B.Sc. Computing Stage 1 2016/17  
Activity Led Learning Project  
Semester 1 – Networked Tic-Tac-Toe

# Group Details

**Group Name:**

**Group Members:**

<First Name> <Surname> <Student ID>

# Submission Notes:

Do not edit this Section

* Your submission must be uploaded to Moodle **by 23:55 on Monday 12th December.** The submission point will appear in the ECU177 – Computing Moodle Page.
* Place this report in a directory **alongside all the code needed to run your game**. Compress it to a zip folder and upload the zip to Moodle.
* Your game will be tested using Python 3, a Raspberry Pi and the EC Building computers (so ensure it works with all these).
* **You submit only one report and one game per group**. All team members should agree on the final submission.
* Each report section has a maximum word count. **Any words over the maximum will not be marked**. You need not write up to the limit.

# Game Documentation (max 500 words)

*Give sufficient and clear instructions for staff members to run your game. Screenshots / diagrams are allowed if helpful (not included in word count).*

To enable multiplayer functionalities, first start by executing the Server.py file. To run the actual game, execute the Main.py file (twice if planning to play multiplayer). To change game modes (Multiplayer (local/multiplayer) or vs. the AI) go to the settings menu from the main menu. Finally, to start the game click “Play the Game”.

https://github.com/JamesL33/Tic-Tac-Toe-Project/blob/master/README.md

# Game Design (500 words / 1 page)

*You can also discuss what tools (e.g. Tkinter etc) if any you used; what programming techniques (e.g. OOP, functional etc); how you represented the game components (e.g. lists, dictionaries); and how your different functions communicate with each other. You can also discuss how these things changed over the course of the project.*

When I first started creating the game I was using the TKinter module. This allowed me to create the Tic-Tac-Toe game with a GUI. This was not asked of us however I felt that it was a better way to play the game. Down the line, I realised that this would not be the best idea since TKinter is in my opinion a fiddly way of creating a user interface therefore I decided to remake my game while using the Pygame module. This made more sense because this module was created to aid creating games in Python.

I represented the game using a list. I decided to use a list and not a two-dimensional array because there was no need to overcomplicate it. This list simply contains noughts or crosses which are in turn draw on the display.

In the networking portion of the game I represented the messages that were sent using lists which contain an element telling the game which function to perform and then the arguments that the function needed. The pickle module was also a huge part of the networking in the game since when I send data over the network I want to preserve the data types hence the use of the pickle module.

I created the game by separating it into several smaller sections these made up the six classes which were used. These classes are:

Game\_AI – This class handled the AI portion of the game in our case this class had a function which return the move of the AI.

Game\_Functions – This class has all the function which will be used in the playing of Tic-Tac-Toe such as the Take\_Turn function.

Server – This class was used to create a socket object which the client would connect to. This class also handles all the server based functionality.

Client – This class handles the sending of information to the server and the storage of the data which is received.

Main – This class handles the main portion of the main such as running the game and keeping the game window open while it is necessary.

Menus – This class handles all of the drawing of the new screens and menus for the game.

Changes over the creation of the game – At first the game was created using TKinter then it was changed and remade using the Pygame module. This was an important switch as it eventually made making the game easier and less hassle. This is an important part of programming since there is no reason for making it difficult for yourself if there is no need.

During the creation of the game it was rewritten several times to accommodate new features such as the networking an Ai; this was not a very effective way of creating the game however I will take this into account in the next project.

# Project Management (max 500 words / 1 page)

*Describe how you managed the project: how the team divided work, how you communicated, how you managed and merged code, how you came to joint decisions and how you resolved any disputes.*

Managing the project as a group was a challenge at the start as some group members like James and Goncavlo …. On the other hand Shivani and Nimra required extra time…. However, we did manage to keep up with the tasks and complete them on time….

We had 4 group members altogether hence why we decided to split into pairs and come up with a simple game code to begin with. Subsequently, both codes did function properly but we chose to work ahead with James and Goncavlos code and built up from there as a group. As a group this was our first task and we succeeded…

Good group communication is one of the key elements of successful group work, and throughout this project we made sure all team members were contributing towards the project equally and interacted with each other with respect and integrity. Having trust in one another increased the confidence which made it easier for us to interact with each other and make decisions effectively. In addition, to increase our level of communication, we made sure we all stayed in contact outside our Lab hrs, so we made a group chat on whatsapp, repository on Github and exchanged email addresses for further enquiries. Furthermore, we also arranged two group meetings throughout the project to talk about the progression and further plans to implement the project.

We were introduced to Github….which helped us get started on our code .managing and merging the code…..We agreed on decisions and then …

Disputes….

Successful group work means everyone in the group contributes to the overall group dynamic…

In conclusion, as a group we agreed on individual tasks based on our past experiences and what we were confident to achieve best results in. However, some tasks were quite challenging for Nimra and Shivani, ie coding and the networking stage of the game, however James and Goncavlo who were more experienced members of the group supported them which is one of the advantages of group work.

# Work Allocation (max 500 words / 1 page)

*Describe which team members did what. If you prefer (or you cannot agree), each team member can write a short statement here.*

James Lee – GitHub management and main contributor to the creation of the game along Goncalo. Goncalo created the game menus which are used thoughout the game. I created the Networking, the game AI and also the game logic. I also create a text based version of the game and a Tkinter version of the game. This can all be seen in the history on the GitHub. I stopped developing on the other two versions of the game when I decied to work on the Pygame version so they were removed from the GitHub.

As a group we decided to divide the tasks in to pairs, as this would broaden our horizons, therefore each group member would contribute towards the assignment. Shivani and Nimra had started from the basics compared to James and Gonsalo who were more advanced regarding the tasks that required programming. Although Shivani and Nimra had created a basic tic-tac-toe game, they had managed to create it successfully. However Shivani and Nimra had found the networking phase of the project, very complicated, they had to do additional research regarding the networking. This process required extra time for them to understand the process, however they were unsuccessful in building the network for

# Group Reflection (max 500 words / 1 page)

*As a group reflect on your project: what went well, what did not, what would you have done differently? What else would you have done given more time? What areas needed more time and though and most importantly, what lessons can you learn from this to do better on the next project?*

Working as a group – The group did not work that well together; James tried to set up different methods of communication however there was never a definitive ways of communicating. If we were going to do the project again It would have worked better if there was more communication between the group.

Networking – The networking of the game was a lot more difficult than we first thought therfore it took a lot more time to work on than James first thought however we have now go a working networked game.