**Team \*|∅**

Stock Market Simulator/Analyzer

Project Name: Yggdrasil

Chris Weir - This Week

Tony Simonutti - After Chris

Cody Doyle- After Tony

Tong Liu - After Cody

**11/13/2014**

**Actions**

* Plugin generic class now works
* Implemented config files for some of the plugins
* Yggdrasil dynamic assignment for ports updated
* Database fetch from webpage works
* Stock query from webpage works
* Python CGI seems to be working nicely
* Login and Register Pages completed
* Web interface mostly built <3

**Risks**

* Haven’t developed algorithms for stock prediction. Original deliverable seems out of reach given our accumulated knowledge of the subject. ~~May have to revise this deliverable or remove it to make room for testing to avoid the tar pit.~~ Yea, this isn’t going to happen, too much networking needs to get done, higher priority
* Avoid requesting too much data and getting Yahoo mad at us
* Sockets isn’t hard to set up, but Yggdrasil is turning into a fairly complex message routing application, which may pose a risk to our deadlines...
* Plugins will only be generic, nothing too special
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Work on graphical display of stock data.
* Pull together the web interface
* Work on Yggdrasil (implementing server protocols and routing)
* Fix Yggdrasil threads
* Clean up code, remove silly debug stuff
* Get installer to make default configs

**11/13/2014**

**Actions**

* Networking handshakes fully implemented for transferring information between servers
* Working on graphical representation of Stock Data
* Moving plugins into a factory structure

**Risks**

* Haven’t developed algorithms for stock prediction. Original deliverable seems out of reach given our accumulated knowledge of the subject. May have to revise this deliverable or remove it to make room for testing to avoid the tar pit.
* Half our team has sold out to real companies.
* Avoid requesting too much data and getting Yahoo mad at us
* Sockets isn’t hard to set up, but Yggdrasil is turning into a fairly complex message routing application, which may pose a risk to our deadlines...
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Work on graphical display of stock data.
* Pull together the web interface
* Work on Yggdrasil (implementing server protocols and routing)

**11/10/2014**

**Actions**

* Yggdrasil server coming along slowly
* finished exams

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* Sockets isn’t hard to set up, but Yggdrasil is turning into a fairly complex message routing application, which may pose a risk to our deadlines...
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Work on Yggdrasil (implementing server protocols and routing)
* Pull together the web interface
* something

**11/6/2014**

**Actions**

* Continued work on Yggdrasil (routing and protocols)
* EXAMS!!!!!!!!!! (AAAAAAAAAA)
* Started linking HTML together
* Implemented base classes for components

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* Sockets isn’t hard to set up, but Yggdrasil is turning into a fairly complex message routing application, which may pose a risk to our deadlines...
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* 5th iteration ends: November 10th. Objective is to implement graphical data representation using database data. → second half of project (first half was completed for stakeholder #1)
* Work on Yggdrasil (implementing server protocols and routing)
* Link together HTML pages
* Implement socket communication between Yggdrasil and web interface
* Reformat code to be implemented with classes
* WORK ON (and hopefully finishing) THE EXAM

**11/3/2014**

**Actions**

* Assigned work for pushing forward with Yggdrasil
* Worked on Yggdrasil server (designing routing and protocols)
* Worked on our exams

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* Sockets isn’t hard to set up, but Yggdrasil is turning into a fairly complex message routing application, which may pose a risk to our deadlines...
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Work on Yggdrasil (implementing server protocols and routing)
* Link together HTML pages
* Implement socket communication between Yggdrasil and web interface
* Reformat code to be implemented with classes
* WORK ON THE EXAM

**10/30/2014 (and 10/23/2014 since class was cancelled)**

**Actions**

* Finished stake holder 1
* Tony actually made it to class (Yay!)
* Implemented Handskhaking between implemented plugins
* Wrote generalized networking between plugins
* recorded our meetings during stake holder 1 for stake holder 2

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* May be difficult to set up sockets communication with Yggdrasil
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Continue working on mainpage.html (Use PHP to fetch information from MySQL Database)
* Continue working on Yggdrassil (Write data recieved from Bifrost to MySQL database)
* Continue to fix general HTML issues (CSS code not centralized)
* Design general Plugin class and start implementing current plug ins to be of this class

**10/27/2014 (and 10/23/2014 since class was cancelled)**

**Actions**

* Created Odin login page (PHP to database)
* Odin portfolio wireframed
* Odin XML creator interface runs on a web server and produces XML file and re-routes form data to python/php CGI file for socketing to Yggdrasil main program
* Proof of concept databasing plugin created
* Yggdrasil handshake protocol defined
* Bifrost implements handshake protocol
* Began implementation of Yggdrasil class
* We have CRC cards!
* Individual components proof of concepts completed :D
* Tony went to a job interview

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* May be difficult to set up sockets communication with Yggdrasil
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Finish creating
* Implement socketing handshake between plugins
* Write generalized networking methods for classes

**10/20/2014**

**Actions**

* Worked on chunking
* Researched databasing with python
* Worked on HTML interface which is turning out awesome
* First iteration is complete

**Risks**

* Avoid requesting too much data and getting Yahoo mad at us
* May be difficult to set up sockets communication with Yggdrasil
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Work on interface to Yggdrasil
* Experiment with databasing
* Begin working on second iteration which consists of combining the modules into one, fluid interface

**10/16/2014**

**Actions**

* Added support for multiple markets
* Strongly leaning towards using bootstrap for frontend
* Determined framework and formatting standards for Bifrost feed handler (user inputs desired information via web app (javascript), that creates a standardized XML file of desired fields that can be read by main Bifrost program which pulls the desired stock information and finally saves it to a predetermined XML file for storage)
* Worked on designing chunking algorithm. Turns out it’s hard.

**Risks**

* More features/mechanics from Yahoo/Google Stocks API that might have been missed or misunderstood  
  ~~Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)~~
* Bifrost implementation currently in python. May be difficult to set up sockets communication with C++ Yggdrasil
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Keep working on Bifrost feed handler: implement features to access past stock information, implement chunking
* Experiment with databasing

**10/14/2014**

**Actions**

* Submitted elaboration
* Successful weekend meeting in which we determined framework of our project
* Successfully implemented rudimentary Bifrost program that pulls csv formatted stock information from [REDACTED] finance API given a list of stock tickers in XML and updates an existing XML file containing the stock information
* Successfully implemented simple GUI that fetches and displays stock information
* Group is quickly becoming more familiar with Yahoo! stocks API

**Risks**

* More features/mechanics from Yahoo/Google Stocks API that might have been missed or misunderstood  
  ~~Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)~~
* Bifrost implementation currently in python. May be difficult to transfer to C++ (especially GUI)?
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Keep working on Bifrost feed handler: implement features to access past stock information
* Research options for web UI frameworks
* Experiment with databasing

**10/9/2014**

**Actions**

* Successful Wednesday Meeting
* Finished 80% on use cases
* Almost done with elaboration
* Tinkered with feed handler
* Finished domain model

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Finish elaboration
* Submit elaboration
* Assign coding work (Bifrost feed handler) for individuals for the weekend
* Research options for web UI frameworks

**10/6/2014**

**Actions**

* Worked on use cases
* Worked on elaboration
* Made up a secret handshake

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Finish elaboration
* Get the ball rolling on code (feed handler)
* Try not to die

**10/2/2014**

**Actions**

* Worked on use cases
* Tinkered with encoding for retrieving ticker fields
* Took 2nd round of exams
* Created basic UML diagram

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Finish elaborating on use cases
* Work on elaboration
* Sleep

**9/29/2014**

**Actions**

* Made a list of use cases
* Had a meeting with everyone present
* Went to the career fair
* Tinkered with Yahoo API (wget to pull stock data over http)

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Finish elaborating on use cases
* Work on elaboration
* Sleep

**9/25/2014**

**Actions**

* Made UML Diagram
* Had meeting with almost all members present

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Some plugins in the schedule may be more difficult to implement than originally intended and may take longer than the allocated time period
* Must take care to not violate legal terms for use of free stock data

**Plans**

* Begin working on Use cases
* Set up Gantt chart
* Start using scrum meetings
* Tinker with APIs and http stock data requests

**9/22/2014**

**Actions**

* Finished Inception
* Created SVN branches
* Had first successful meeting with all members present

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Must take care to not violate legal terms for use of free stock data
* Group members use different coding environments

**Plans**

* Begin first major code deliverable - Bifröst feed handler
* Revise inception deliverables as necessary
* Create a UML diagram
* Set up Gantt chart
* Start using scrum meetings

**9/18/2014**

**Actions**

* Got a UML tool
* Built a makefile
* Tinkered with make
* Read coding standards
* Worked on personas
* Discussed scope
* Decided on naming convention for modules
* Decided on a project name

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API (will make progress once we have a chance to start tinkering with some basic code)
* Must take care to not violate legal terms for use of free stock data
* Group members use different coding environments

**Plans**

* Finish inception
* Create a UML diagram
* Create SVN branches (dev, qa, prod)
* Next meeting is Saturday afternoon

**9/15/2014**

**Actions**

* Had a meeting at new meeting time on Saturday afternoon
* Defined coding standards <http://www.possibility.com/Cpp/CppCodingStandard.html>
* Started drafting a framework (paper and pencil)
* Decided on make as a build tool
* Extended User Scenarios
* Fixed folder bug in SVN repo
* Started working with make

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API
* Must take care to not violate legal terms for use of free stock data
* ~~Group still not acclimated with basic market concepts~~
* Group members use different coding environments

**Plans**

* Each group member will acclimate themselves with make
* Go from paper and pencil to UML tool
* Build a test makefile as a group
* All users will read coding standards and begin adopting them
* Create Use Cases based on the personas and scenarios
* Expand on the scope of the project and modules we want to implement

**9/11/2014**

**Actions**

* Finalized project idea (stock market analyzer)
* Group familiarized themselves with svn
* Set up a weekly meeting time (Saturdays at 4:30pm)
* Languages chosen: C++, C (Can add more as needed)
* Defined scope: Project will process stock data, and store that data in a database; needs a convenient method of accessing data.
* Decided on Peer Review Metrics

**Risks**

* Group still isn't acclimated with Yahoo/Google Stocks API
* Must take care to not violate legal terms for use of free stock data
* Group still not acclimated with basic market concepts
* Group members use different coding environments

**Plans**

* Define Coding Standards
  + Many existing standards are overly complex for a project this size
  + May define our own? - the one Tong found looks good
* Create a graph of different components and how they will interact (UML)
* Begin framework of project
* Decide on a build tool - make

**9/8/2014**

**Actions**

* Set up Trello
* Set up Google Code (with Subversion)
* Set up Google Group
* Brainstormed Project ideas
* Shared contact info
* Had first meeting (Conferenced in sick group member)
* Picked MIT open source license to avoid RPI intellectual property issues
* Set up a gantt chart interface linked with Trello

**Risks**

* Not yet acclimated with Yahoo/Google stocks API
* Not yet familiar with basic market concepts
* Not yet familiar with Subversion
* Learn OS specific functionality

**Plans (for next week)**

* Start our project/Finalize Project idea
  + Choose Language(s)
  + ~~begin working on framework (divide into components, put on gantt chart)~~
  + Define scope of the project (roughly)
* Set up a weekly meeting time <Done>
* Learn how to use Google Code (Play around with it)
* Define coding standards
* ~~Decide on build tool (GNU Make)~~
* ~~Create project schedule/gantt chart (using Ganttify with Trello)~~