GENESIS - Learning Outcome & Mini-project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1	11-11-2020	Amit Das, Aditi Dutta, Smita Senapati			

Contents

TABLE OF FIGURES

Miniproject -1 [Team] : SUDOKU SOLVER (VISUALIZING BACKTRACKING)

Module/s

Python and pygame library.

Topic and Subtopics

The various areas covered in this program are as follows

- Object Oriented Programming
- Using the pygame Library
- Event based programming

Objectives & Requirements

OBJECTIVES

The object is to develop an algorithm that can solve any 9x9 sudoku while visualizing the active steps being taken by the algorithm in real time to solve the puzzle.

REQUIREMENTS

HIGH LEVEL REQUIREMENTS

ID	Requirements
HH_01	0 should be used to represent the blank spaces
HH_02	Index of empty spaces should be found out first.
HH_03	Backtracking algorithm is used for checking if any repeated number exists in the row, column or grid.
HH_04	GUI should be developed
HH_05	GUI should get updated at every step.
HH_06	Finally, the solved board should be displayed.

LOW LEVEL REQUIREMENTS

ID	Requirements
HH_01_LL_01	The unsolved board is iterated over by 2 nested loops.
HH_02_LL_01	Index of all zeros are returned
HH_03_TT_01	Validation and backtracking for rows
HH_03_TT_02	Validation and backtracking for columns
HH_03_TT_03	Validation and backtracking for boxes
HH_04_TT_01	Pygame library is used for GUI development
HH_05_TT_01	Execution of program should start only on a keypress (eventbased)

Design

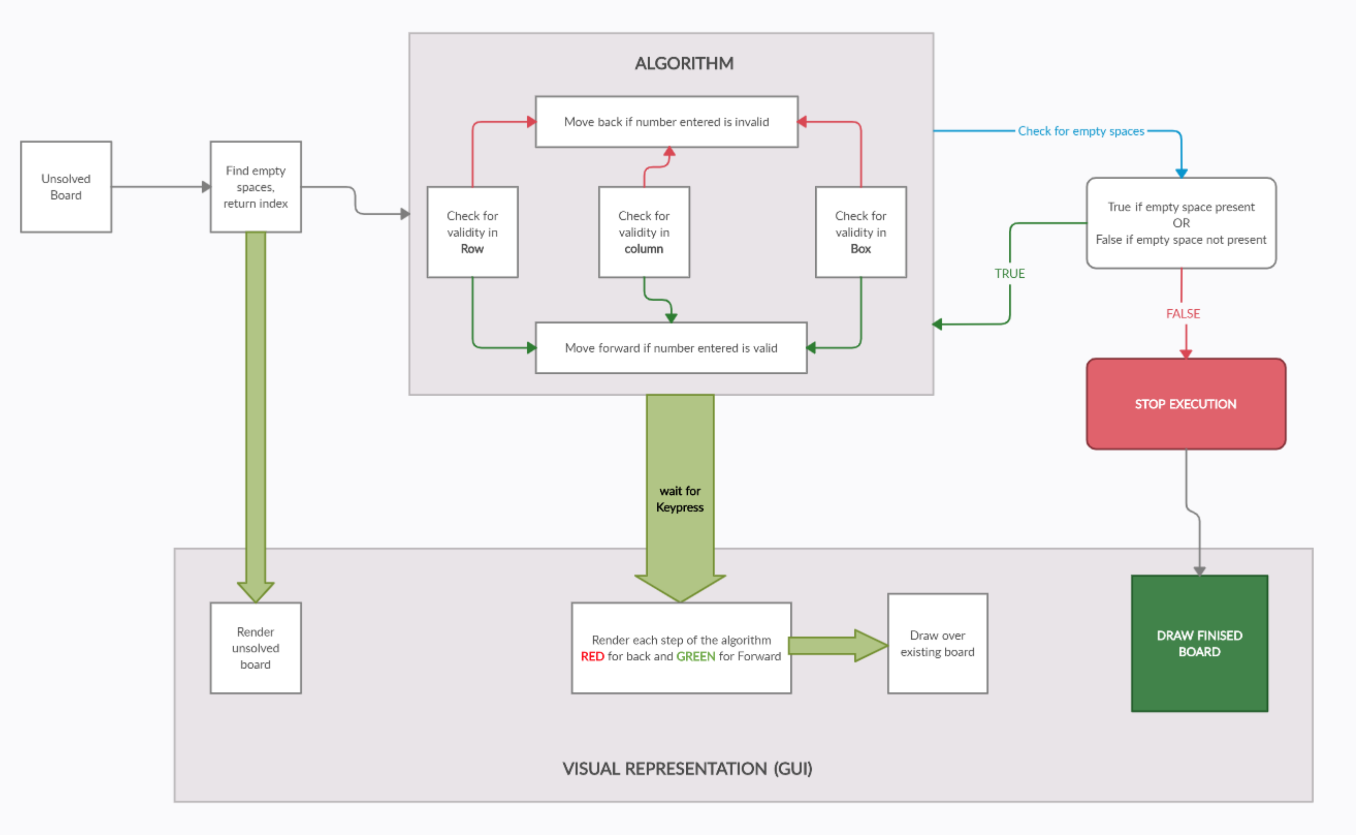

ALGORITHM
Unsolved Board
Find empty spaces, return index
Move back if number entered is invalid
Check for validity in Row
Check for validity in column
Check for validity in Box
Move forward if number entered is valid
Check for empty spaces
True if empty space present OR False if empty space not present
TRUE
FALSE
STOP EXECUTION
wait for Keypress
Render unsolved board
Render each step of the algorithm RED for back and GREEN for Forward
Draw over existing board
DRAW FINISED BOARD
VISUAL REPRESENTATION (GUI)

Figure 1 : Python UML

Test Plan

Test ID	Test Description	Input	Expected Output	Result
T_1_HL01_01	'0' should be used to represent blank space indexes.	Unsolved sudoku puzzle	Displays the correct blank space indexes.	Pass
HL01_02	'0' should be used to represent blank space indexes.	Unsolved sudoku puzzle	Doesn't display the non-zero element's indexes.	Pass
HL02_01	Algorithm is used to check for validation in rows	Unsolved sudoku puzzle	No repeated elements for a particular row was displayed.	Pass
HL02_02	Algorithm is used to check for validation in rows	Unsolved sudoku puzzle	No repeated elements for a particular column was displayed.	Pass

HL02_03	Algorithm is used to check for validation in rows	Unsolved sudoku puzzle	No repeated elements for a particular grid was displayed.	Pass

Implementation Summary

The project is programmed in Object oriented style. The GUI.py file imports the Cube.py file as a module and uses its Class and methods to solve the puzzle. The GUI component of the project is contained in the GUI.py file. Entire code documentation generated using pDoc is present in the Documentation directory of the repository. Running the GUI.py file and hitting the SPACE key starts the execution of the program and one can visualize in real time how backtracking algorithm works.

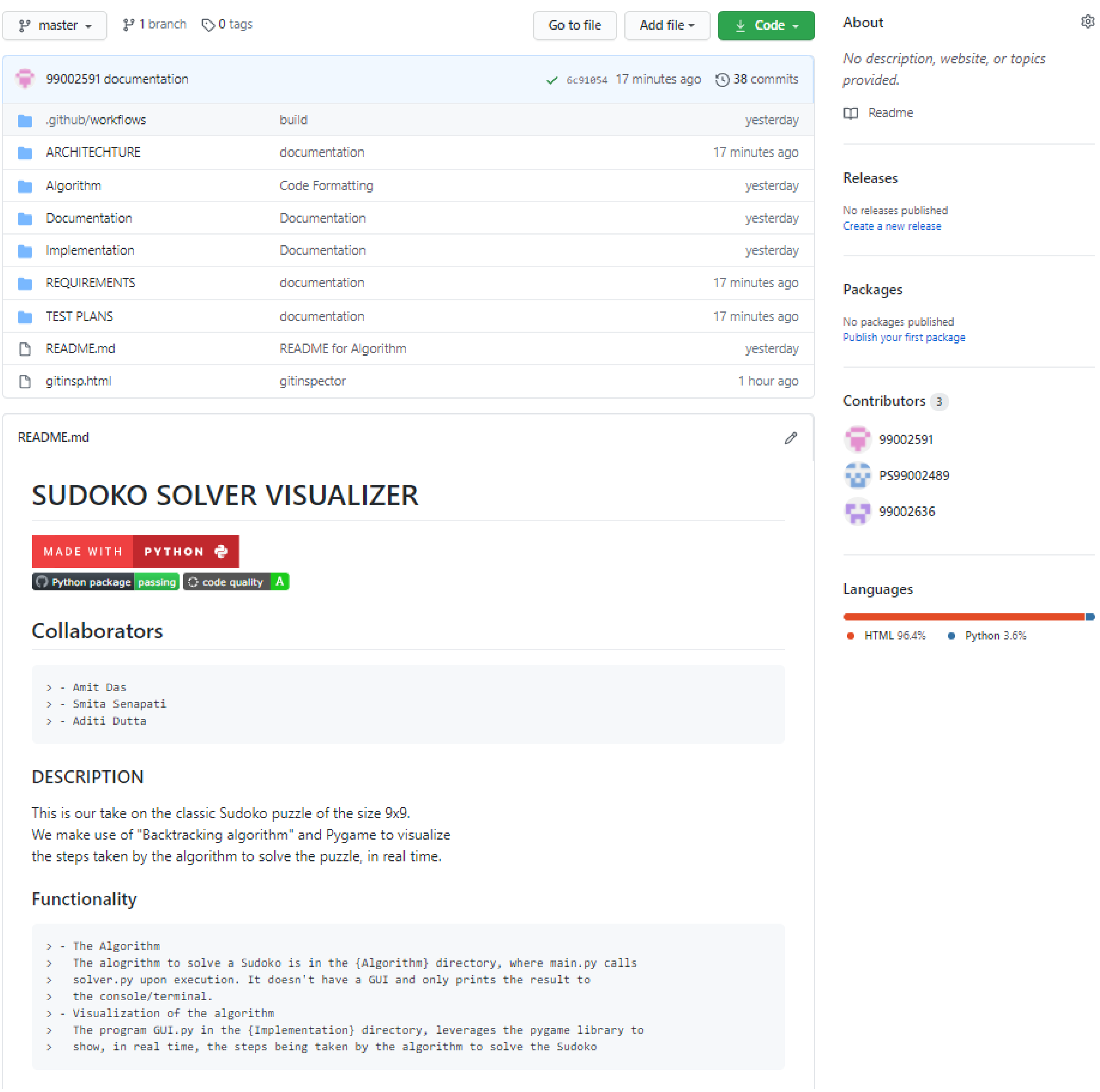
Video Execution

Link to video showing execution

Git Link

https://github.com/99002591/2009MYSEMB03-PYTHON-8

Git Dashboard



99002591 documentation ✓ 6c91054 17 minutes ago 38 commits

.github/workflows	build	yesterday
ARCHITECHTURE	documentation	17 minutes ago
Algorithm	Code Formatting	yesterday
Documentation	Documentation	yesterday
Implementation	Documentation	yesterday
REQUIREMENTS	documentation	17 minutes ago
TEST PLANS	documentation	17 minutes ago
README.md	README for Algorithm	yesterday
gitinsp.html	gitinspector	1 hour ago

README.md

SUDOKO SOLVER VISUALIZER

MADE WITH PYTHON

Python package passing · code quality A

Collaborators

```
> - Amit Das
> - Smita Senapati
> - Aditi Dutta
```

DESCRIPTION

This is our take on the classic Sudoko puzzle of the size 9x9.
We make use of "Backtracking algorithm" and Pygame to visualize
the steps taken by the algorithm to solve the puzzle, in real time.

Functionality

```
> - The Algorithm
>   The alogrithm to solve a Sudoko is in the {Algorithm} directory, where main.py calls
>   solver.py upon execution. It doesn't have a GUI and only prints the result to
>   the console/terminal.
> - Visualization of the algorithm
>   The program GUI.py in the {Implementation} directory, leverages the pygame library to
>   show, in real time, the steps being taken by the algorithm to solve the Sudoko
```

About

No description, website, or topics provided.

Releases

No releases published

Packages

No packages published

Contributors 3

- 99002591
- PS99002489
- 99002636

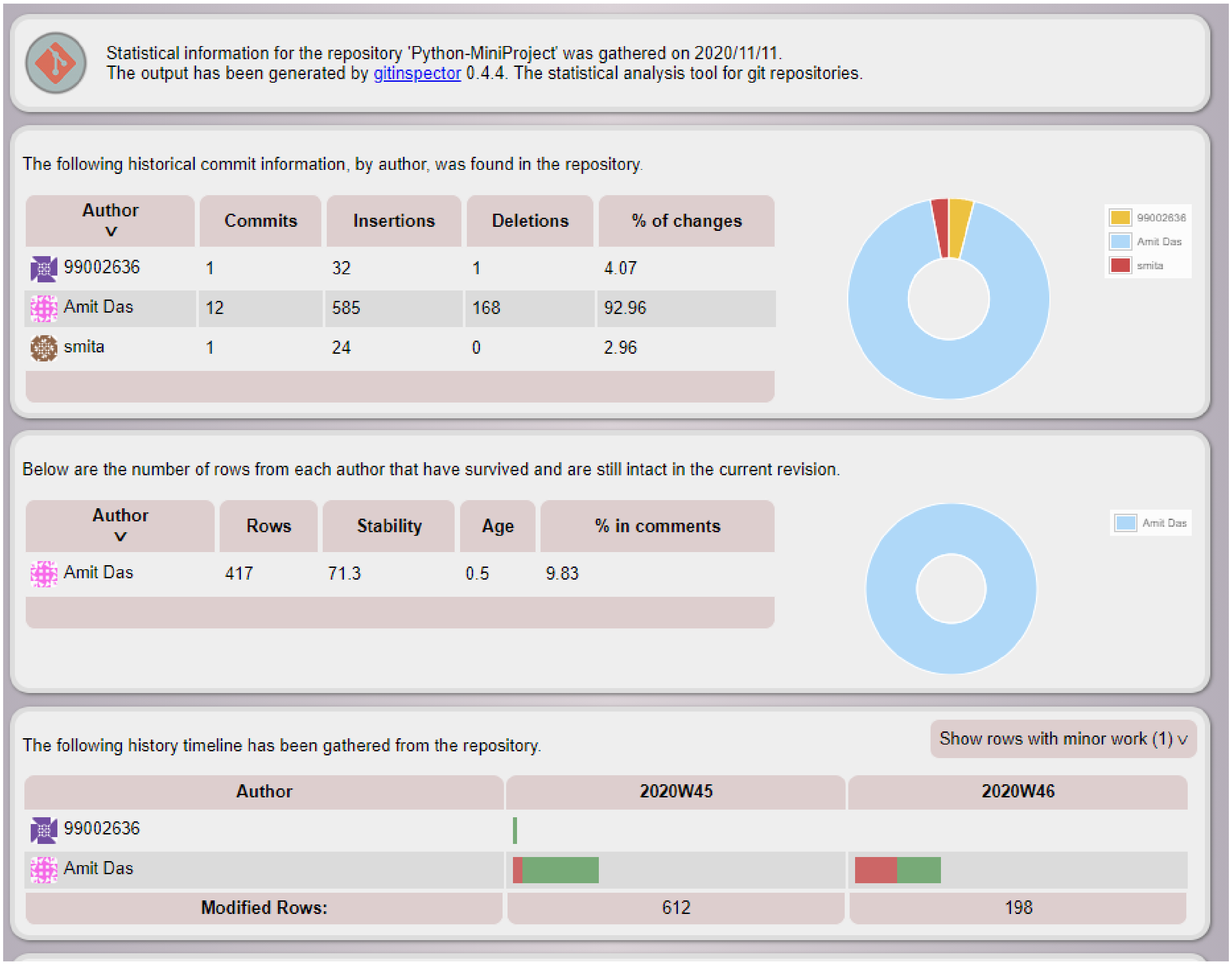
Languages

- HTML 96.4%
- Python 3.6%

Figure 2 : Dashboard Python

Summary

Git inspector summary



Statistical information for the repository 'Python-MiniProject' was gathered on 2020/11/11.
The output has been generated by gitinspector 0.4.4. The statistical analysis tool for git repositories.

The following historical commit information, by author, was found in the repository.

Author v	Commits	Insertions	Deletions	% of changes
99002636	1	32	1	4.07
Amit Das	12	585	168	92.96
smita	1	24	0	2.96

Below are the number of rows from each author that have survived and are still intact in the current revision.

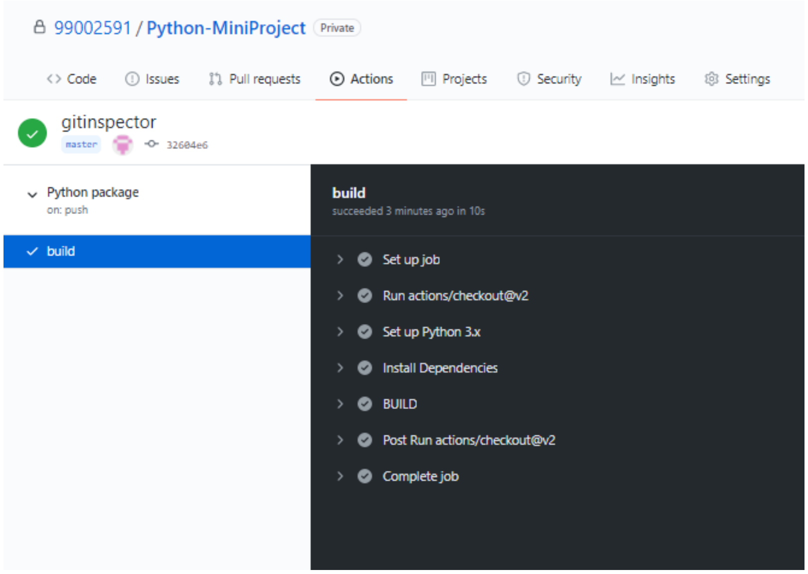
Author v	Rows	Stability	Age	% in comments
Amit Das	417	71.3	0.5	9.83

The following history timeline has been gathered from the repository.

Author	2020W45	2020W46
99002636	🟩	
Amit Das	🟥🟩	🟥🟩
Modified Rows:	612	198

Figure 3 : Gitinspector Python

Build



gitinspector
master 32684e6

Python package
on: push

✓ build

build
succeeded 3 minutes ago in 10s

- Set up job
- Run actions/checkout@v2
- Set up Python 3.x
- Install Dependencies
- BUILD
- Post Run actions/checkout@v2
- Complete job

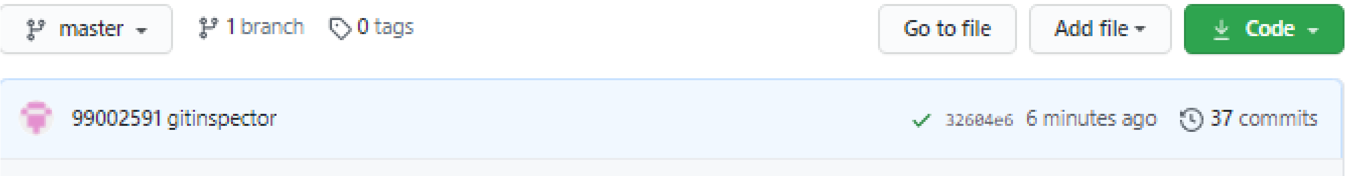
Figure 4 : Build python

Code quality

- Code quality was improved after taking into account the warnings thrown by Codeacy.
- No unused variables
- Use of easily understandable variable names.

Issues

- Gitinspector is not taking into account all the commits. While there are 37 commit until this time, gitinspector can only fetch about 14 commits.



99002591 gitinspector ✓ 32684e6 6 minutes ago 37 commits

Individual Contribution & Highlights

- Developed the backtracking algorithm.
- Helped in development of UI.

Challenges faced and how were they overcome

- UI development was a bit challenging. We tried to make a full-fledged GUI with pygame and tkinter but due to time constrains had to give up on ktinder.

Future Scope

- Addition of checking mechanism to see if puzzle is solvable before solving it.
- Functionality to play the game manually