DS Mini project-2020

Batch 2018-22

Synopsis and design document

Project Name: Alternate Reality

Abstract:
It is an interesting story related project with different endings of a story. The different endings are selected by the users. In this java related concepts are present. The **TREE** concept is the major data structure present in this project.We basically use the concept of binary tree for this project.For every possible way the user can violate the story plan, an alternative story plan is generated.The user can build up their own story. The user can build up the story according to their preferences. In every step there are choices for users to select their next step of the story. In this way the story is continued till the end. The story can have a number of conclusions according to what the user chooses. Every option chosen leads to a twist in the story. Games of this kind have already been made and are popular around the world. In a way, this project can help the user to enhance their thinking skills and boost their creativity.

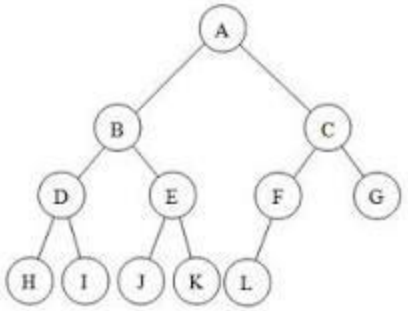
What is new:
In this project, we will implement our own creativity in creating options of different versions of a story for the user to create. In fact, there have been discussions on how binary trees can tell a story. However, what we plan to do is create a game in which the user gets to create their own story from the given options to them.

Modules:
We are going to use C++. The whole project is based on the concept of Tree DS. We are going to import various C++ libraries. The major library we are going to import is the Standard Library. The following are the functions we are going to include in our project.

1. Create nodes
2. Display parent nodes and children nodes
3. Insert a scene of a story
4. Delete a scene of a story
5. Reverse the story
6. Print the story
7. Count the number of scenes (length of tree)

Architecture/ Design Diagram:
The data structure we will be following is a complete binary tree. In this tree, some nodes would not have their child nodes while, other nodes may have child nodes. Using binary tree we will create options for users to select in order to create their own story. Initially, there will be a root node. The root node is the start of the story. Then later we create parent nodes from the root node and from parents nodes create their children nodes and so on. The building of story will have

questions or words that are asked to the user and the user will have to choose an option. The user will also be allowed to make changes in their story or restart.



Hardware and Software Requirements:

1. Laptop with preferably Windows OS
2. DEV C++

Team Members Tasks and Responsibilities:

1. Arundarasi Rajendran (18070122081):
    - Research on tree DS
    - Code
    - Improvements
2. Aditi Goyal (18070122003)
    - Story writer
    - Code
    - Design Algorithm
3. Pinnam Laxmi Priyanka (18070122043)
    - Research on tree DS
    - Code
    - Bug fixes
4. Sri Neelima Chinta (18070122070)
    - Story writer
    - Code
    - Design Algorithm

Gantt Chart:

Task	Duration	Start Date	End date
Gather Requirements	**3**	**15-Feb-20**	**18-Feb-20**
Research on Tree DS	**7**	**20-Feb-20**	**27-Feb-20**
Design Algorithm	**2**	**28-Feb-20**	**1-Mar-20**

Coding	**24**	**1-Mar-20**	**25-Mar-20**
Test	**1**	**25-Mar-20**	**26-Mar-20**
Bug fixes	**2**	**26-Mar-20**	**28-Mar-20**
Improvements	**4**	**28-Mar-20**	**1-Apr-20**
Final Testing	**2**	**1-Apr-20**	**3-Apr-20**
Submission	**26**	**5-Apr-20**	**1-May-20**

References:

1. https://www.cc.gatech.edu/~riedl/pubs/riedl-aiide05.pdf
2. https://www.researchgate.net/publication/302360069_The_Binary_Tree_Roll_Operation_Definition_Explanation_and_Algorithm?_esc=publicationCoverPdf&el=1_x_2&enrichId=rgreq-4619542b50c490647c2189a7dc407192-XXX&enrichSource=Y292ZXJQYWdlOzMwMjM2MDA2OTtBUzozNTk2MjE2NTE5MTA2NjZAMTQ2Mjc1MTg5MTk3MA%3D%3D
3. https://academic.oup.com/comjnl/article/34/5/438/553944
4. http://cslibrary.stanford.edu/110/BinaryTrees.pdf
5. https://users.cs.duke.edu/~rcd/papers/1996_sigcse.pdf
6. http://cslibrary.stanford.edu/110/