CSC309 Individual Final Project Proposal

Tian Yue Liu

1004743753

skyjoy.liu@mail.utoronto.ca

Project Name: VeryGraphical JS

My proposed JavaScript library aims to provide a tool that can easily generate various visual maps using given data. Some of the examples include: single/double elimination brackets, round robins, and family trees. In the library, it will include both functions to generate skeletons for these structures as well as functions to alternate their appearances, including size, color, and position in the viewport. As a developer, there sometimes exist the need to showcase the input data in an intriguing, comprehendible way. With this library, it saves the need to manually plan out and physically code the layout of the graphs. For a front-end user, based on developers' specific design, should be able to input relevant data and be presented with a graphical representation (diagrams). Use cases include: 1) a clearer representation for "IMLeague", a tournament info website that I have personally used. 2) a family tree on "Wikipedia" for the British Royal family. 3) a family tree to present my own small family for self-introductions.

For alpha release, I have implemented that family tree functionality of my library. The users of my library can easily use my library to create a family tree graphics on the browser. Currently, the functionalities include:

1) makeMember –create a member on the family tree given the person's name, and generation
2) markDeceased - indicate on diagram that member with given id is deceased
3) specifyGender – indicate on diagram the gender of a member
4) addAge – add and display the age of the member with given id
5) addBirthday – add and display the birthday of the member with given id
6) addDeathday – add and display the death date of the member with given id
7) addGeneration – add and display the generation of the member with given id
8) drawTree – draw the family tree on the browser using DOM manipulate

Link to deployed page:

https://very-graphical-library.herokuapp.com/examples.html

The page contains two family trees, created by codes that can be found in https://very-graphical-library.herokuapp.com/examples.js. User can interact with the family tree by clicking on a member, and his/her full information should be displayed on the bottom left

Currently, the structure of the library contains two objects: FamilyTreeMaker, and FamilyMember. FamilyTreeMaker stores much of the functionalities, while also having common information for a family. FamilyMember stores personal information of a specific member.

FamilyTreeMaker:

```
{
        family_name: “Bezos”,
        population: 17,
        members; [ FamilyMember objects ],     // array of FamilyMember objects
        youngest_generation:  5,
        members_by_generation: { “1”: [ FamilyMember objects ], “2”: [ FamilyMember objects ] … },
 }
```

FamilyMember:

```
{
        name: “Jeff”,
        last_name: null,//this is specified if different from family name
        family_name: “Bezos”,
        family_id:  5,     // index of the person in the family, 5 means 5th person in the family
        age: 50,
        generation: 2,
        birthday: 12/25/2000,
        diedOn:  “Currently Alive”,
 }
```

Initially, the element for a family member is a div with his/her name only. Through library functions, more elements, such as age, generation, and birthday are added to/removed from the html body. This is done using “createElement()”, “appendChild()”, ‘removeChild()”, and other DOM manipulating methods. The information being then displayed come from the attributes of the objects mentioned above. Detailed functionalities as of right now can be found in previous section.

For the final submission, I plan to implement functionalities that allows the user to clearly identify sibling/partner/parental relationships between family members. In addition, I also plan to include another graph generator, Elimination Match Result maker, on top of the Family Tree Maker. If time allows, I will also add in graph makers for round robin matches and knock out matches. The challenging part of the final submission will be finding ways to show relationship between html elements, such as drawing a line connecting them or arrows pointing from one to another.