FLABEE



B03901155 黃安

B03901156 黃于瑄

B03901188 宋玉美

Tuesday, December 30, 2014

# The Project

### Description

This project is a game which is very similar to “Flappy Bird” game, but this time we make a bee trying to fly through dozens of honeycomb. The principle is the same, the bee can’t touch neither the honeycombs or the top and bottom of the screen.

To explain the game more detail,

1. Project name = Flabee
2. Project type = Game, Windows Application
3. Window size = 800 x 480 pixel
4. Third-party Library = SDL 2.0.3
5. Description =

* When we open the game, an intro containing the game name; creator’s name; and a picture will appear for 5 seconds
* Then the start screen appears, with our bee flying in the middle of the screen and also a start button shows up there, ready to be clicked. There is also an instruction on how to play the game.
* By pressing the START button or the ENTER key, the game will eventually start with a background music
* Then by clicking on the screen or pressing the UP key, the bee will fly upward with a sound effect
* The score is still 0 (zero)
* You have to fly the bee and go through dozens of honeycomb, by going through 1 (one) set of honeycomb 1 (one) point will be added to your score, and you can see it directly on the middle top of the screen
* When the bee touches the honeycombs or top and bottom of the screen, the bee will die with a sound effect, and the game will end
* When the game ends, a Game Over screen will appear, by showing your score and also a START button to restart the game
* If you press the START button, then the game will restart with the score back to 0 (zero)
* The game will close if you close the window

### Function

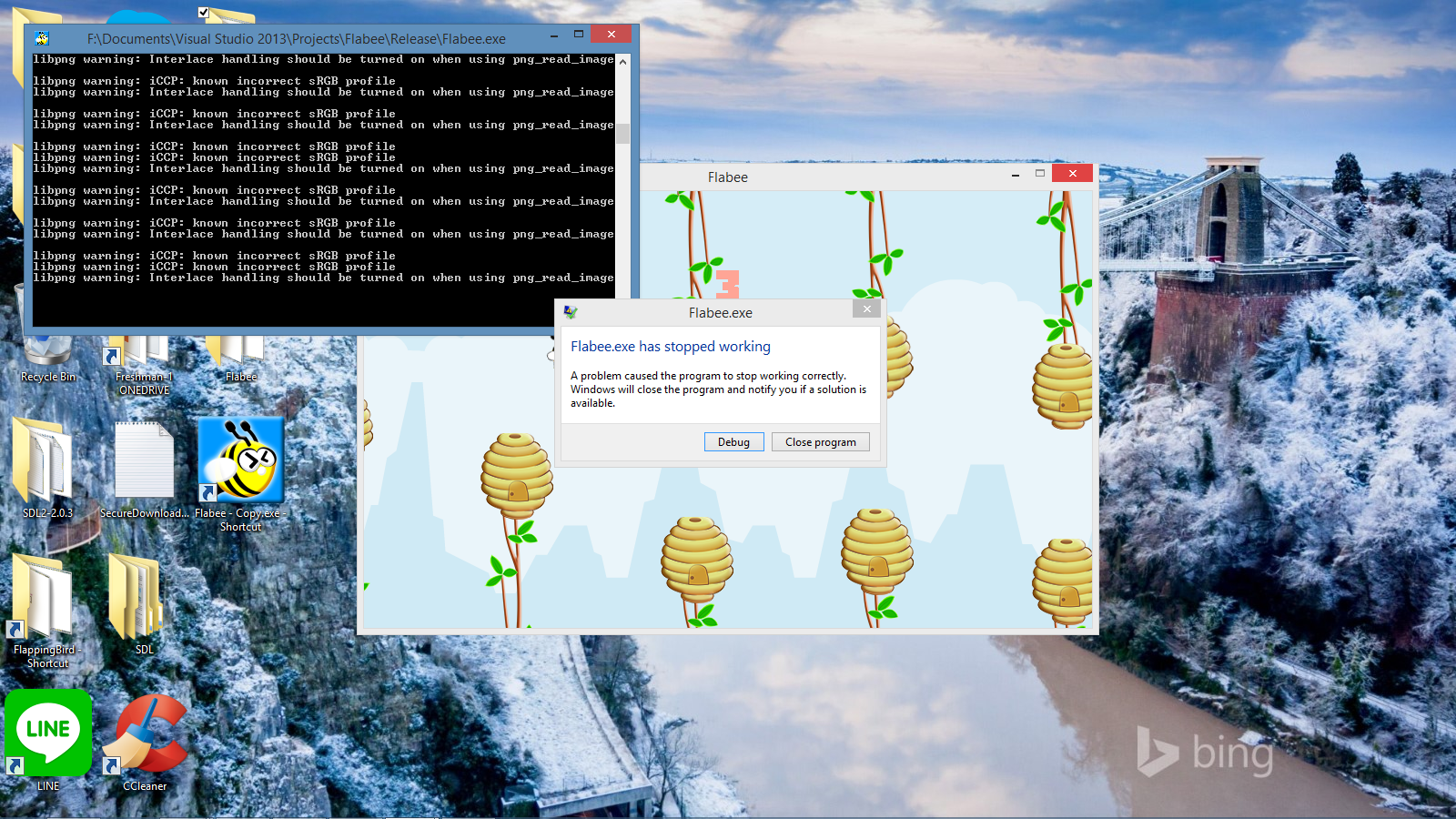
The game’s functions are :

* The game can be controlled with 2 ways, by clicking the mouse and pressing the keyboard
* The bee fly upward by clicking the mouse and by pressing the keyboard, but automatically fall downwards when the user does nothing
* The honeycombs appear with random heights
* The background is moves along with the game, causing the game feels more real
* The game is completed with background music and sound effects
* The game can count the score, but can’t save the score for the high scores
* The game can be restarted over and over again

### Problems

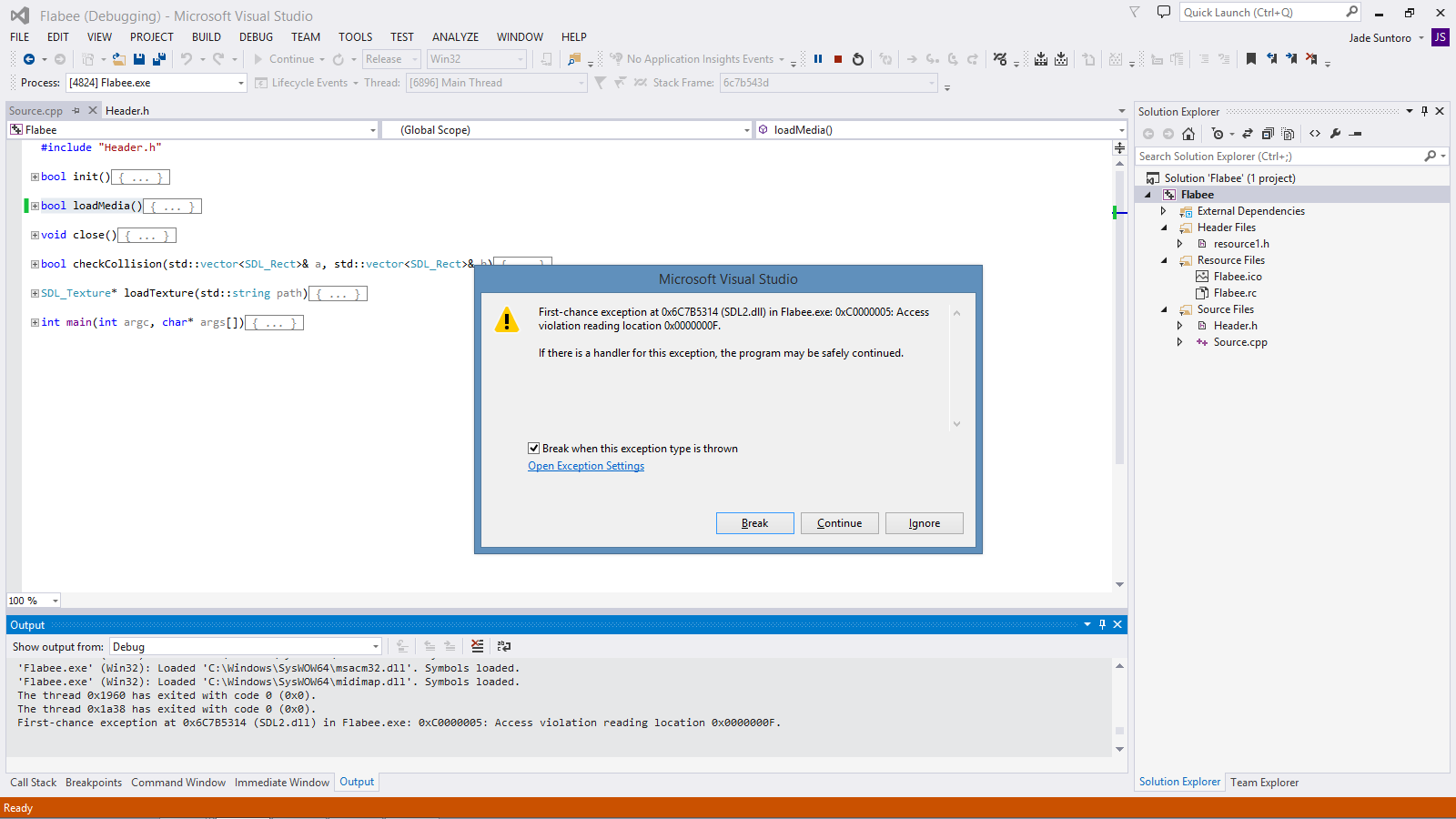
The problems when developing this project will be discussed personally by each person. Some of our main problems that we had as a group are :

* We are looking for some guidance to build a c++ game, as we had never done this before, we all are newbies. So we searched for information, and got a Flappy Bird clone on Github that was built with c++, but as soon we saw the code, we really didn’t understand least of it, it was because he used a library that was built by himself (SGE – Simple Game Engine Library) and he didn’t even write a how-to or documentation file, so it was really hard to understand his code. So we moved on, we searched for other game engine or libraries that can be used, and we decided to use the SDL (Simple DirectMedia Layer) 2.0.3 which has a lot more how-to and documentation file. And we started to write our kind of Flappy Bird, with SDL.
* And we think the biggest problem is, this…



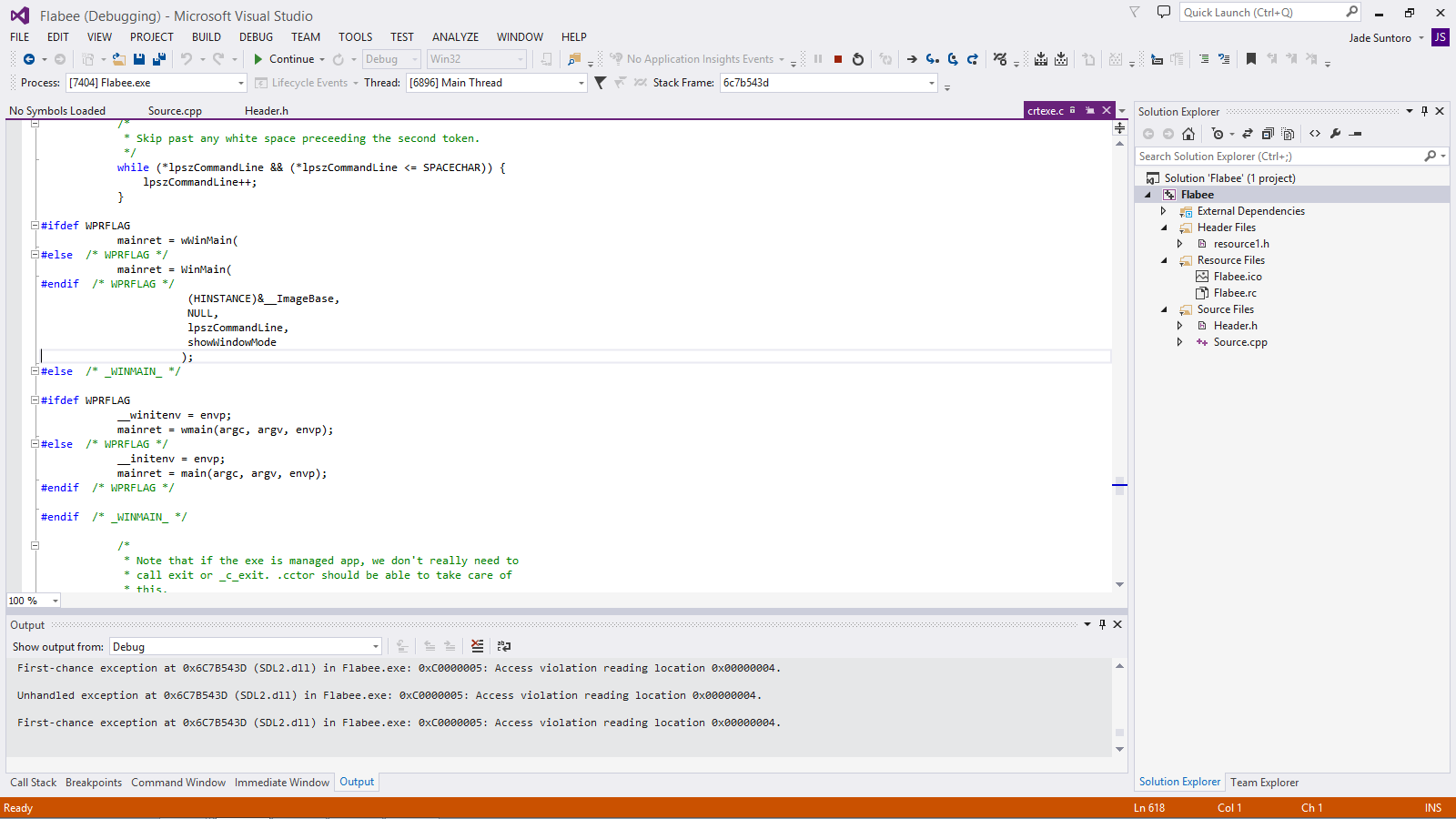
The program can run perfectly, but after some time, it unexpectedly stop working! We still haven’t figured the reason why it stops, while the other group have figured out the reason why is their game also stopped working. But, as we can see from the screenshot above, there is no single error printed in the console, it were only warnings, which appear because the RGB profile is incorrect and interlace handling should be turned on when using png\_read\_image. I have already searched this kind of warning on the internet, and it doesn’t seem to be the source of the error. Since the warning is already there all along while the application is running, so the sudden stop of the application does not cause the appearance of the warning.

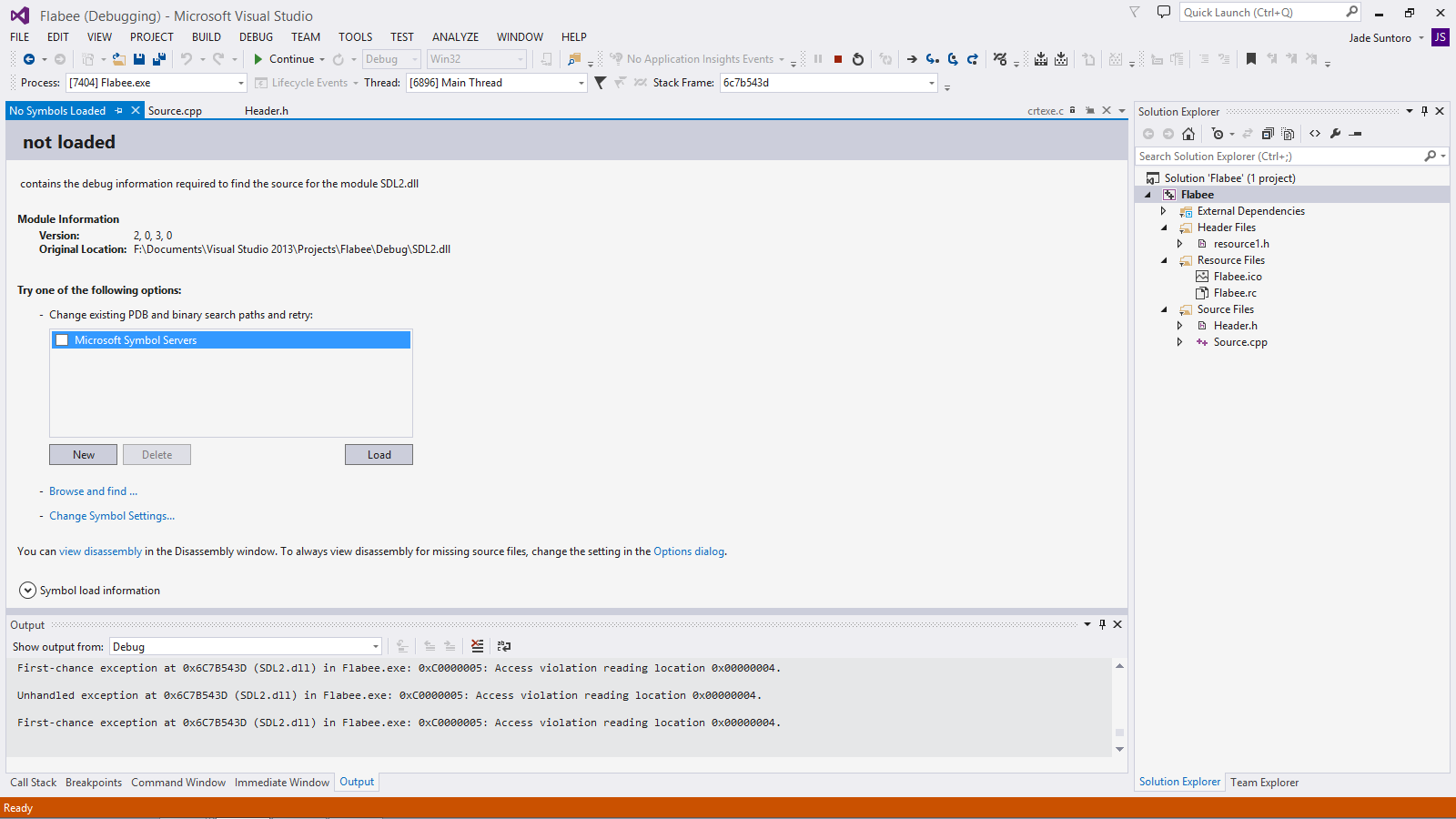
Then, if we choose to “Debug” the program, we will run into the Visual Studio and also got a message saying exactly like the picture below.



But the weird thing is the location of error is always changing and we also can’t figure out the reason. We did search for solutions on the internet and asked some upperclassmen, but they also couldn’t help. Even sometimes the search leads to nothing, as the problem / error could not be found on the internet.

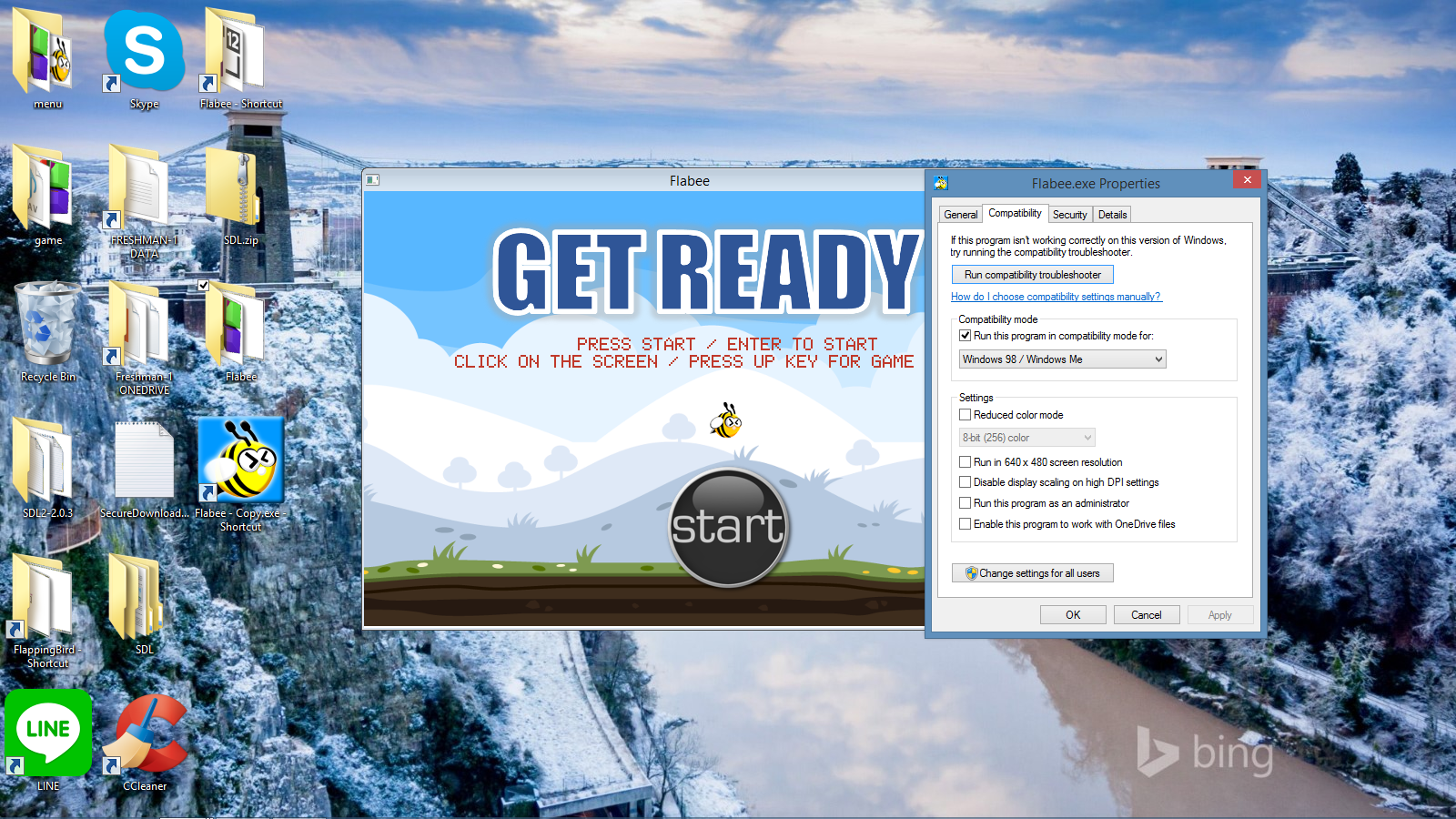
Then if we choose “Break” from the window above, we will jump to one of these picture below, sometimes it can be the first one, but sometimes it can also be the second one. So it really is confusing. And the problem still haven’t been solved.





But, we can still run the program without problems if we do some settings. Just like the picture below,

1. Right-click on the application
2. Choose ‘Properties’ or ‘內容’
3. Choose the ‘Compatibility’ or ‘相容性’ tab
4. On the ‘Compatibility mode’ or ‘相容性模式’ box, check the checkbox
5. Choose ‘Windows 98 / Windows ME’ from the dropdown menu
6. Run the application



Well this is not a solution, but it is the best we can do.

### SDL Library

Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including [Valve](http://valvesoftware.com/)'s award winning catalog and many [Humble Bundle](https://www.humblebundle.com/) games.

SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. Support for other platforms may be found in the source code. SDL is written in C, works natively with C++, and there are [bindings available](https://www.libsdl.org/languages.php) for several other languages, including C# and Python. SDL 2.0 is distributed under the [zlib license](http://www.gzip.org/zlib/zlib_license.html). This license allows you to use SDL freely in any software.

* Video
* 3D graphics:
  + SDL can be used in combination with the OpenGL API or Direct3D API for 3D graphics
  + Accelerated 2D render API:
  + Supports easy rotation, scaling and alpha blending, all accelerated using modern 3D APIs
  + Acceleration is supported using OpenGL and Direct3D, and there is a software fallback
  + Create and manage multiple windows
* Input Events
* Events and API functions provided for:
  + Application and window state changes
  + Mouse input
  + Keyboard input
  + Joystick and game controller input
  + Multi-touch gestures
* Each event can be enabled or disabled with [SDL\_EventState](https://wiki.libsdl.org/SDL_EventState)()
* Events are passed through a user-specified filter function before being posted to the internal event queue
* Thread-safe event queue
* Force Feedback
* Force feedback is supported under Windows, Mac OS X and Linux
* Audio
* Set audio playback of 8-bit and 16-bit audio, mono stereo or 5.1 surround sound, with optional conversion if the format is not supported by the hardware
* Audio runs independently in a separate thread, filled via a user callback mechanism
* Designed for custom software audio mixers, but SDL\_mixer provides a complete audio/music output library
* File I/O Abstraction
* General purpose abstraction for opening, reading and writing data
* Built-in support for files and memory
* Shared Object Support
* Load shared objects (DLL on Windows, .dylib on Mac OS X, .so on Linux)
* Lookup functions in shared objects
* Threads
* Simple thread creation API
* Simple thread local storage API
* Mutexes, semaphores and condition variables
* Atomic operations for lockless programming
* Timers
* Get the number of milliseconds elapsed
* Wait a specified number of milliseconds
* Create timers that run alongside your code in a separate thread
* Use high resolution counter for profiling
* CPU Feature Detection
* Query the number of CPUs
* Detect CPU features and supported instruction sets
* Endian Independence
* Detect the endianness of the current system
* Routines for fast swapping of data values
* Read and write data of a specified endianness
* Power Management
* Querying power management status

### Game Control

There are 2 ways to control the game, which are :

* + 1. Mouse
       - Click the START button to start the game
       - Click anywhere on the screen to make the bee fly upwards
    2. Keyboard
* Press the ENTER key to start the game
* Press the UP key to make the bee fly upwards

### Work Distribution

* Together
* Texture loading
* Load medias
* (with helps from the internet ☺)
* 黃安:
* Showing the honeycombs
* 黃于瑄:
* Showing the bee
* Sound effects and background music
* Mouse click
* Draws the bee, start button, game background, start background
* 宋玉美:
* Collision
* Score
* Game loop
* Keyboard press
* Draws the honeycomb, intro picture, game over background
* Combine and simplify the codes

### 個人心得

* 黃安:

原本我們這組是要用一個叫做Simple Game Engine的library來設計遊戲，但設計者沒有寫說明書跟使用方法那些，而我看內容也看不太懂…….，跟組員討論過之後，決定改用Simple DirectMedia Layer，有比較清楚的說明書，而且很容易在網路上找到使用方法、跟遇到問題時該如何解決。我負責的部分是作為阻礙的蜂巢的部分，一開始我設定的class只能產生出上半部或下半部，在只有一組蜂巢時還好，但一旦要製作出數組時就會變得有點複雜，程式碼變得很冗，同樣的東西要重複打出一大串看起來很累不夠簡潔，而且也多了要如何依照固定的間隔前進的問題，這時我才決定要把class裡包含上半部跟下半部的位置跟長寬等等，程式碼才變得簡單多了，然後再利用goto，當前一組蜂巢還沒跑到一定的距離時下一組就跳過不執行移動的方程式，這些東西看起來很簡單，卻花了我一些時間才想到，現在回想還覺得當時苦惱的自己好笨。最後能幫忙完成一個遊戲覺得很開心，而且我覺得畫面很可愛!!!(雖然老師好像不太注重這部分) 想說未來再來做做看，想做更複雜的RPG!!!!因為我不太擅長計程，英文也有點爛爛的，上網查說明也看不太懂，所以有很多地方都是靠組員幫忙的，負責遊戲控制的同學因為要整合大家寫好的程式碼，還要花時間看懂跟稍微修改很辛苦，真的很感謝他們跟我一起完成專題!!!!!

* 宋玉美:

At first, I was really confused that I did not know what to do, since my group is full of really busy people, including myself☹ and I does not even have any programming experiences. For the first 2 weeks, we did not do anything for this project, we were so busy that we forgot to do the project. To be honest, there was a time when I felt so upset because the other members of the group were so busy with their own things QQ. They did not even think about the project at all. I started to think that this project was kind of an impossible task. But I am still really excited to do this project, because I was thinking how proud and happy I will be if I get the project well done.

On the 3rd week, we finally decided to focus on the project and tried to distribute the work evenly. However, because of each member’s activities, and I think I was the one with more time available to do the project TT. I do not want to blame the others, but I felt a little burdened and it seemed like I was the only one who cared about the project. I even thought of doing this project on my own (which will be very difficult).

On the 4th week, we finally worked as a group, the other members really helped me so much and I felt really grateful. While doing the project, I think the most difficult part is to combine all of the codes and then trying to simplify the code as much as possible. Because we all have different styles to write codes, so first I had to understand all of their codes and tried to rewrite or just combined them.

When I was writing my own parts, which are the collision and scoring part, I had to open the picture of the bee and honeycomb and measure them with pixel, then trying to make a precise collision boxes. The collision part was not difficult, just like the scoring part. To write the scoring part, I need to think of a way so the score will not increase continuously, and I think it goes smoothly just like I planned. YEAY!

It took me two days to combine the codes so it becomes a complete application and can be run without a problem, I even did not sleep like how I supposed to, I always slept at 2 AM or more. It really were tiring days. However, it really meant a lot to me to see the project done, and we can also play the game. Even though the program is not 100% built with my own hands, but I am really happy. Well, it is just like what Neil Armstrong said, “One step for man, one leap for mankind.” I know, maybe for others this much is nothing, but for me this much, is just like a whole new world. A world full of codes, and I have to understand each of them to make it run properly.

But it really shook me that after some times the program stops running. I thought that I finally did it, but to think that another problem comes up, UGH, really don’t know what to do. I can’t even figure out the reason of the error. Although I debugged the program again and again, searched for solutions all over the places, but it still won’t work. I felt really sad, though. So right now, I am just hoping that 老師 will appreciate our project, as we already tried our best to do this project from zero. Still, I felt really happy to see the game finally done. HOORRAY! 謝謝老師, 謝謝組員!☺