9.3HD

COS10009

Introduction to programming

**Thanh Minh**

**Student ID: 103809048**

Code Explanation document

Swinburne University of Technology

1. **Program’s Window and implement some features**

A picture containing logo

Description automatically generated

Figure 1: set up the ZOrder to arrange the layers for later uses in the program

Hidden will be at the back of the program when I want to hide content and create the effects so that the program can have a better looking.

I have created the constant HEIGHT(800) and WIDTH(1500) for the size of the window, and put it on the top of the file so that everyone can see clearly.

Graphical user interface, text

Description automatically generated

Figure 2: set up for the window

If I change false into true, the program’s window will be played in full screen. I also create an array for Player for displaying ships in the Selecting page.

**Logo

Description automatically generated with medium confidence**

Figure 3: End code of program

Without these 2 lines of code, the program cannot run so I must include it in every single Gosu program.

Text

Description automatically generated

Figure 4: States and procedures of Update for program

Text

Description automatically generated

Figure 5: States and procedures of Draw for program

First I will have to set the @scene = :start so that the program will start at the entry page and will call all the procedures related to start such as draw\_start, update\_start, …

I must have each screen of the program, it should have different functions so that can increase the efficient of the code, make the program run smoother, faster. The code is also can be visualized easily.

1. **Entry page**

First, I need to set up in order to draw up the entry page for the program.

Text

Description automatically generated

Figure 6: Setup for the entry page

I need to set @scene = :start by default so that the program can start and draw the entry page first. I also nneed to import some sounds and musics into my program. I also define the font size for drawing title, description of the program so that the user can know what is on the screen.

For the @hover\_options\_1,2,3 , I will use it for creating effects while suffering through options so that the user can feel like the program is make with a lot of effort.

For two images that I have imported, start.jpg is for the background of the screen to decorate the screen. The other one is for pointing at the current option (there are some RPG using this function for their menu)

Graphical user interface

Description automatically generated

Figure 7: draw\_start procedure

After importing background image, I need to draw it so that it can display, but I will make the ZOrder of it is Background so that it won’t hide other contents.

I have made the ZOrder of @point and @hover\_options\_1,2,3 are hidden so that @point and black rectangle won’t display on the screen until their ZOrder are changed.

A picture containing text

Description automatically generated

Figure 8: update\_start procedure

In figure 8, I have set up for the effects when suffering through options, the black box and the pointing rocket will be displayed at the current option like in figure 9

Graphical user interface, website

Description automatically generated

Figure 9: Entry page

Text

Description automatically generated

Figure 10: button\_down\_start procedure

For this procedure, when I press the left mouse button or use Enter at the current option, it will direct me to the next page. When I choose the Play option, the start music will be played to announce me that I have entered the next stage of the game. And when I choose to close the program just choose the Exit option then the program will close immediately

1. **Instruction page**

Text

Description automatically generated

Figure 11: set up for the instruction page

Before moving forward, I need to import music and background image for this page.

Text

Description automatically generated

Figure 12: draw\_instruction for the instruction page

For this procedure, mainly I draw the background image and the description for the user to read how to play this game. So that the user can understand more about the program.

A picture containing text, outdoor, mountain, nature

Description automatically generated

Figure 13: Instruction page

Graphical user interface

Description automatically generated

Figure 14: Update\_instruction for instruction page

This page is the easy page to be created among all the pages so in the update\_instruction there is no code but if I have time to develop this project more, I will decorate more thing on this page and adding effects. Because there are some programs, games that in the instruction page, their page are so interactive and clearly show how the game work.

Text

Description automatically generated

Figure 15: button\_down\_instruction procedure for instruction page

When the user wants to exit and go back to the previous page, they can press Q or Space to go back.

1. **Selecting page**

At all the start of each page, I need to set up things for later use.

Text

Description automatically generated

Figure 16: Set up for Selecting page

After setting up, I also need to include the array for player for drawing it.



Figure 17: Add array for player

After importing the media files, I need to draw out what I have set up above in the draw\_selecting.

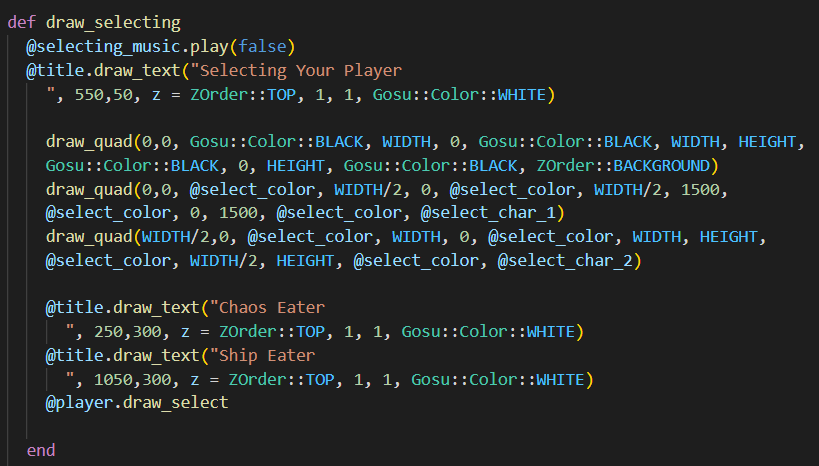


Figure 18: draw\_selecting procedure for my program

For the first quadratic shape, I made its color Black and the ZOrder is Background and for the next two quadratic shapes I made it hidden so that when the user is choosing the player, the current option shape will be go up with the color of grey.

With the @player.draw\_select, I will call the draw\_select procedure for the array Player in another file so that the player can be drawn out.

Text

Description automatically generated

Figure 19: Array for Player

Above is the array of player I have created in another file to keep the main file shorter and clear.

Graphical user interface, text

Description automatically generated

Figure 20: update\_selecting procedure

With those lines in the figure 20, the current option will be visualized for the user to see whenever they want to choose which player.

Graphical user interface, application

Description automatically generated

Figure 21: selecting page

The current option is chaos eater so the box on the left will be grey.

Text

Description automatically generated

Figure 22: button\_down\_selecting procedure

With this procedure in figure 22, the program will know what the next step is when the user has chosen the player. The new array will be created depend on which ship the user choose and it will call exact what the user has chosen.

1. **Play page**

Before going further, I first need to set up and import the musics and text’s font size. And I don’t forget to include the files related to this file. The reason why I use the Sample for shooting music is the sound won’t interrupt the music and create annoyed noise.

Text

Description automatically generated

Figure 23: initialize and import music

For the drawing parts, it is quite simple than others when I just need to draw out the essential things such as ship, bullets ,… more over I also want to count the scores of the players so I will use @count for counting the scores.

Text

Description automatically generated

Figure 24: draw\_playing procedures

Text

Description automatically generated

Figure 25: update\_playing procedure part 1

For the figure 25, whenever I move the user, I will call the procedure from other files so that the player can be moved smoothly. I also generate the enemy and using rand < FREQUENCY which means the frequency of creating enemy will be randomly chosen and that value will be smaller than constant FREQUENCY. I also create procedure for automatically delete the bullets and enemy when both of them leave the screen (go out the visible area)

Text

Description automatically generated

Figure 26: update\_playing procedures part 2

I also make the explosions after appearing, it will be deleted. When the bullets hit the enemy, 4 things are going to be happened. First and second, the enemy and bullets will be deleted. Third and fourth, the score will increase by 1 and the explosion animation will be displayed.

There are 2 cases to lead to the end of this game: first case, the user press escape to exit, second stage is when the user get hit by the enemy and depend on how the game is ended, the different messages will be shown.

Text

Description automatically generated

Figure 27: initialize\_playing procedure

The reason why at this step I start to create the empty array and @count is if I created it sooner, when the user reset the game, the scores and enemy won’t be reseted and the user’s player will keep die so that’s why at this stage I start to create the empty array.

Text

Description automatically generated

Figure 29: button\_down\_playing procedure

For this procedure, everytime I press space for left mouse button, or enter button, the ship will shoot, and the music will be played.

Text

Description automatically generated

Figure 30: Explosion array in a different file

Text

Description automatically generated

Figure 31: Enemy array in a different file

Text

Description automatically generated

Figure 32: bullet array in a different file

1. **Credit page**

First, I need to import the music for the credit stage and the font size.

Text

Description automatically generated

Figure 33: set up for credit page

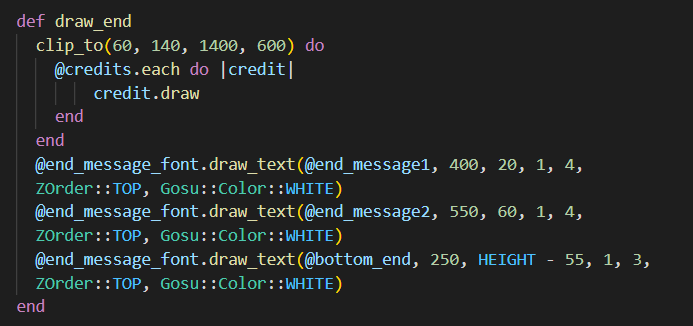


Figure 34: draw\_end procedure

For the figure 34, the displayed text will be looped to create animation for it, I need to draw a box with 600px height and 1400 px width. A long with that, the messages for the credit will be drawn.

Text

Description automatically generated

Figure 35: initialize\_end procedure

Depend on what the circumstance the first message will be different. And the current scores will also will be displayed along with the instructions to follow. the system will opened up the credits.txt to display the text from it and printing line by line.

Text

Description automatically generated

Figure 36: update\_end procedure

For this procedure, it help to scroll the text from the file and reset it so that it will keep displaying.

Text

Description automatically generated

Figure 37: button\_down\_end procedures

Base on what the user presses, the previous stage will be recall to display or can be closed immediately by pressing Q or Escape.

Text

Description automatically generated

Figure 38: Credit array in a different file