# Design Overview for <<Games Menu>>

Name: Trung Kien Nguyen

Student ID: 104053642

# Summary of Program

Describe what you want the program to do… one or two paragraphs.

Include a sketch of sample output to illustrate your idea.

The program that I intend to make is a Games Menu, which allow user to access and play about 4 different games, including Gems Collection (an upgraded version of Whack a Ruby), Star Wars (an upgraded version of Sector5), 2048 and Obstacles Overcoming (a version of Flappy Bird). Some games likes 2048 or Overcoming Obstacles will have 3 levels, including easy, hard and super hard.

1. In Gems Collecion
   1. Click on the diffrent gems to get different numbers of point.
   2. Click on the rocks, or on places other than gems and get 1 score minus.
   3. Earn 80 scores in 30 seconds to win the game.
2. Star Wars
   1. The game lasts for 30 seconds. You will have 5 lives. Earn 80 scores to win.
   2. Move Your ship with the Left, Right, Up and Down arrow keys (or A, D, W, S keys respectively).
   3. Shoot Enemy ships by pressing the Z key and earn 3 score for each.
   4. If your ship crashes into an enemy, your will lose 1 life. When your life is 0, you will lose the game. On the other hand, you will get an extra life if your ship gets one from space.
3. 2048
   1. Use the Left, Right, Up and Down arrow keys (or A, D, W and S keys respectively) to move the tiles.
   2. Tiles with the same number merge into one when they touch. Add them up to reach 2048!
4. Obstacles Overcoming (It will be similar to Flappy bird)
   1. Use the Space bar to jump, make sure you go through the bricks wall perfectly
   2. You will lose if your reach the ground, or crash in the wall
   3. If the playing time reaches 60 seconds, you win the game

References: “Learn Game Programming with Ruby - Bring your ideas to life with Gosu” (Mark Sobkowicz, foreword by Julian Rashke)

# Required Data Types

Table : <<record name>> details

In “games\_menu.rb”

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| @image\_gems\_collection | Gosu::Image | Demo Image for Gems Collection |
| @image\_star\_wars | Gosu::Image | Demo Image for Star Wars |
| @image\_2048 | Gosu::Image | Demo Image for 2048 |
| @image\_oo | Gosu::Image | Demo Image for Obstacles Overcoming |
| @song | Gosu::Song | Background song |
| @font | Gosu::Font | Texts |

In “gems\_collection.rb”

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| @gemmining, @question, @underground | Gosu::Image | Background Image for Opening, Instruction, playing game |
| @ruby, @danburite, @sapphire, @amethyst | Gosu::Image | To draw different gems that player should collect |
| @rock | Gosu::Image | To draw rock that player shouldnot collect |
| @start\_song, @game\_song, @end\_song | Gosu::Song | Background songs |
| @collect\_sound | Gosu::Sound | Sound effect for collecting |
| @font | Gosu::Font | Texts |
| @x\_i, @y\_ I, @v\_x\_i, @v\_y\_i (with i from 0 to 9) | Integer | Position, velocity of 10 different opjects in the game |
| @playing | Integer | Define the which screen should display:  0 means opening screen  -1 means instruction  1 means main playing screen  2 means ending screen |
| @time | Integer | Display the playing time |

In ”star\_wars.rb”

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| @space\_0, @question, @space\_1, @space\_2 | Gosu::Image | Background Image for Opening, Instruction, Playing game and Ending Screen |
| @spaceship, @enemy, @heart, @large\_enemy | Gosu::Image | To draw different objects in the game |
| @start\_song, @game\_song, @end\_song | Gosu::Song | Background songs |
| @shoot\_sound | Gosu::Sound | Sound effect while shooting |
| @font | Gosu::Font | Texts |
| @score | Integer | Display the score |
| @playing | Integer | Define the which screen should display:  0 means opening screen  -1 means instruction  1 means main playing screen  2 means ending screen |
| @time | Integer | Display the playing time |

*Records*

*Player*

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :x | Integer | Position of the main ship |
| :y | Integer |
| :angle | Float | The direction of the main ship |

*Bullet*

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :x | Integer | Position of the bullets |
| :y | Integer |

*Enemy*

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :x | Integer | Position of the enemy ships |
| :y | Integer |

*Heart*

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :x | Integer | Position of the hearts |
| :y | Integer |

In ”2048.rb”

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| @start\_image, @instruction\_image | Gosu::Image | Background Image for Opening, Instruction, Playing game and Ending Screen |
| @image\_2, @image\_4, @image\_8, … @image\_2048 | Gosu::Image | To draw different tiles in the game |
| @start\_song, @game\_song, @end\_song | Gosu::Song | Background songs |
| @shoot\_sound | Gosu::Sound | Sound effect while shooting |
| @font | Gosu::Font | Texts |
| @max\_score | Integer | Display the current max score |
| @playing | Integer | Define the which screen should display:  0 means opening screen  -1 means instruction  1 means main playing screen  2 means ending screen |
| @time | Integer | Display the playing time |

In ”overcoming\_obstacles.rb”

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| @sky\_background | Gosu::Image | Background Image |
| @x, @y | Gosu::Image | Position of objects |
| @song | Gosu::Song | Background songs |
| @jump\_sound | Gosu::Sound | Sound effect while jumping |
| @font | Gosu::Font | Texts |
| @score | Integer | Display the current score |
| @playing | Integer | Define the which screen should display:  0 means opening screen  -1 means instruction  1 means main playing screen  2 means ending screen |
| @time | Integer | Display the playing time |

*Records*

*Brick*

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :x | Integer | Position of the bricks |
| :y | Integer |

Table : <<enumeration name>> details

|  |  |
| --- | --- |
| Value | Notes |
| (none yet) |  |

# Overview of Program Structure

First, I will create a Game Menu ruby file, then make 4 games in 4 separated files as listed, and finally link to the Main Menu.

In Menu, players use mouse to choose their favorite games. Each time playing, the program will record the game’s name, the score, victory/defeat, and there will be the button to view the last 10 plays. At any time of playing, players can press K key to return to the Game Menu, or Esc key to return to Main Menu.

In each game, there will be 4 main screen, including menu of that game, instruction, playing screen end ending screen (game over). This can be done by using a “playing” variable: 0 means that game’s menu (not the Main menu), -1 means instruction, 1 means playing, 2 means ending. There will be many different methods to change between screens such as keyboard, mouse, variables changing, … . For example, when the game is over, player can press Space bar or left-click to return to playing screen and play the second time. Also, each screen will have an unique background and a theme song, to enhance the experience of players.

Besides some essential procedures/functions of Gosu such as initialize, button\_down, draw and update, there will be some specific ones for each game:

* Star wars: turn down, turn left, move up, move down, along with many different classes to manage the main ship, enemy ships, enemy mother ship, lives (hearts), explosions.
* 2048 and Overcoming Obstacles will have easy\_click, hard\_click, superhard\_click function to choose the prefered levels
* 2048 will have 4 functions to tiles the block, some functions to score, to decide game over or not …
* Overcoming Obstacles will also have easy\_click, hard\_click, superhard\_click function to choose the prefered levels

Used media: many images (jpg, png), songs and music (wav, ogg)

* Images: most from openclipart.org and Google Images
* Sound: some from movies like Star wars, some from websites like freesound.org, incompetech.com or tannerhelland.com