndex to determined which image we want to show). Then give the value to r, g, b. Finally, write asm\_GetImageDimension function. Just give mimagewidth, mimageheight into iw and ih and we finish the assignment.

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

**WORD COUNT:\_\_\_\_\_\_220\_\_\_\_\_\_\_ [ Must be filled or zero score]**

The most difficult part is at the first step. It is hard to know what is the program means. I try to ask lots of people to know what function is at which part and try to know how every part of the program works, then I know how to do it. But because I never write a program which can move or change pattern before, I spend some times to think how to change color, insert a image or move a snake which is made by lots of dots. Thanks for teacher taught how to write during the class, so I can write a little bit quicker. But it is still have some problem I met when I was write the assignment. The biggest program is that I forgot what is USES means so that I spends lots of time debug until my friends told me I have to put register into USES to avoid changing the answer in the register. It is the biggest problem but I though it is the most stupid problem. And other program I met just because I am not so familiar with assembling programming so that I ignore something that can cause a bug, but it is easy to debug. Finding something I ignore is much easier than finding some concept I forgot.

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

**WORD COUNT:\_\_\_\_\_\_115\_\_\_\_\_\_\_ [ Must be filled or zero score]**

This homework is much harder than before. Instead, it is more accomplishment then before, too. When I knew that I have to write a program that can show the dots and move dots, I was sharked and scared because I never wrote a program that can show the pattern before. But when I try to do this part, I found that it is not such difficult as I thought. And I was excited when I finish some part of the homework such as moving the dots, change the color, or insert a image. I found that this is not so difficult. Instead, it is an interesting homework. I found accomplishment and happiness inside the program!