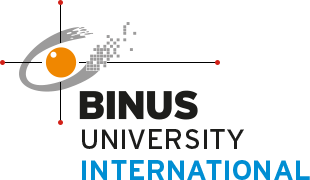
**Project Report**

Odd Semester (2021)



# BINUS UNIVERSITY

# BINUS INTERNATIONAL

|  |
| --- |
| **Assignment Cover Letter****(Individual Work)** |

**Student Information:**

**Surname:** Koesmanto   **Given Name:** Edward Alvin   **Student ID Number:**2501963141

**Course Code :** COMP6047001 **Course Name :** Algorithm and Programming

**Class :** L1AC **Lecturer :** Jude Joseph Lamug Martinez, MCS

**Type of Assignments:** Term Final Project

**Submission Pattern**

**Due Date :** 17 January 2022 **Submission Date :** 17 January 2022

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: Edward Alvin Koesmanto

# **Project Specification**

## **Overview**

This program is a game of Sudoku, the goal of the game is to finish a randomly generated Sudoku board with a decent difficulty. The program uses recursive backtracking to keep a solution in mind of the computer.

1. Program Input:
2. Mouse Click that signifies the position selected in the grid
3. Number inputted by the user from both Numpad and Number Column of the keyboard
4. Program Output:
5. A sketch of the number inputted (grey colour) by the user
6. A number inputted by the user if it is entered and correct
7. A strike if number of inputted by the user is wrong

## **Third-Party libraries/modules used**

1. Pygame

* Making games with Python Language

1. Numpy

* Data Processing

1. Dokusan

* Generating the Sudoku Board

## **Solution Design**

Files Involved are solver.py and SudokuGrid.py

**Solver.py**

* Purpose: Stores the functions that makes the program able to do recursive backtracking that will be involved in the main file.

A screenshot of a computer

Description automatically generated with medium confidence

* The findempty() function takes in 1 parameter, the board.
* It will see if there is a cube empty in the board and returns the index of that cube.

­ Text

Description automatically generated

* The valid() function takes in 3 parameters the board, number entered and position of the board being checked.
* It will check if each element in the row and column is equal to the number just entered or not.
* If the position being checked is the position that the number has just been entered, it will ignore it and move on to another position.
* box\_x and box\_y will be assigned to their boxes of row and column.
* It will assign the index of cube being checked by multiplying by 3 because in the assignment, the box is being divided by 3 to get the value of the box.
* If value of the position being checked is equal to the number just entered and the index is equal to the position of the board where the number has just been entered, it will ignore it and move on to another position.
* If valid() is false then the board is not suitable.

Text

Description automatically generated

* The solve function takes 1 parameter which is the board.
* It will assign the findempty() function to the find variable.
* If findempty() does not find an empty cube, it will return True.
* If findempty() finds an empty cube, it will assign row and col to find
* It checks if index selected is valid or not using the valid() function.
* If it is valid, it assigns the index of row and col of the board to i.
* It will then check the possible number that may be entered to complete the board from 1-9 again and again until True or 1-9 has all been tried and nothing is correct. It will then return False.

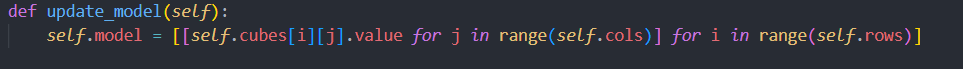
**SudokuGrid.py**

* Purpose: The making of the GUI is in this file.

Text

Description automatically generated

* The Grid class will generate a random board using Dokusan and with the help of Numpy, turn it into an array. It will then reshape the array to a 9x9 board and change the type of the elements inside the board as integers.
* The constructor method defines the rows, columns, cubes of board, width, height, model, and the position selected.



* The update\_model() function will update the model according to the 9x9 columns and rows.

Text

Description automatically generated

* The sketch() function will allow the user to sketch a number inputted by the user but not entered yet. It is a preview of the number.

Text

Description automatically generated

* The place() function will validate if the number sketched by the user is suitable or not to be entered.

Text

Description automatically generated

* The draw() function will draw the board.
* It will first draw the grid lines with a thicker line to separate the boxes when it is 3x3.

Text

Description automatically generated

* The select() function will select the box clicked by the user

Text

Description automatically generated

* The clear() function will clear the box selected

Text

Description automatically generated

* Returns the box clicked by the user

Text

Description automatically generated

* The is\_finished() function checks if the board has any missing number or not.

Text

Description automatically generated

* The Cube class defines the values needed to make the GUI such as value which cannot be changed, the temp which is the number sketched by the user, row, column, width and height of the window.

Text

Description automatically generated

* The draw() function will draw the window and boxes of the grid. It will also show the selected box with a blue outline.

Text

Description automatically generated

* The set() function will set the value of the number inputted.
* The set\_temp() function will set the value sketched by the user.

Text

Description automatically generated

* The redraw\_window() function will redraw the window everytime there is a change to the window, such as number sketched, inputted, time allocated and strikes.

Text

Description automatically generated

* The format\_time() function will format the time to hours:minutes:seconds and show the time running the program.

Text

Description automatically generated

* The main() function will set everything for the run of the program.
* It will draw the window, set caption, set board and set values to none.

A picture containing logo

Description automatically generated

* This shows while True, the program will run
* The play\_time variable shows the time spent running the program.

Text, calendar

Description automatically generated

* The allocation of keys.

Text

Description automatically generated

* It shows what the return key does.
* If the selected box has a sketch, it will validate if the number entered is correct to be placed there or not.
* If it is correct, it will turn the sketch to a real value into the board and print “Success”
* If it is not correct, it will clear that cube, erase the sketched value, and add a strike and print “Wrong”.

Text

Description automatically generated

* If the board is done, it will print “Game over” and a game over window will show.
* After 2 seconds, the program will exit.

Text

Description automatically generated

* Same as the above but using Numpad.

Text

Description automatically generated

* Shows the clicked cube using mouse.



* Sketch the value clicked by the user but not entered yet.



* Redraw the window and update the window.

Text

Description automatically generated

* Call the main() function and the Sudoku window will be shown.
* After the board is all done, the game window will close and exit the program.