BSC – HGP - Project

Go

UI Design Document & Report

# Division of Work

Student Name1: Paula Carine Oehme Student Number1: 2957903

Student Name2: Sarah Narayamy Tavares Silva Student Number2: 2960992

Please complete the sections below with regard to the estimate of the division of work between the two partners

If the work was split in the range of 45% to 55% per partner, then that is fine and simply say “Work was evenly divided”. If this was not the case, then state with a summary sentence. This is the important statement of this file.

Division of work: work was evenly divided \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Code repository log (if applicable)

Paste here

## Percentage of work completed by each partner on each class / task

Some areas require more work than others so this is only for reference. An average of these values will not be calculated.

|  |  |  |
| --- | --- | --- |
| **Filename / Task** | **Paula Oehme** | **Sarah Silva** |
| GoBoard | 70% | 30% |
| Menus/Buttons/labels | 30% | 70% |
| Code Documentation | 50% | 50% |
| UI Design | 50% | 50% |
| Learning rules of draughts | 50% | 50% |

# UI Design

**Submission:** Edit this template and submit it as part of your submission.

**Length**: Should be 4 pages approx. Word count is flexible, but all decisions should be clarified.

To achieve good marks in this item ensure that this document is well structured and addresses each of the following headings and subheading. The explanation of each UI Design Choice should be clear, precise and show substantial consideration has been made, references are welcome. All decision should be explained regardless of how basic they are. Do not cut and paste justification from the internet (plagiarism) or notes but include references and explanations in your own words where appropriate.

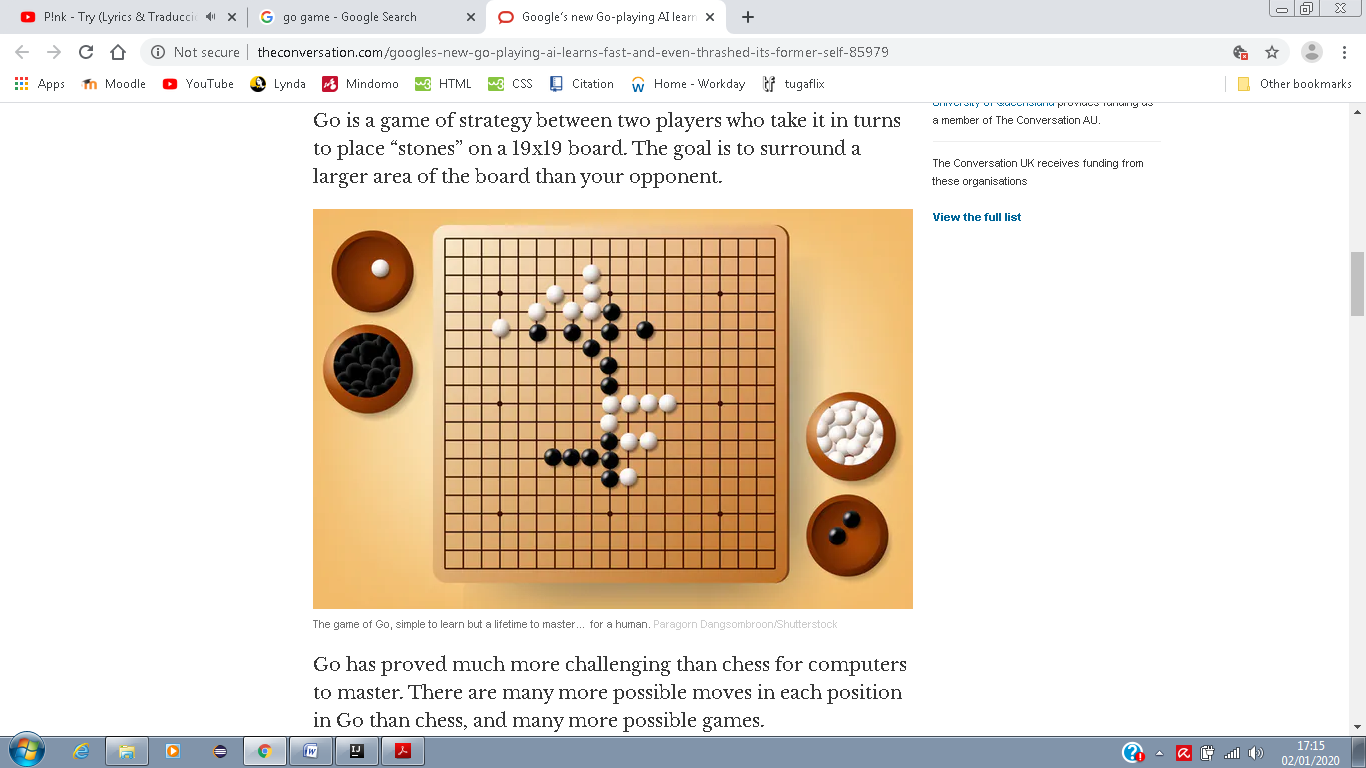
**Student Name:** Paula Oehme / Sarah Silva

* Include multiple screen shots of the application each focusing on a different component clearly labelled
* Clearly indicate what is working and not work.
* Discuss each component under the following headings
* Location: e.g. The button was placed in the bottom right to as it is the default location to confirm and action
* Colour: The colour scheme was chosen to avoid the main form of colour blindness and produce high contrast for the visually impaired.
* Size:
* Style:
* Etc.

**N.B.** Clearly mention any additional features here either visual or functional so that appropriate marks are awarded

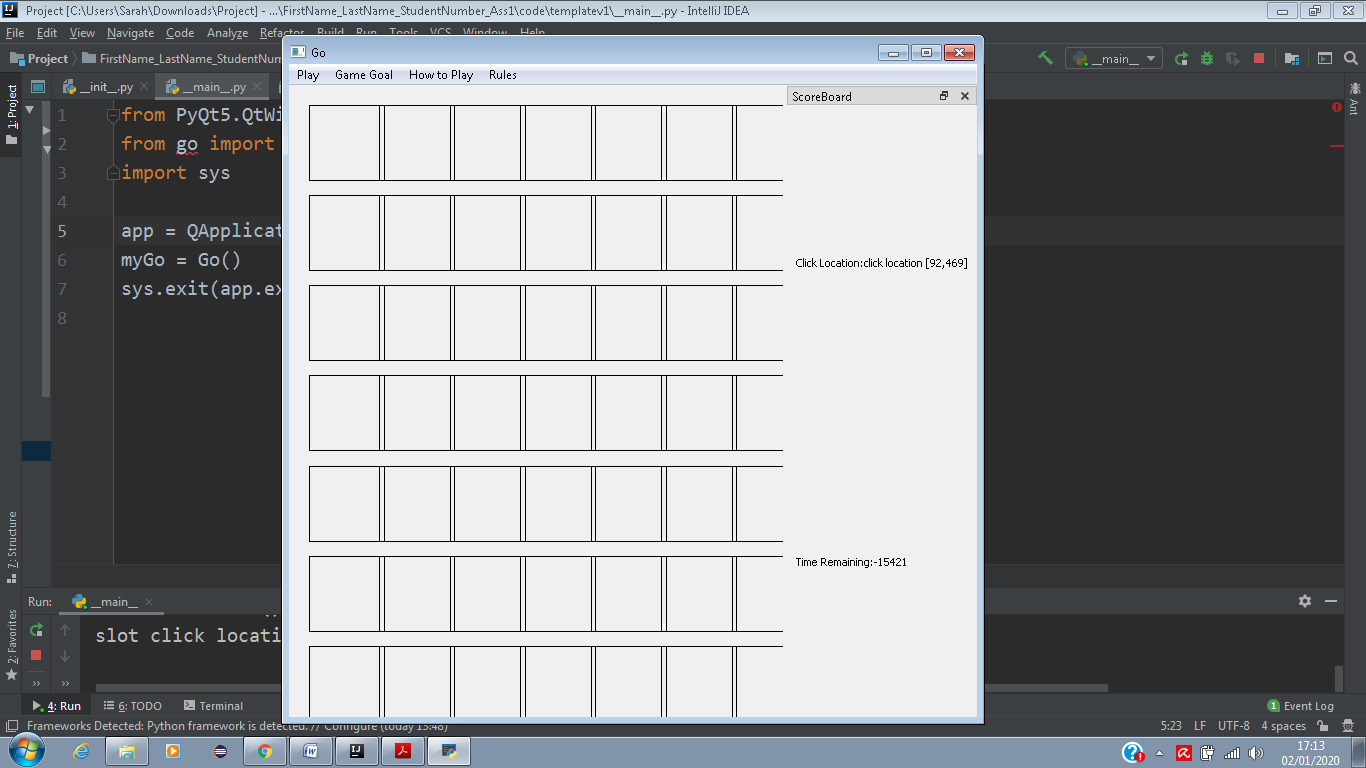
* **Introduction**

Go game is a strategy board game, invented in China over 2500 years ago, in which the goal is to gain as much more territory than the opponent by capturing the opponent’s stones. It does not have many rules, which seems to be easy to either play or understand, but it is actually a quite complex game, even more than chess. The picture bellow shows an image of a board for the game, as well as the stones (black and white) used to play it.



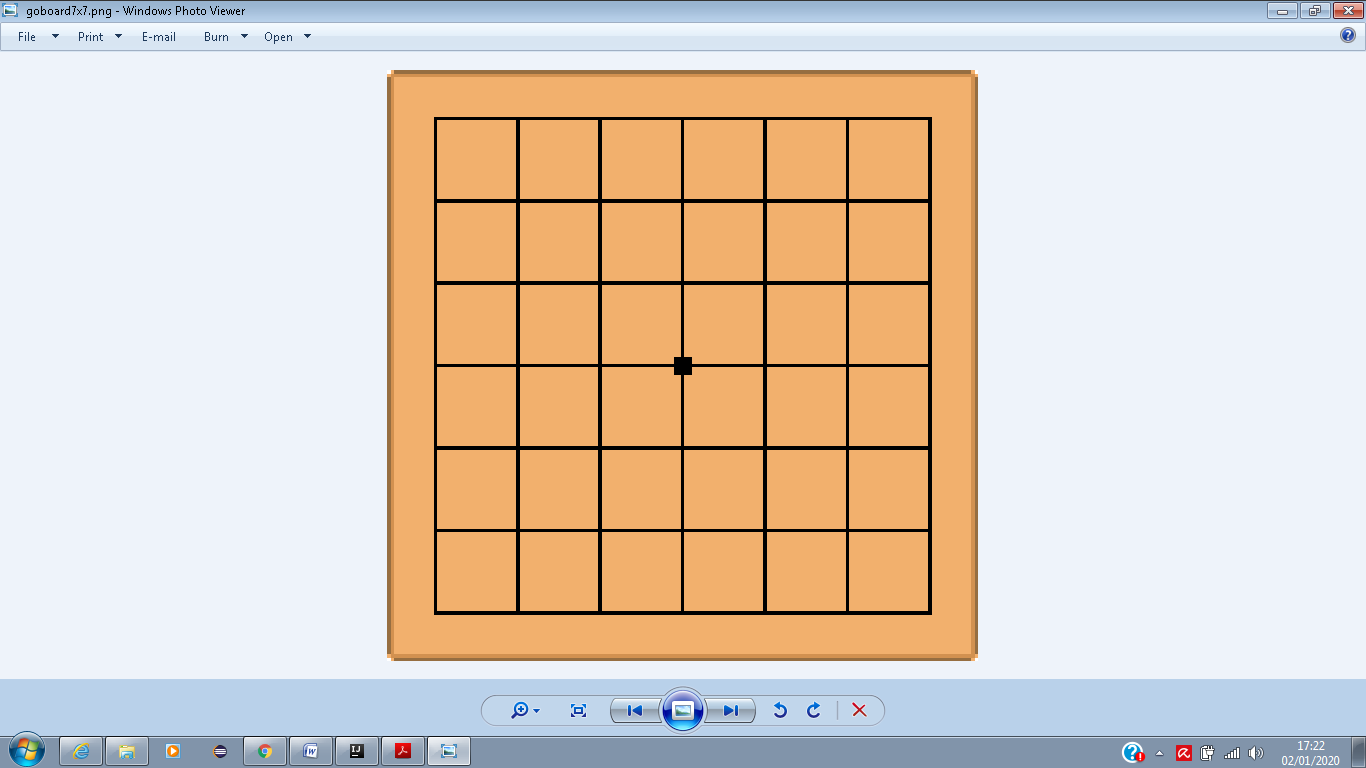
* **Menu**

The menu was chosen to be placed on top of the board in the application since it seems to be a pattern; also, it makes it easier for the user to locate possible icons for help, etc. The aim of the menu was to provide information about the game such as rules and how to play it, as well as a link to actually Play the game by resetting an existing one or simple starting a new one and to exit the game.



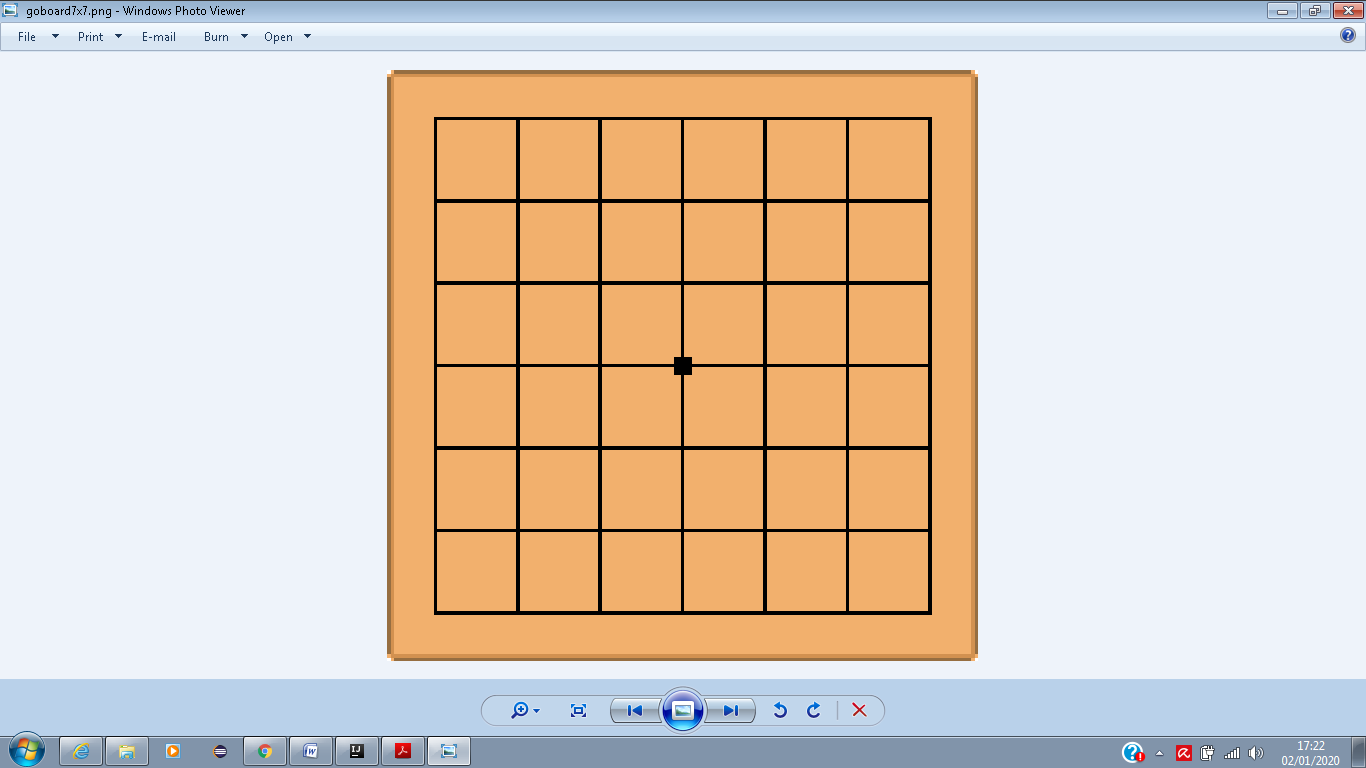
* **Board**

Unfortunately we were not able to make the board work in the application. The plan was to use the image bellow as a 7x7 board and the stones would be placed on it.



* **ScoreBoard and Time counter**

The plan for the ScoreBoard and the Time counter was to place them right bellow the board itself in order to each player to have easy access to each score points and the opponent’s. The picture bellow shows how it should have been. Unfortunately we were also not capable of doing so.



**ScoreBoard:**

Player 1: Player 2: Clock:

# Screen Shots of Working/Not Working Features

**N.B. Be sure to comment what is working and not working for each of the tasks. The boxes should be expanded to contain the content.**

All code should be testable where possible and error message should be displayed to show where code has failed.

|  |
| --- |
| **Task 1 (1 image with description + what is working/not working)** |
| Unfortunately we could not find a way to generate the board for the application. The following image is what we generated. The plan at start was to place the ScoreBoard bellow the Board itself with the points for both players and whose turn was it. The clock counting the minutes would be placed right beside the ScoreBoard with the counting down clock. |

|  |
| --- |
| **Task 2 (6 images of working Menus/buttons/Labels including description + what is working/not working)** |
| We used icons on the menus to make the options visually easier to recognize by the user. The idea for How to Play menu options and Rules options was to print out images showing how the rules work in order to make it more understandable by the user. Unfortunately, the QMessageBox does not allow it to print images as parameter, thus we decided to leave without it. |

|  |
| --- |
| **Task 3 (2 images + what is working/not working)** |
| Since we could not do most of the game logic, we set up a skeleton of what the code would be and how the logic would work. Bellow, there a screenshot of the logic for placing a stone using mouse click on the board. For that, it must verify if placing the stone in the free intersection will not violate any of the other rules. |

|  |
| --- |
| **Task 4 (2 images + what is working/not working)** |
| The image bellow shows the logic for a method to check for the suicide rule. This means that if placing a stone in an intersection will immediately allow that stone to be captured, the program will not allow the placement of such stone. |

|  |
| --- |
| **Task 5 (2 images + what is working/not working)** |
| In this skeleton of the ko method, the logic is to keep track of the previous move, and if the following move would mean to endlessly repeat capturing and recapturing the stones, the move will be denied and the user will be allowed to place the stone in a different location. |

|  |
| --- |
| **Task 6 (2 images + what is working/not working)** |
| The capture method states the logic as follow: the program will check if a stone or a group of stones can be completely surrounded by the opponent piece. In that case, the program will allow the placement of the stone and the piece/pieces that were surrounded will be captured, adding points to the user who captured the stones. |

|  |
| --- |
| **Task 7 (2 images + what is working/not working)** |
| This task uses the same logic as the previous task which states the logic as follow: the program will check if a stone or a group of stones can be completely surrounded by the opponent piece. In that case, the program will allow the placement of the stone and the piece/pieces that were surrounded will be captured, adding points to the user who captured the stones. |

|  |
| --- |
| **Task 8 (2 images + what is working/not working)** |
| In order to check for the winner and to end the game, the following image shows the logic of the method.    If there was two consecutive passes, the game must be ended and the result would be displayed, showing the winner and how many points each player have made during the game. |

|  |
| --- |
| **Task 9 (2 images + what is working/not working)** |
|  |