Meeting Jan. 18th 2016

Meeting with Dr. Osgood and Geoff McDonnell
- Two members in our group have taken CMPT 394 (Michael R and someone else) and know of a process called consultative diagramming and using a qualitative model
  - Build diagrams in a constructive way that how a system in the world might work
  - It's a big jump to go from what you want to go to a simulative model
  - Diagrams are good for showing what a simulation model might contain
  - The physical diagram is awkward and is hard to collaborate with
    - This is where this project comes in handy
- This program is to allow to build these qualitative models
- https://insightmaker.com
  - Explore Insights
    - search "health care"
      - click "Responses to Service Work Pressure"
        - Gives us an example of one of Geoff McDonnell's works.
        - Gives an example to a qualitative model
          - BUT no one can collaborate with it at the same time
    - click "Work Pressure"
      - Notice how they add things in to show how each item added is related to certain things.
- Is there any programs that do this?
  - Well, yes! https://kumu.io
- The blue torso and head in the qualitative model are called an agent.
  - Show states of a person
  - Contains things inside of it. If dragged they come with it
- In the course of a semester we won't get all of the features but we have a list of what needs to be added in terms of priority.
- The people in the circles are images the client wants to drag into the program.
  - In other words having the ability to drag images into the program would be useful.
- "Don't worry about types of transitions here"
- Osgood opens up a program called "AnyLogic Professional"
  - Shows an example of a state chart
    - Should be an entry point, and some exits
  - Shows a stock and flow diagram
    - You have stocks (squares) and they go to either a stock or cloud connected by flows.
    - Flow should drag with stock if stock is dragged.
    - Osgood then shows the diagram become a simulation in action but we don't have to worry about doing that.
- This program should be used as a drawing program and as well not be just for professionals but the normal user.

- The program should be collaborative: to be able to join online since people live in different parts of the world.
- One person will highlight and comment about something in google docs
    - In this program people should be able to collaborate and discuss things via comments on the diagram so they can get feedback on how to improve the model.
- Added new features that they want near the bottom of the list [which is not as crucial to get]. Should be available to use in the updated document.
- "I would stress the importance of existing mechanisms for **operational transformation**"
- https://operational-transformation.github.io/ot-for-javascript.html
    - Build on top of this there is no point of starting from scratch.
    - Strongly looking at existing tools for this
- http://www.loria.fr/~ignatcla/pmwiki/pub/papers/IgnatCSCW06.pdf
- One thing that would be good to have are permissions (can edit, can comment, can read) and the links should determine this.
- It should be noted that things on the list are not linear and the order we do them in.
- It should also be noted that the stakeholders will change priorities over time and it's very common.
- STRONGLY recommend to look at the syllabus together
- STRONGLY recommend to do things in small features and talk often to Osgood about it.
- Also do not neglect server side things
- Talk to Geoff often. He lives in Australia.

Questions we Have

Q: What would the deliverable look like?
A: A webapp. Ask to set up servers with the computer science department. It would be accessible to people around the world.

Q: What should we use in languages and technologies?
A: Strongly suggest to use operational transformation. Web technologies require not only browser based components but server based components. We will need a database, things that will talk from the client to server side, server side scripting. **GO**, **JSP**, **Apache**, and **Tomcat** are some languages to consider. If there is someone really knowledgeable about GO or frameworks with java. We also need a serious multi-user database. **Redmine** for issue tracking is a good thing to use. Testing frameworks is recommended to use. If we are using Java assertions are built in.

Q: How would you advise to start?
A: The key thing is to start small to do **spike prototypes** if necessary. What he means by spike prototypes is test one features to see how it works and then throw it away. The point is getting knowledge to know what each feature is, nailing down the features via code so that when we program the main program we aren't stuck. This will eliminate a LOT of risks. A lot of googling is needed in other words. Don't plan a huge system right now.

Q: Do we need to make it where each color is represented by a color and see if we can see the action (example we see the drag happening)?
A: He never considered colors for users but it's considered to be a good idea to have. It's up to us as well if we want to show the actions in action.

Attendance:
Corey, Shane, Royce, Jordan, Michael K., Michael R., Angela, James, Mack