./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc56015125)

[Miniproject -1 [Team] 4](#_Toc56015126)

[Module/s 4](#_Toc56015127)

[Topic and Subtopics 4](#_Toc56015128)

[Objectives & Requirements 4](#_Toc56015129)

[Objective 4](#_Toc56015130)

[Requirements 4](#_Toc56015131)

[Design 5](#_Toc56015132)

[Test Plan 5](#_Toc56015133)

[Implementation Summary 6](#_Toc56015134)

[Git Link 6](#_Toc56015135)

[Git Dashboard 6](#_Toc56015136)

[Summary 7](#_Toc56015137)

[Individual Contribution & Highlights 11](#_Toc56015138)

[Summary 11](#_Toc56015139)

[Challenges faced and how were they overcome 11](#_Toc56015140)

[Future Scope (If applicable) 11](#_Toc56015141)

# Miniproject -1 [Team]

## Module/s

Python programming

### Topic and Subtopics

Classes

Object Oriented Programming

GUI with pygame

## Objectives & Requirements

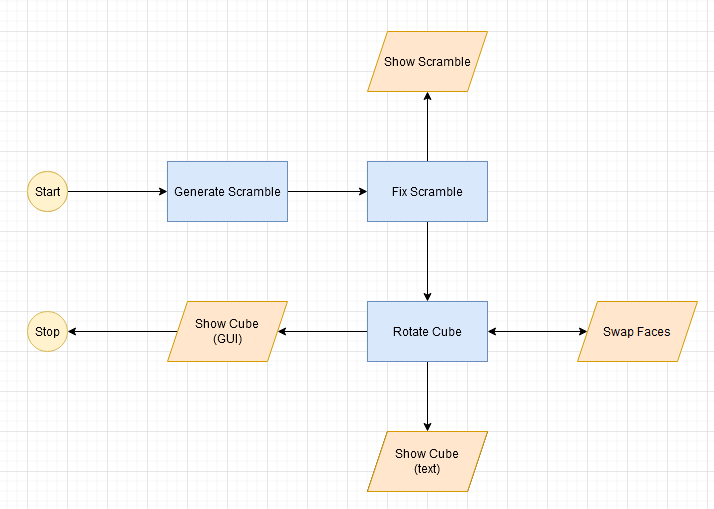
### Objective

To create a program that will generate a scramble for a 3x3 Rubik’s cube and show the scrambled cube as a GUI element.

### Requirements

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| HL\_01 | Generate a scramble of given length |
| HL\_02 | Show the scramble in cube notations |
| HL\_03 | Rotate the virtual cube according to the scramble |
| HL\_04 | Display the cube in text format |
| HL\_05 | Display the cube as a GUI element |
| HL\_01\_LL\_01 | Use random generator to append turns to scramble list |
| HL\_01\_LL\_02 | Iterate thorough the generated scramble list and fix consecutive turn occurrence |
| HL\_03\_LL\_01 | Swap two elements in the cube according to the face links |
| HL\_05\_LL\_01 | Draw a grid on the window |
| HL\_05\_LL\_02 | Draw each face of the cube |
| HL\_05\_LL\_03 | Arrange each face to its proper position |
| HL\_05\_LL\_04 | Scale the window to fit the screen |

## Design



## Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Expected input** | **Expected output** | **Actual output (Pass/Fail)** |
| TC\_1\_HL\_1 | Scramble of given length | Scramble length | Generate a scramble of given length | Pass |
| TC\_2\_HL\_2 | Scramble in cube notations | The scrambled length must be converted to cube notations | Show the scramble in cube notations | Pass |
| TC\_3\_HL\_3 | Rotating the virtual cube | Scramble in cube notations | Rotate the virtual cube according to the scramble | Pass |
| TC\_4\_HL\_4 | Cube in text format | Virtual cube | Display the cube in text format | Pass |
| TC\_5\_HL\_5 | Cube as a GUI element | Text format of cube | Display the cube as a GUI element | Pass |
| TC\_6\_HL\_01\_LL\_01 | Use random generator to append turns to scramble list | random generator | Append turns to scramble list | Pass |
| TC\_7\_HL\_01\_LL\_02 | Iterate thorough the generated scramble list and fix consecutive turn occurrence | scramble | Cube with generated scramble list and fixed consecutive turn occurrence | Pass |
| TC\_8\_HL\_03\_LL\_01 | Swap two elements in the cube according to the face links | coordinates | Swapped elements in the cube according to the face links | Pass |
| TC\_9\_HL\_05\_LL\_01 | Draw a grid on the window | cube | A grid on the window | Pass |
| TC\_10\_HL\_05\_LL\_02 | Draw each face of the cube | Face | each face of the cube | Pass |
| TC\_11\_HL\_05\_LL\_03 | Arrange each face of the cube | All the Faces | Arrange each face to its proper position | Pass |
| TC\_12\_HL\_05\_LL\_04 | Scale the window to fit the screen | Cube Window | Scale the window to fit the screen | Pass |

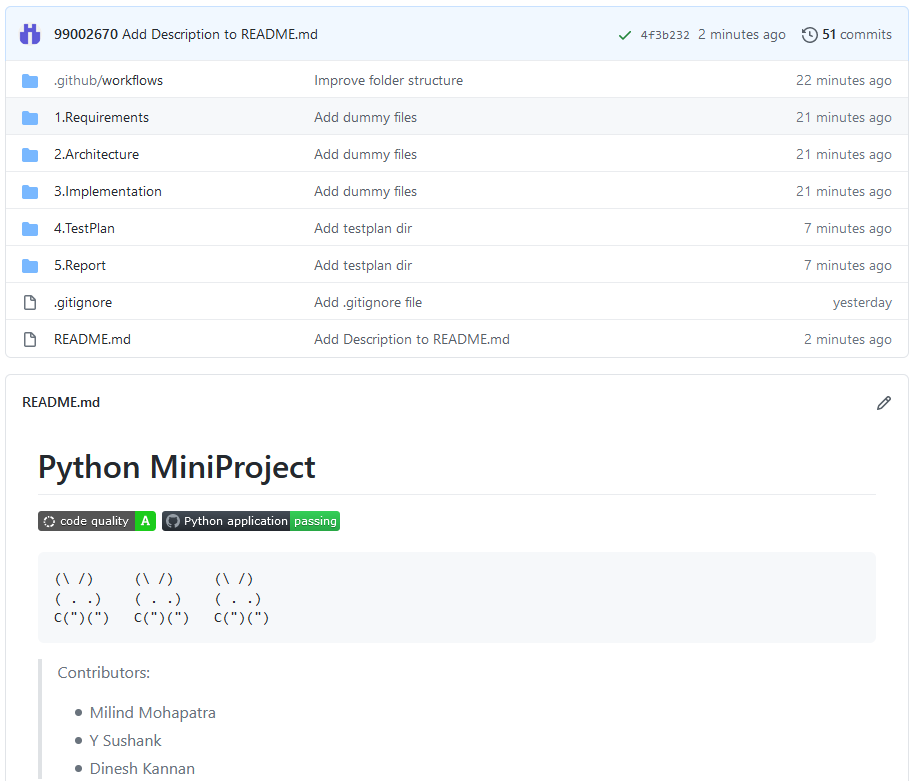
## Implementation Summary

Once the program runs, it automatically generates a 3x3 Rubik’s cube scramble and the fixes it by removing any consecutively occurring same face turns and shows the scramble in cube notations. Then the program rotates a virtual cube and displays the cube as a 2D GUI element in its scrambled state.

### Git Link

### <https://github.com/99002670/2009MYSEMB03-PYTHON-7>

### Git Dashboard

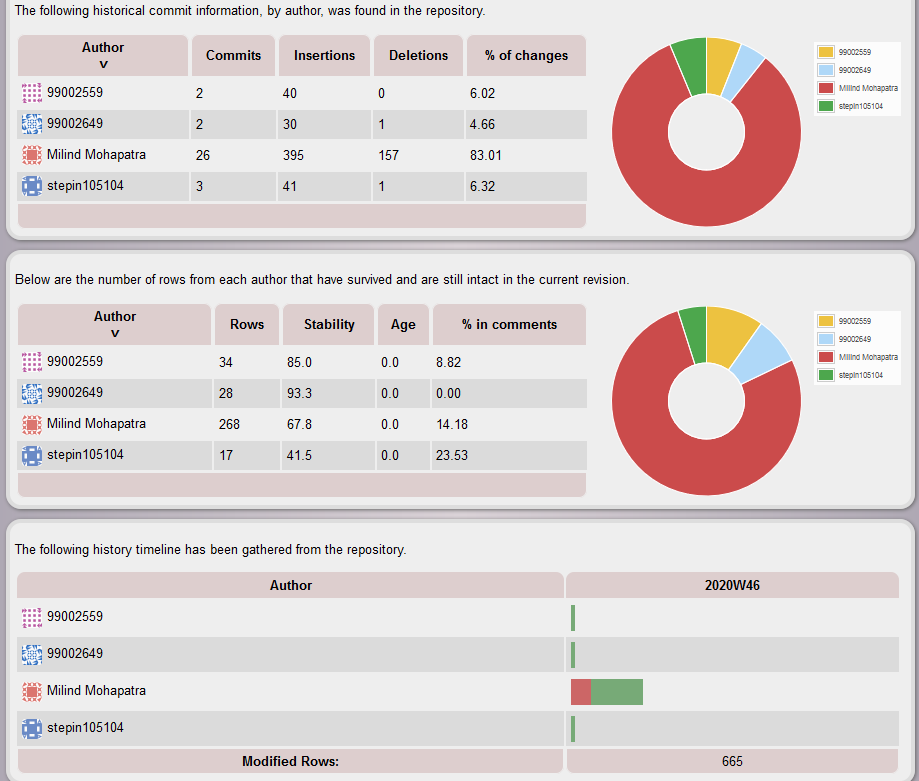


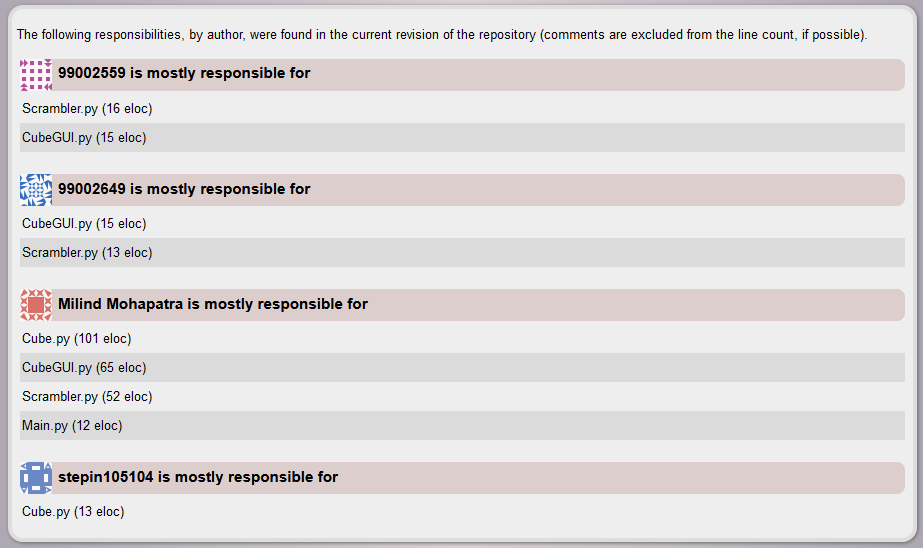
### Summary

The git repository consists of the following directories:

1. Requirements: Consists of high level and low level requirements
2. Architecture: Consists of uml diagram
3. Implementation: Consists of the project files
4. TestPlan: Consists of the high level and low level test plans
5. Report: Consists of the reports generated

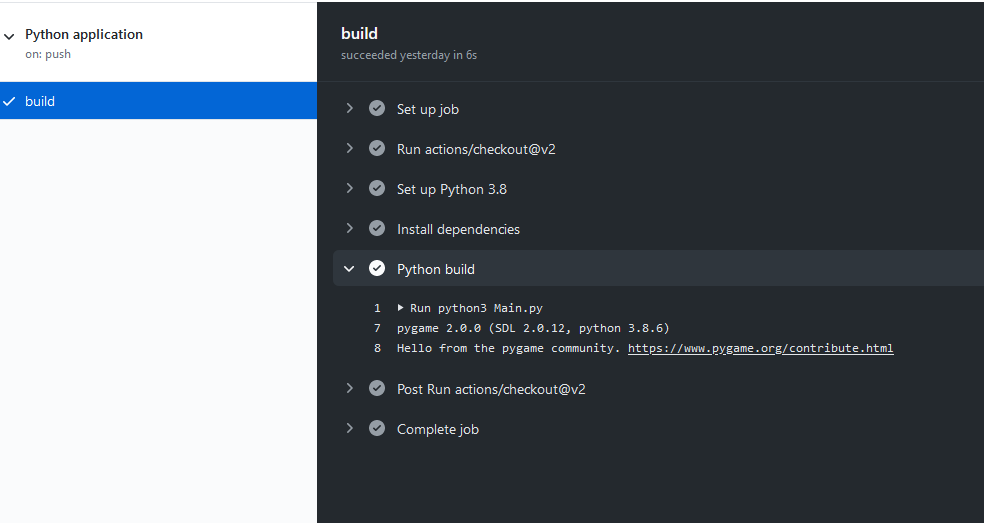
#### Git inspector summary





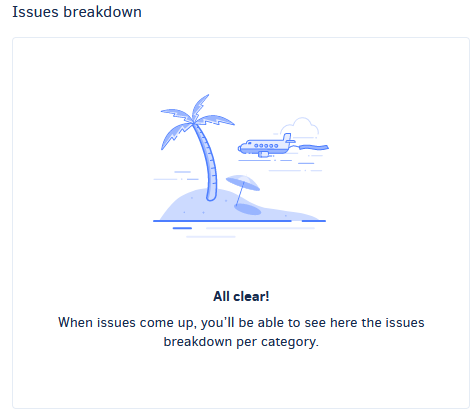
#### Build

* Install dependencies: <pip install pygame>
* Run Main.py: <python Main.py>



#### Code quality

* Code quality was improved after considering the warnings thrown by Codacy.
* No unused variables
* Use of easily understandable variable names.



## Individual Contribution & Highlights

Dinesh Kannan(99002559)

1. Add int function for cube GuI class
2. Add generate Scramble function

### Summary

Once the program runs, it automatically generates a 3x3 Rubik’s cube scramble and the fixes it by removing any consecutively occurring same face turns and shows the scramble in cube notations. Then the program rotates a virtual cube and displays the cube as a 2D GUI element in its scrambled state.

### Challenges faced and how were they overcome

* Integrating the contributions of all three members was a little difficult
* Git inspector does not show all the commits that are made

### Future Scope (If applicable)

* Add timer functionality to time solves
* Add solve storage in a local file