Ryan Mahoney

Citadel: A Note Taking Tool for Narratives

Problem:

Many stories gain their followings as a result of immersive storytelling and rich details. For example, a person reading George R.R. Martin's series A Song of Ice and Fire is flooded with details that make every plot point complex and each event meaningful.

However, when reading a narrative involving many characters, events, and details it can become difficult to recall how these entities interact. A reader of Martin's works may struggle to maintain a grasp on the many families, betrayals, and all-important deaths. Taking notes, as in any discipline, can help an individual better understand what is going on and why. There are benefits to creating diagrams in particular, such as :

- Family trees can help to visualize the structure and lineage of a family
- Timelines can help to visualize the order of events in a straightforward fashion
- Taking personal notes on the storyline can help a user reflect on what happened and how they felt

These methods come at a slight cost. It can be a taxing process to re-read, scan, and then efficiently plan out the distribution of this data. An error in memory or understanding during the plotting process may result in a faulty visualization. Logistic and user oriented errors can deter someone from pursuing a solution that may promote their understanding of the source material.

Thesis:

The aforementioned graphs are important tools, but actually putting them together may deter some people. It is the aim of this project to develop a solution that encourages and streamlines note taking and diagram making.

Developing a personalized curriculum seems like an effective solution, but is fairly difficult to execute on a large scale. What may be a more effective solution is to develop an accommodating software tool. This could be done by parsing user input, translating that input into machine-readable code, using that code in a way that creates graphs, and presenting the data concisely. It is more likely that an individual will get into a habit of taking notes if most of the organizational work is simplified in this way.

Graphviz, a graph and diagram construction program, is open source and capable of constructing the graphs desired. Unfortunately, one needs a working knowledge of the command line environment and its input language, DOT, in order to use it effectively. For the grand majority of people, these are limiting factors that make it unusable.

The aim of this project is to develop an intuitive interface for Graphviz, store user notes over time, and enable repeated and comfortable use of the tool to an average user. Outlined below are the requirements and components required to compose this tool, as well as a set of milestones ordered by their importance and feasibility.

Important Software Requirements:

- Graphviz: A program capable of organizing DOT language into diagrams, which shall be leveraged to create family trees, timelines, etc.

Important Terms:

- Entity: A person, place, or event in the storyline. These are defined by the user.
- Media: Used to describe the article the user is taking notes on. This may be a movie, book, time in history, etc.

Minimum Viable Product:

1) The user’s notes shall be recorded and stored in a parseable and low upkeep format. This content shall be organized to represent each entity’s characteristics
2) The ability to add entities and media information to each document shall be implemented to suit an average user. This shall be done using a simple GUI capable of displaying and adding entity information

3) The GUI shall showcase all of the entity notes already made and act as an interface for the graph making software
4) Relationships between characters shall be stored, with parentage taking initial priority. This is done to easily construct family trees, discussed below
5) A back end program shall parse the format used to store notes. The user shall indicate which elements need to be gathered for a given circumstance
6) The output obtained from parsing the note files shall be converted into DOT language via another back end program. This program shall consider that relationships between entities can be translated into DOT edges and groups
7) Family trees of two different varieties, one immediate and the other extended, shall be implemented. An immediate tree shall show the parents, grandparents, children, and siblings of a particular character. An extended tree shall show the immediate tree, as well as aunts and uncles, cousins, and extended generational lines for a character.
8) Timelines shall be constructed by taking user defined events as input. The events shall be organized by their relative position to other events (i.e. one comes before the other). The time unit relevant to the media (years for reality, chapters for books, episodes for television, etc.) shall be displayed as a secondary characteristic. This is done to be as inclusive as possible for the different varieties of a narrative
9) “Chapter” notes provided by the user shall be organized into a timeline and juxtaposed with the media’s description. This is meant to help the user compare their personal opinions with the content of the section
10) A graph showing the relationship between events and locations shall be added, to indicate how important some places may be to the subject media
11) A “participants” characteristic shall be added to the event entity. These individuals may be added from the pool of characters or by adding an entry

12) The GUI shall allow a user to copy the characters from one media entry to another. This considers that many series have the same characters repeat from entry to entry
13) A basic web page showcasing the program's capability shall be developed

Supplementary Goals:

1) Color Coding shall be added to family trees to indicate if an individual is alive, dead, or of an unknown vitality
2) A note type called a counter shall be implemented. This will allow a user to keep track of how many times a specified trigger occurs
3) In relation to the individual chosen, titles shall be added to each node of a family tree to indicate how they are related to the individual (i.e. aunt, uncle, father, etc.)
4) The user shall be capable of giving image links to be associated with entities. Images shall be stored as links in each entity's note, then displayed in each graph. Without a link, an entity shall have a placeholder image provided.
5) User defined titles that translate to a hierarchical or tree-like structure can be implemented. For instance, most military institutions follow a structure

Long Term Goals:

1) An installer, capable of providing a new user with the program and its components, shall be developed
2) The program's visual components (color, layout, etc.) shall be adjusted to satisfy regular users