BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:		**Surname**	**Given Names**	**Student ID Number**
	1.	**Alifio**	**Rasyid**	**2201798295**

Course Code	**: COMP6335**	**Course Name**	**: Introduction to Programming**
Class	**: L1BC**	**Name of Lecturer(s)**	: 1. Monica Hidayat
Major	**: CS**		
Title of Assignment (if any)	: Tic Tac Toe Game		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 20-11-2018**	**Submission Date**	**: 20-11-2018**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Alifio Rasendriya Rasyid

Tic Tac Toe

Name : Alifio Rasendriya Rasyid

ID : 2201798295

Introduction:

Tick Tack Toe is played in a 3 by 3 board where the goal of the game is to get either 3 of a player's own icon (X or O) in a row, either diagonally or vertically. In a classic Tick Tack Toe game, if neither player makes a mistake, the game is drawn and basically both players will have to repeat the game until it a player eventually loses. In this particular version of Tick Tack Toe though there are no draws as the oldest icon in the board would be deleted as soon as all the spaces in the board (for a particular player) are filled, this results in a much more complex gameplay as it now requires the players to simulate and predict the next few moves as well as remembering which of the icons are going to be deleted.

Discussion:

Modules:

- PyGame is the main module used in this project, all of the screen, images and audio are loaded into Python using this module. Furthermore PyGame was used to detect all the events that are happening; for instance, mouse clicks and collision between "rect" (hit boxes).

Why a Tic Tac Toe game?

I decided that I wanted to make a simple casual game after browsing through my phone's Google Play (app store), many of the top games in the market are super casual games that has interactive yet very simple gameplay; some of these games includes, Flappy Bird, Ludo King, Plato, etc.

The thing is, there are already tons of Tic Tac Toe games available on the market, so how did I differentiate my game from the others?

Gameplay

Each icon (X or O) represents the players, X (player 1) will start first followed by O (player 2) and so on. Every time each players insert their 5th icon (X or O) the oldest icon will automatically disappear.

If they wanted to, the players are also able to go back to the main menu anytime by simply clicking on the Tic!Tac!Toe! Logo on the upper left corner.

To quit, players have to simply click on the quit button on the main menu screen or pressing the close button.

How it Works?

Firstly pygame.mixer and pygame are initialize and all the object (class) are declared. Next before the main game loop are started func.update_screen() reads status.menu and automatically start the game in an active mode. While the loop are ongoing, func.check_events() repeatedly detect any sort of event happening; for instance clicking the quit button would trigger a sys.exit() command which would instantly close the game.

Other functions in the game includes check_mouse_click(), check_box_clicked() and check_win(); check_mouse_click() essentially controls what to do when a certain thing is clicked (other than the quit button), while check_box_click() detects which of the 9 boxes are clicked and updates status.pointer, lastly check_win() basically goes through the boxes and check if any players wins the game.

In short the game started with an update_screen() function and then proceed to the loop. Inside the loop, check_events() function is constantly checking for any sort of event, it includes checking mouse clicks as well as an update_screen() function at the end of the loop, it also includes a check_win() function although only when the game is on the in-game screen (while the players are playing the game).

Class

In total there were 6 Classes used in the project, this includes:

- GameStatus()

This class store the status of the menu, pointer(points which box), turn and win which tells which player has won the game. It also includes a function which resets the status (if the player chose to restart the game).

- Settings()

This class store the setting of the screen as well as a function which loads up the music/sound.

- Start_Menu()

This class loads and set up the game's logo, start icon and quit icon on the start menu as well as drawing it to the screen.

- Game_Screen():

This class loads and set up the game's logo and squares to the in-game screen as well as drawing it to the screen.

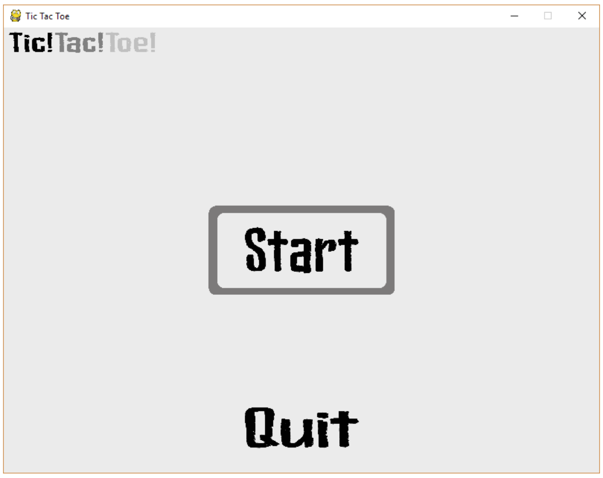
- Icon():

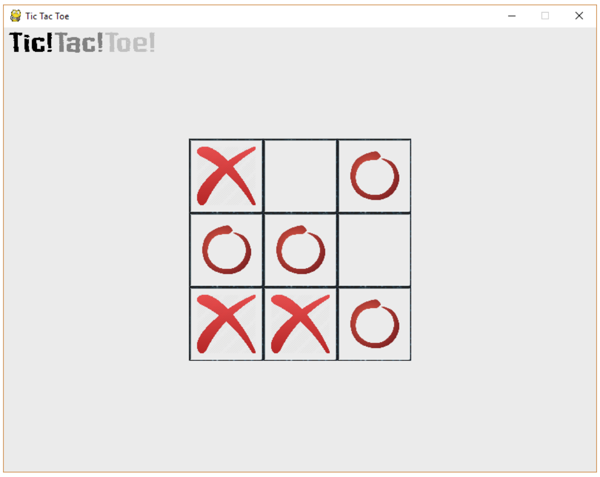
This class loads and set up all the character's icons (X and O), the data of the icons present are stored in a list called icons_rect_present_data. The class also has 2 functions, one updates the icons_rect_present_data list the other draws it to the screen.

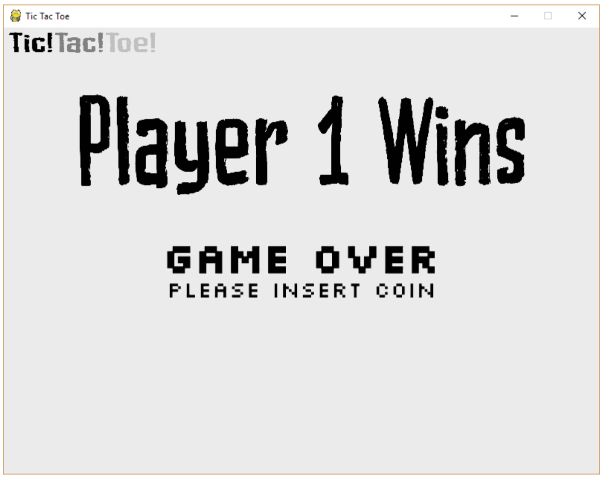
- Game_Over():

This class loads and set up the game over icon as well as player win image. The draw function automatically detects which player wins based on the number of turns of the game, thus it blits player 1 winning image only when player 1 wins and the other if player 2 wins.

Evidences

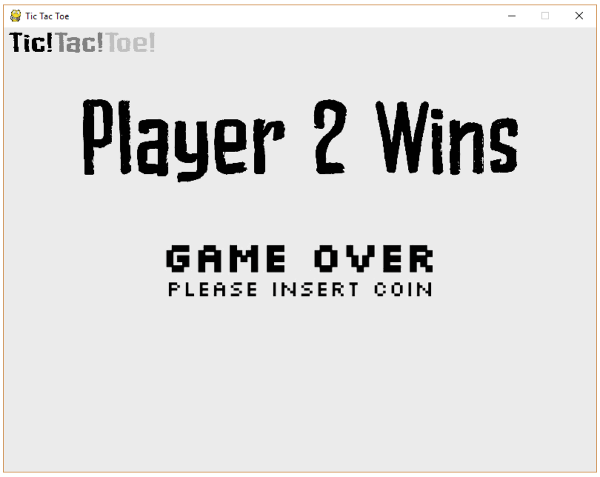






Player 1 Wins

GAME OVER
PLEASE INSERT COIN



Player 2 Wins

GAME OVER
PLEASE INSERT COIN

References

- Eric Matthes (2016), *Python Crash Course (Project 1: Alien Invasion)*
- Tech With Tim, *Pygame Programming Tutorials,* retrieved from https://www.youtube.com/playlist?list=PLzMcBGfZo4-lp3jAExUCewBfMx3UZFkh5