STORY GENERATOR

PROJECT REPORT

by

SHUBHAM SHARMA
RK19PDB06 B06

MASWOOD AHMAD
RK19PDB70 B70

ANIKET
RK19PDB53

Department of Intelligent Systems
School of Computer Science Engineering
Lovely Professional University, Jalandhar

October, 2020

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we shall take full responsibility for it.

Shubham Sharma B06

Maswood Ahmad B70

Aniket B53

Place: Jalandhar
Date: 28th October, 2020

TABLE OF CONTENTS

BONAFIDE CERTIFICATE

Certified that this project report "STORY GENERATOR" is the bonafide work of "SHUBHAM SHARMA, MASWOOD AHMAD, and ANIKET" who carried out the project work under my supervision.

Ashish Shrivastava

Assistant Professor

ID - 25703

School of Computer and Engineering

1. INTRODUCTION

Story telling is an important component in the world of games. In most genres, a game's storyline is what captures the attention of gamers and it becomes the selling point of the game. Story telling can also play a huge role in a child's growth where it can be used as a form of education for the child. Making use of technologies, story generation can be made automated. With the advancement of technologies and artificial intelligence, there are more and more interactivebased and automatic story generators being developed. There is already usage of automated tools for plot development in the entertainment industry. Besides that, dynamic story generation can be used as a form interactive learning in classroom teachings or online-based educational sites. Children can engage and interact with the program in order to generate stories dynamically. This may help to attract a child's interest and promote creativity.

1.1.MOTIVATION AND BACKGROUND

There are various types of existing story generators, each can be categorized based on the story generation algorithm it uses. Some of the algorithms include the case-based algorithm, rolebase algorithm and data-driven algorithm. Each of these algorithms has its own benefits and limitations.

User interaction is important in maintaining a user's interest towards the story, especially in games for children. For example, the data-driven story generator generates sentences based on highest score between words' correlation. However, the generator does not have fluency and coherence between sentences. This makes the story progression feels inconsistent and does not have fluency. The user does not have control or influence over the story progression, as the story generator does not take in user input during the story generation process. This is also the case for other story generators such as case-based story generator or role-based story generator. The **motivation** behind this project is to come up with a story generation approach which allows more user interaction by keeping in mind the limitations of other techniques.

1.2. OBJECTIVE AND GOAL

This is the main objective of this project. In this project an approach which will allow more user interactions will be developed. This approach's aim is to promote more user interaction during the story generation process. In this user-driven story generation approach,

- Firstly users are given the choices to input the type of story that they are willing to generate, the available options are: the plot story, the Imagination story, and the Pokemon Battle story.

- All the stories have random templates which will print the new story every time the user enters the new choice.
- Based upon the entered choice the user will have to provide several simple input parameters to the story generator during initialization stage.
- Based on these inputs, the story generator will determine the story's template, characters and then generate the story.

The **goal** of this research is to develop a user-driven story generation approach which allows more user interaction. The final output is a story generator application implements the user-driven approach.

2. PROJECT DESCRIPTION

This section discusses about the development stages of the story generator and the details of the main components. Figure 1 shows the process flow of the project along with the modules.

Modules

The story generator project mainly consists of two main components/ modules:

- Story Planner,
- Story Renderer

2.1.Module 1: Story planner

Story planner makes use of information from the user input to determine the next sequence of questions as well as story progression. A fixed type story is used, with varying characters and actions depending on the user's input.

Choice input: The story planner first asks for the particular type of story to be generated which includes:

1. Plot story
2. Imagination
3. Pokemon battle story
4. Exit

Based on the entered choice the action is performed like, if the user enters the choice 1, 2, or 3: corresponding to that further actions are performed for story generation. However, if the user enters the choice 4 he/she will be exited from the story generator. In case the user enters anything apart from the choices, he will saw a message saying "*please enter a valid choice!"*. Based on the entered story choices (1, 2, or 3) the questions are asked from the user regarding the sentence rendering process for story generation process.

For example: the questions asked for the Imagination story are:

- Q1: enter the name of the enemy.
- Q2: enter the name of the father.
- Q3; enter the adjective for the enemy.

The responses to these questions are stored in the *enemy*, *father*, *and enemyadj* variables respectively which are further used in story generation process.

2.2.Module 2: Story Renderer/Generator

The next stage is, where the story generator prompts questions for the user based on his/her given story type input and then the given user input is used to determine the story progression. The sentence renderer is responsible for constructing sentences for the story generator. The variables change accordingly depending on the user-input. Some examples are:

- *enemy*: replaces the enemy of the self in *imagination* type story.
- *father*: replaces the father of self in *imagination type* story.
- *enemyadj*: replaces the behavior of enemy during sentence rendering.

Some of the sentences variables defined are:

- *prob2*: "all of a sudden a psychopathic" + ***enemy*** + "grinned at me showing all the razor sharp teeth. Suddenly it started to claw at my face. From the loss of blood, I collapsed onto the tough ground."
- *sol3*: "I forced my drowsy eyes open to the sounds of a" + enemyadj + " " + enemy + "licking my face."

After the sentence rendering process, the story rendering is done using the predefined templates and randomization process. The rendered sentences are printed randomly during story generation process. The "random" module of python is used to select the random sentences from a list of sentence variables defined.

For example: random.choice([intro1, intro2, intro3]) + random.choice([sol1, sol2, sol3]).

Replacing sentence variables with its random choices is part of the story generator's randomization function to keep the generated story different each time.

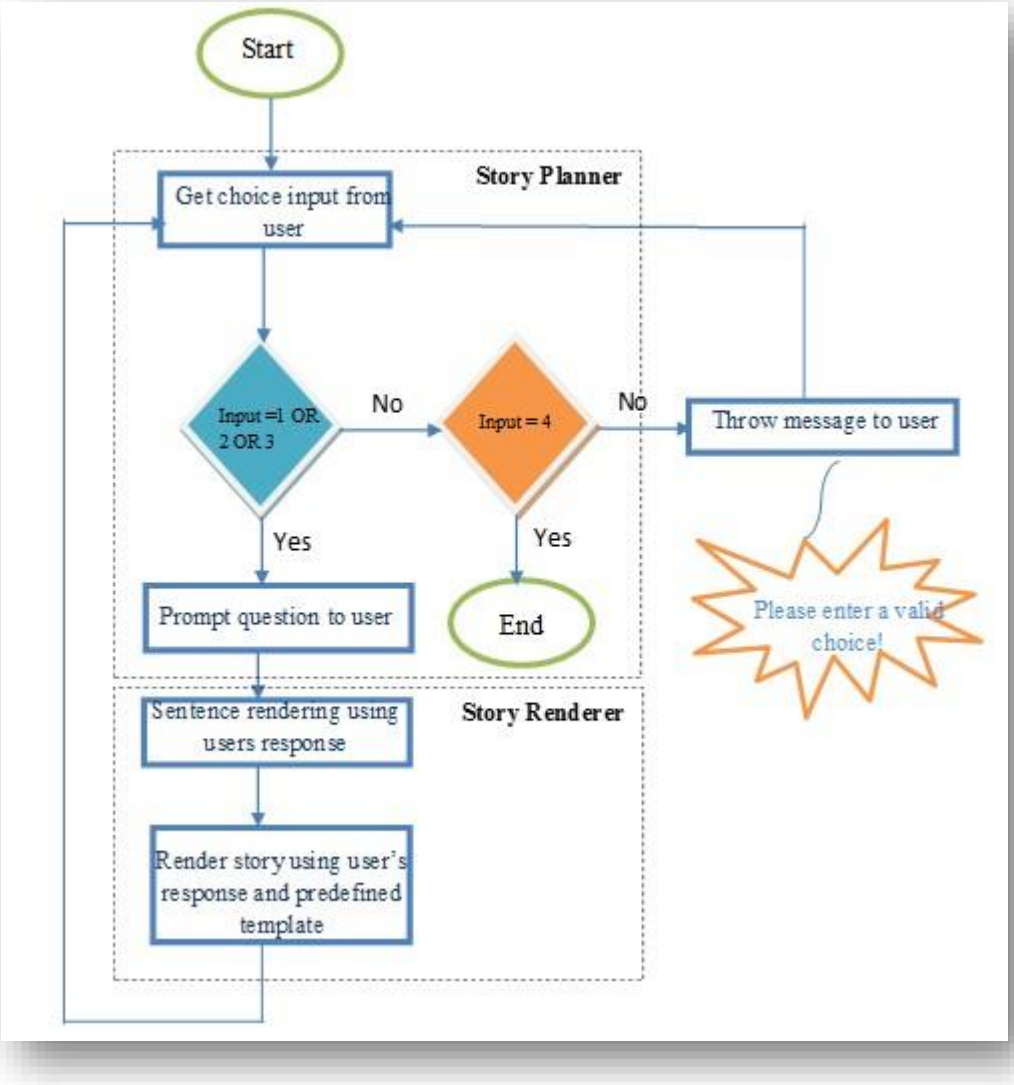


Figure 1: Modules and process flow of the project

3. STUDENTS' ROLES AND RESPONSIBILITIES

In this section, the project tasks required in order to complete this project are described along with the name of the students who have completed the tasks.

Analyzing available story generators	Shubham Sharma (RK19PDB06)
Analyzing available resources	
Reading sources to prepare templates	
Designing the method	Aniket(RK19PDB53)
Coding and implementation	
Testing the code	
Corrections in code (if they were)	Maswood Ahmad (RK19PDB70
Project report writting	

4. IMPLEMENTATION

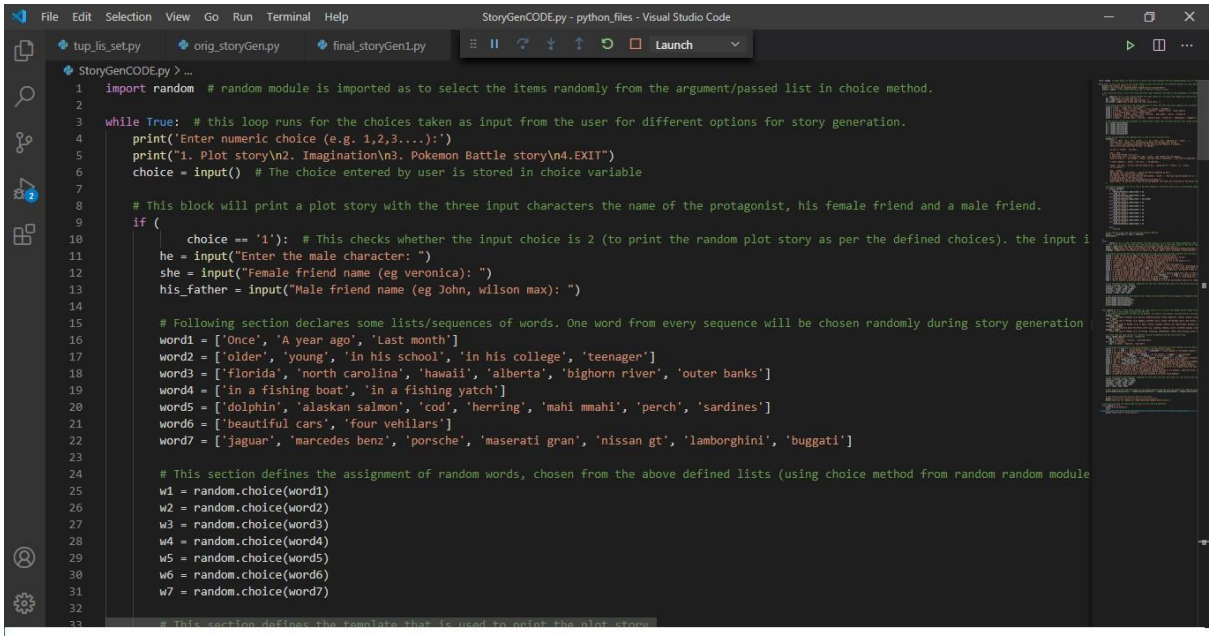
The implementation of the project is done using *python* using *Visual Studio Code* editor.

First of all the random module is imported as it carries the supporting code to randomly select the choices from a given list. The user is given a list of options to select the choice of story type. This process is continued until the choice pressed by the user is “Exit” (i.e. 4).

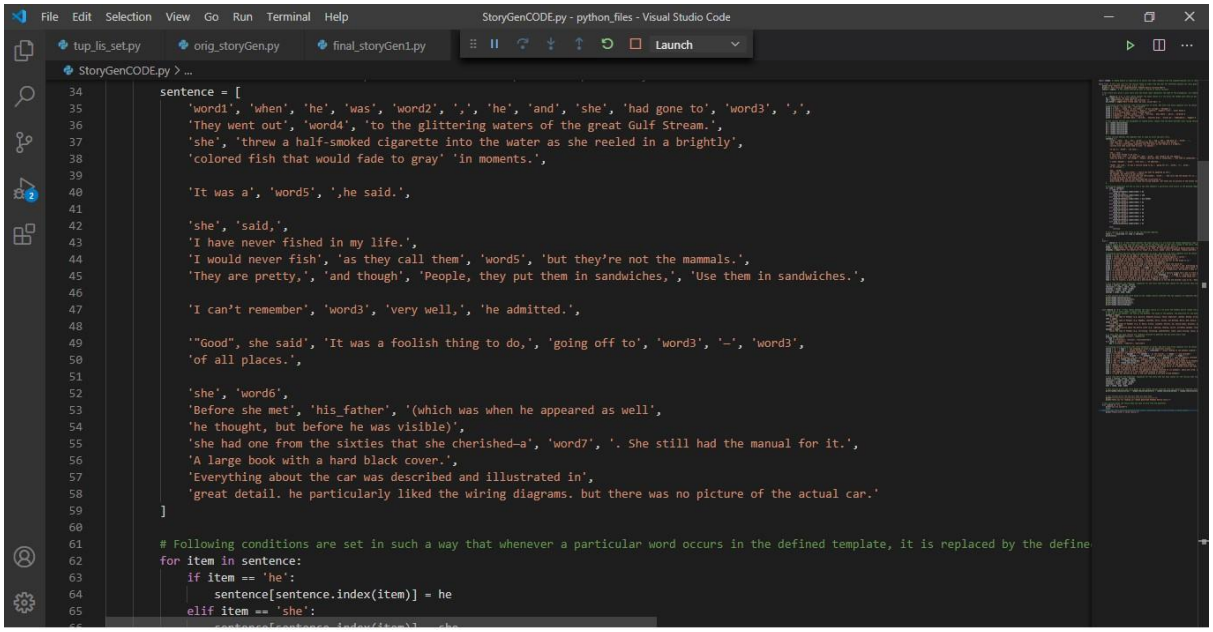
As the user selects the type of the story, some of the questions related to the story type selected by him/her are prompted.

Story type 1 code:

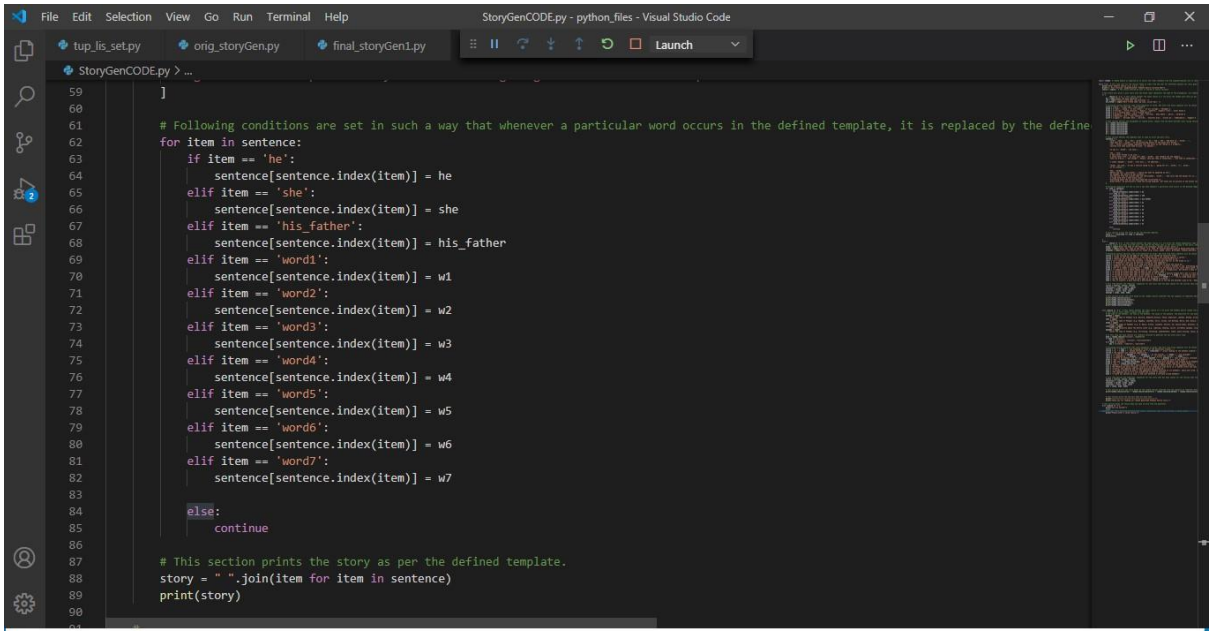
When the user selects the story type - 1, he/she will be asked for the questions related to the plot story as defined by the programmer. The code for story 1 is shown in following figures. Three variable “he”, “she”, and “his_father” are taken to take the input from the users. Some of the lists are also prepared to randomize the story generated every time. Thus every time the user press choice for the same type of story, the variables- *word1, word2, word3, word4, word5, word6, and word7* are used by random choice method to randomly select the words. *W,. w2, w3, w4, w5, w6, and w7* are used to store the results obtained from the random choice method.



The following figure shows the template for the story type 1 (ploy story). The variables in the template will be further replaced by the defined words and user inputs.

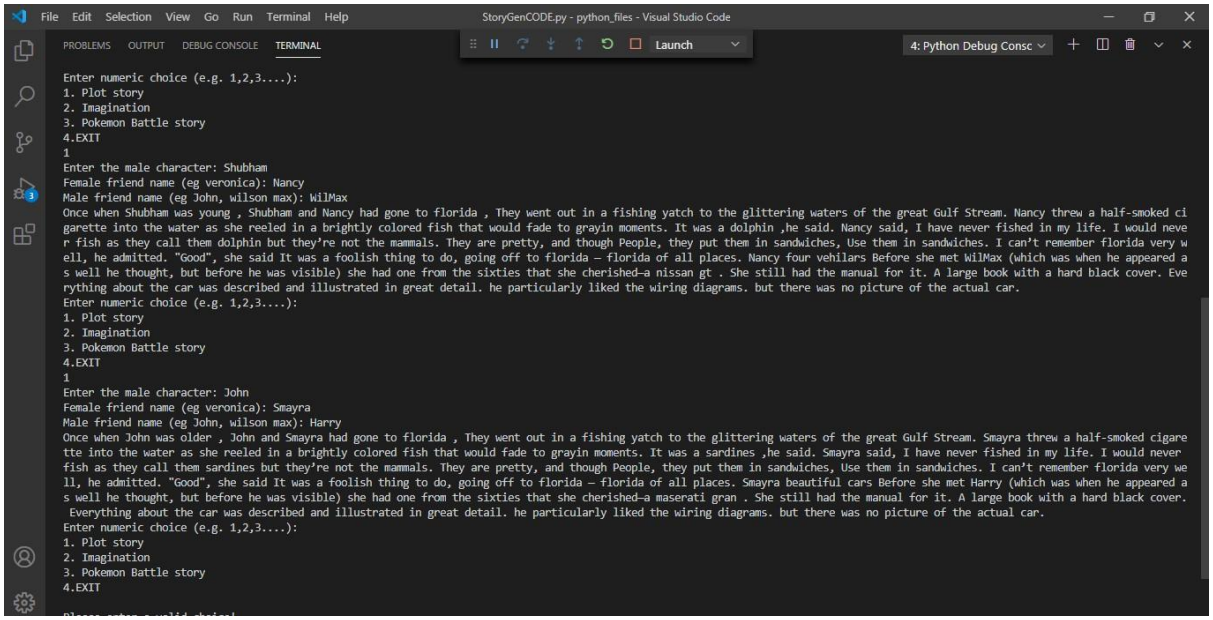


The input words and the predefined words (selected using randomization method) are used to be replaced in the story template during story generation process.



Output:

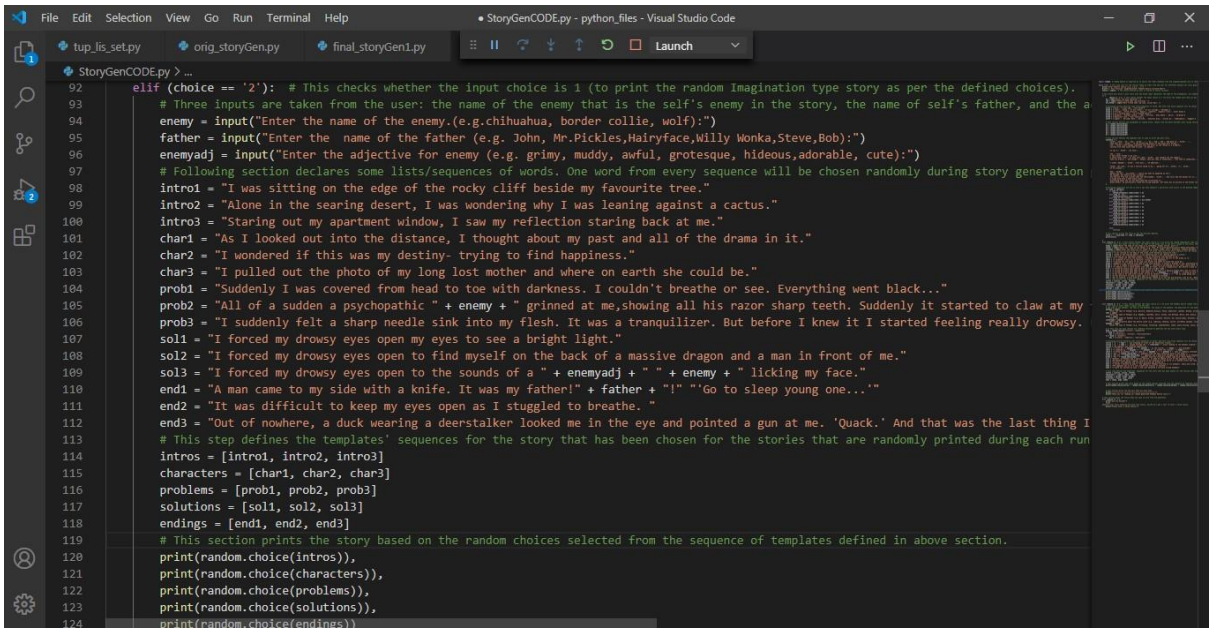
As the user selects the type of the story as 1. (Plot story), some of the questions related to the story type selected by him/her are prompted. Which can be shown in the following figure. Once the users gives the answer to the questions as per his/her requirements, the generated story is reflected on the screen.



Story type 2 code

When the user selects the story type – 2 (Imagination), he/she will be asked for the questions related to the plot story as defined by the programmer. The code for story 1 is shown in following figures. Three variable *enemy*, *father*, *enemyadj* are taken to take the input from the users. Some of the lists are also prepared to randomize the story generated every time.

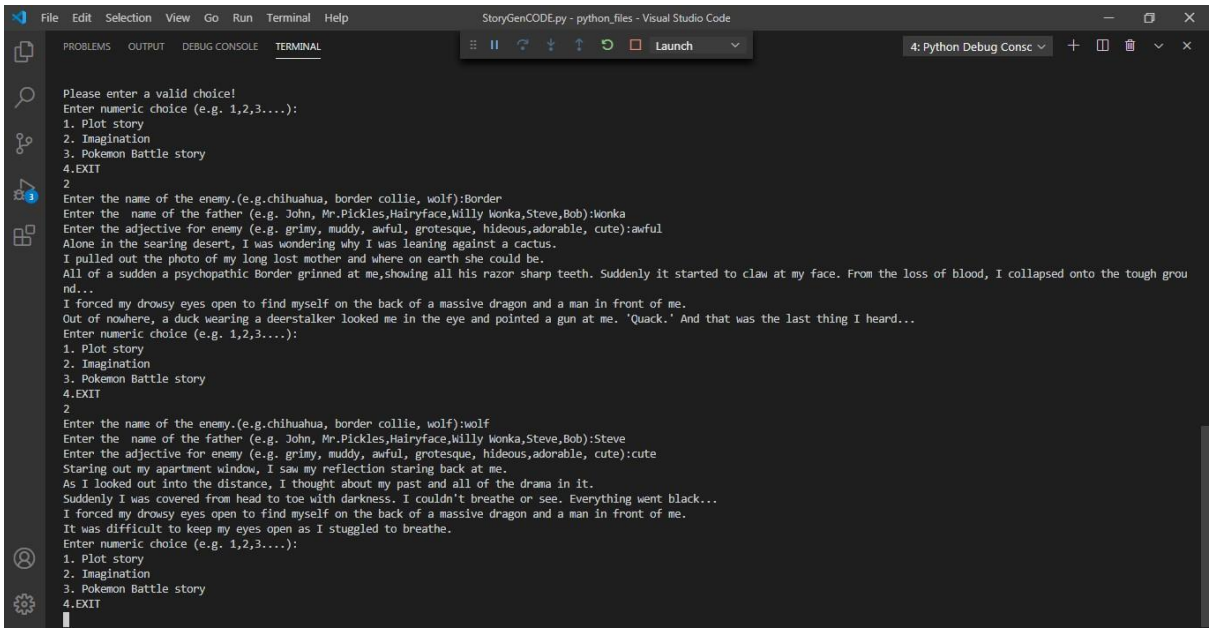
The input words and the predefined words (selected using randomization method) are used to be replaced in the story template during story generation process.



Output:

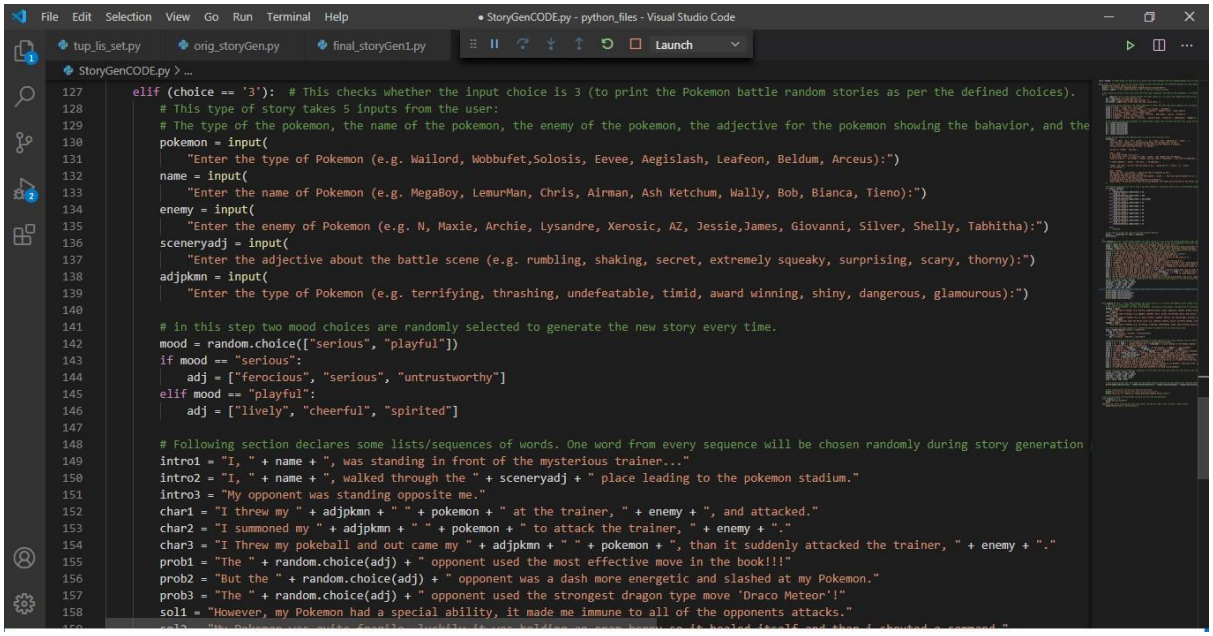
As the user selects the type of the story as 2. (Imagination), some of the questions related to the story type selected by him/her are prompted. Which can be shown in the following figure.

Once the users gives the answer to the questions as per his/her requirements, the generated story is reflected on the screen.

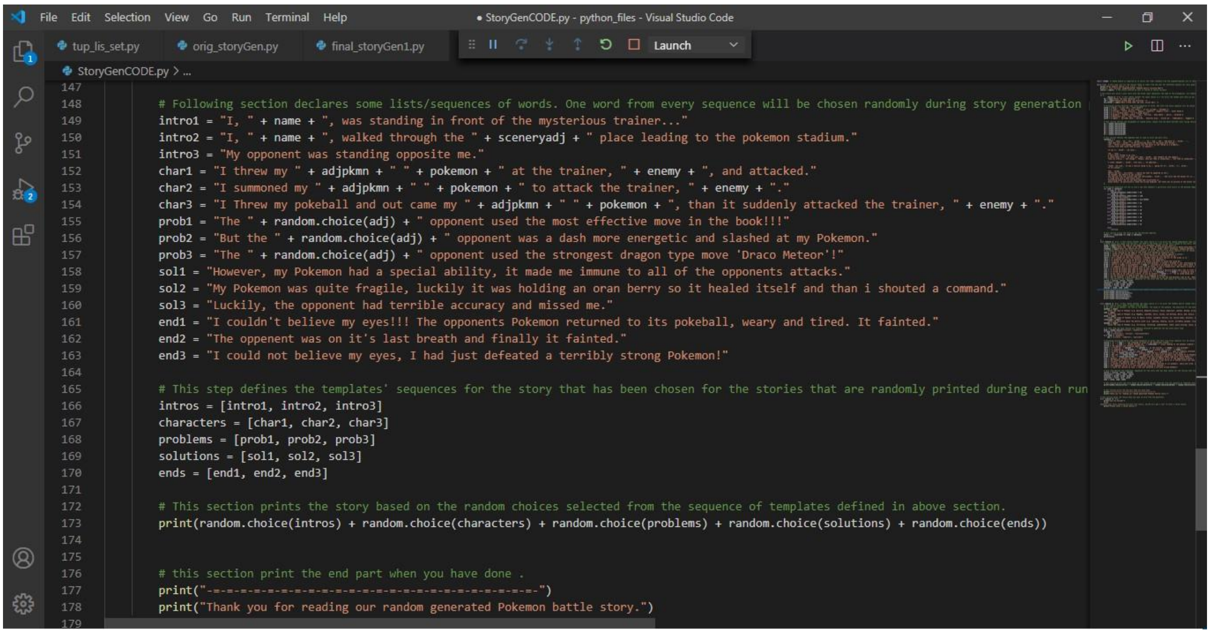


Story type 3 code

When the user selects the story type -3 (Pokemon Battle story), he/she will be asked for the questions related to the plot story as defined by the programmer. The code for story 1 is shown in following figures. Five variables: *pokemon*, *name*, *enemy*, *screenadj*, *and adjpkmn* are taken to take the input from the users. Some of the lists are also prepared to randomize the story generated every time.

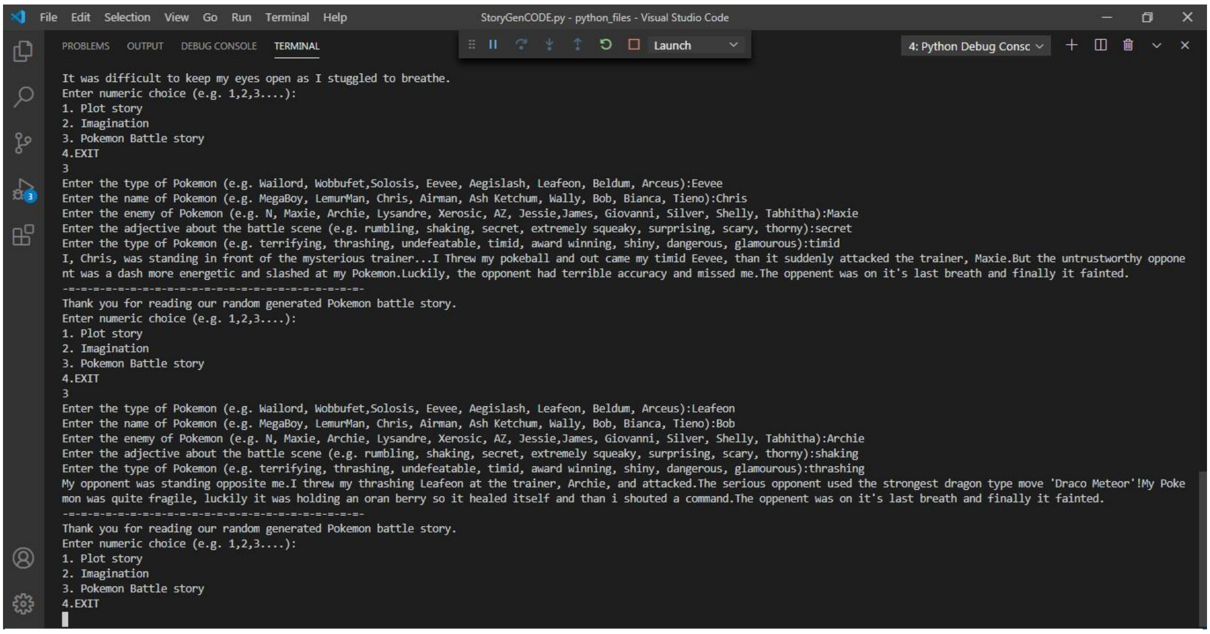


The input words and the predefined words (selected using randomization method) are used to be replaced in the story template during story generation process.



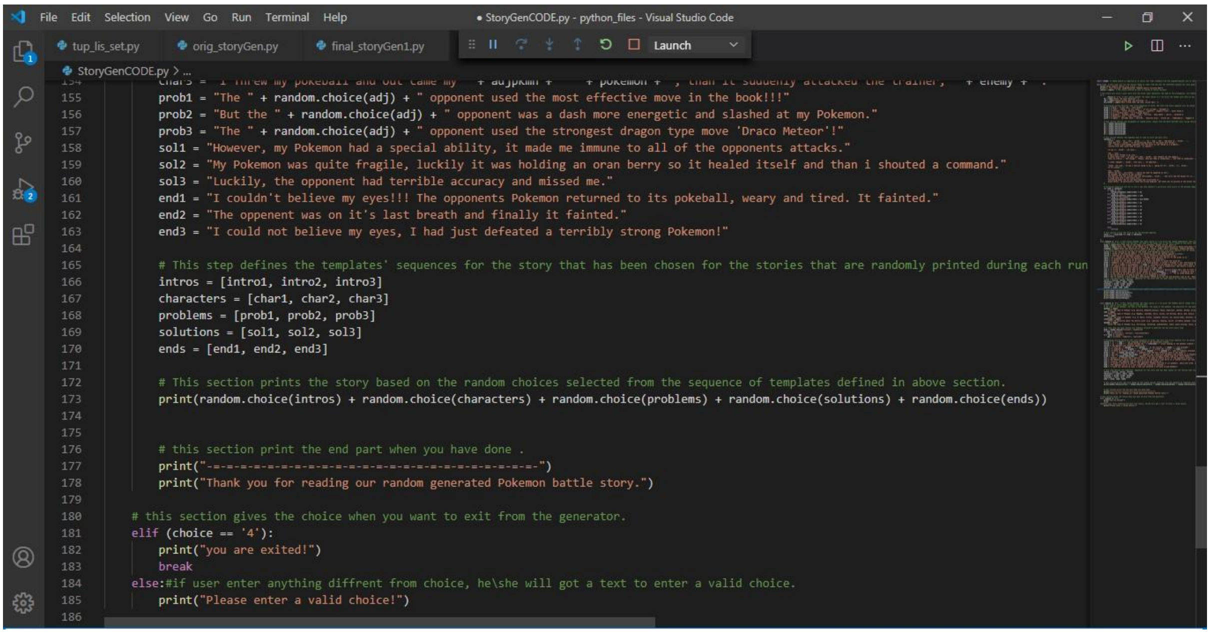
Output:

As the user selects the type of the story as 3. (Pokemon Battle story), some of the questions related to the story type selected by him/her are prompted. Which can be shown in the following figure. Once the users gives the answer to the questions as per his/her requirements, the generated story is reflected on the screen.



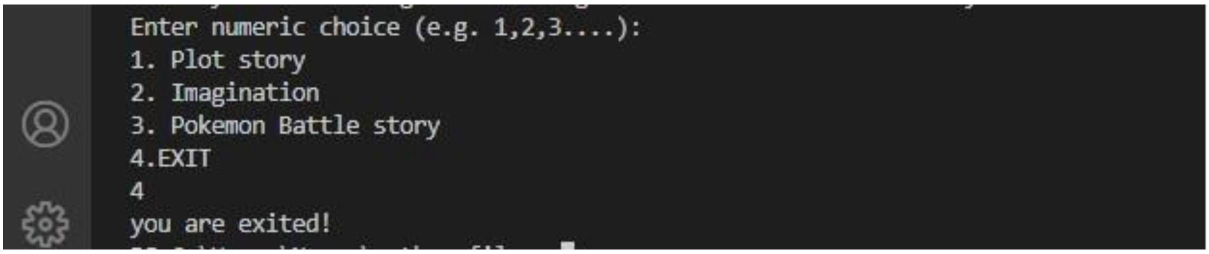
Exit code

As the choice inputted by the user is to be 4, whole loop is terminated and the user is exited from the story generation process.



Output:

As the user selects the choice as 1. (Exit), he/she is exited from the story generation process and they cannot make further choice for the story generation process until next run.



5. TECHNOLOGIES AND FRAMEWORK

Language: Python 3.8.6.

Editor: Visual studio code

6. SWOT ANALYSIS

Strength:

- This project is responsive to the user's requirements as it takes the inputs from the users to generate the story. Thus, the users can frame the story as per their choice of characters' name and their characteristics.

Weakness:

- This project has three types of stores only, which generates different stories randomly using the user's input. Thus, the project can be programmed with more complex story plots, types and iterations for the user inputs.

- The output of the programmed project is shown on the output console of the editor. A more interactive GUI can be provided to the implemented code's output fields.

Opportunities:

- This approach (user driven) can be applied and implemented into educational games for children. The approach allows children to think and provide responses which will then alter the generated story.

Threat:

- Threats related to this projects are that as it generates the stories randomly by taking some inputs fields from the user, as the templates are predefined, the inputs may not fit in the logics of templates, thus may give the story as output which does not satisfies the user logically.