**Sorcerers Cave Project 3**

Author: Eric Sabelhaus

Date: 07/8/16

Professor: Nicholas Duchon

**Table of Contents**

1. [Design Documentation](#Design_Documentation)
2. [Users Guide](#Users_Guide)
   1. [Cave Explorer](#Cave_Explorer)
   2. [Search The Cave](#Search_The_Cave)
3. [Test Plan](#Test_Plan)
4. [Comments](#Comments)
5. Design Documentation

* A complete UML document of the game package is located in the doc subdirectory of the packaged source code.
* The title of the UML document is: EricSabelhausSorcerersCaveUML.png

1. User's Guide: description of how to set up and run your application

* Setup of this application requires at least Java version 1.7
* In order to run the application, the following command must be run within the bin directory of the compiled application code:
  + 1. java -classpath bin game.GameGUI

Once the application is running, select one of the three available caves to explore. Listed below:

1. SmallSimpleCave.txt
2. A\_BigData.txt
3. LargeCave.txt

After you’ve selected your cave, you will be presented the sorcerers cave Graphical User Interface (GUI)

Going from top to bottom, the first element you see is the main set of navigational tabs, which is followed by the content presented on the selected tab. At the very bottom of the GUI is a section of the jobs being performed by individual creatures within the cave.

Within the tabbed section, there are two tabs available, and each serves a distinct purposeThe first tab, which you are presented when the GUI opens, is the Cave Explorer tab. It can be best utilized by following these instructions:

* + Cave Explorer
    - This tab provides an easily navigable tree view of the entire cave, as well as the ability to sort cave elements by various attributes (discussed later).
    - When a part of the cave is selected within the tree, that cave element will be presented on the area to the right which initially says “Try selecting a node from the tree!”
      * If you’ve selected a party, you will be presented with a textual representation of all cave elements within that party which is tab delimited
      * If you’ve selected a creature within the tree (within a party or not), you will be presented with a customized panel which allows editing of that individual creature within the cave.
        + You may select one or more of the attributes to modify using the form which is presented, and the content will immediately be updated once you press the “Update Creature” button
      * If you’ve selected a treasure or artifact, you will be presented with a text panel containing all known details about the item.
    - Within this tab, there is a sorting menu which is located above the tree, here is a brief instruction on how to sort the cave using that menu:
      * Creature – Sorts every creature of every party
        + Age – sorted numerically
        + Empathy – sorted numerically
        + Fear – sorted numerically
        + Height – sorted numerically
        + Name – sorted alphabetically
        + Weight – sorted numerically
      * Treasure – Sorts every piece of treasure on every creature in every party
        + Value – sorted numerically
        + Weight – sorted numerically
    - Once a set of selections has been picked, pressing the Sort button will reorder the elements of the text accordingly

The second tab available within this view is the Search The Cave tab. It can be best utilized following these instructions:

* + Search The Cave
    - Within this tab, you’re provided a way to search for specific things within the cave, and whatever elements match your search parameters should be returned in a textual format for further inspection.
    - The top bar of this tab provides the searching functionality, and to the left of this top bar is the button to actually submit your search.
    - To the right of the submit button is the search selector, where you may choose to search by Name, Type, or by Index.
    - To the right of the search selector is the input field where you can submit your search parameters
    - Once a search is executed using the submit button, you will be presented with the search results
      * When a search is executed, the content of the item searched for will be displayed in result text field which is just below the top bar of this tab. If an illegal search parameter is entered an appropriate popup will notify of the change necessary to perform the search.

1. Test Plan:
   1. Sample Input
      1. The following is the sample input provided during testing:

// p:<index>:<name>

p : 10001 : Unity

// c:<index>:<type>:<name>:<party>:<empathy>:<fear>:<carrying capacity>

c : 20001 : Woman : Lucy :10001 : 17 : 22 : 20

c : 20002 : Woman : Jane :10001 : 22 : 15 : 25

c : 20003 : Worg : Brandon : 0 : 30 : 21 : 0

// t:<index>:<type>:<creature>:<weight>:<value>

t : 30001 : Gold : 20001 : 50 : 2000

t : 30002 : Gold : 0 : 75 : 5000

t : 30003 : Gems : 20002 : 10 : 10000

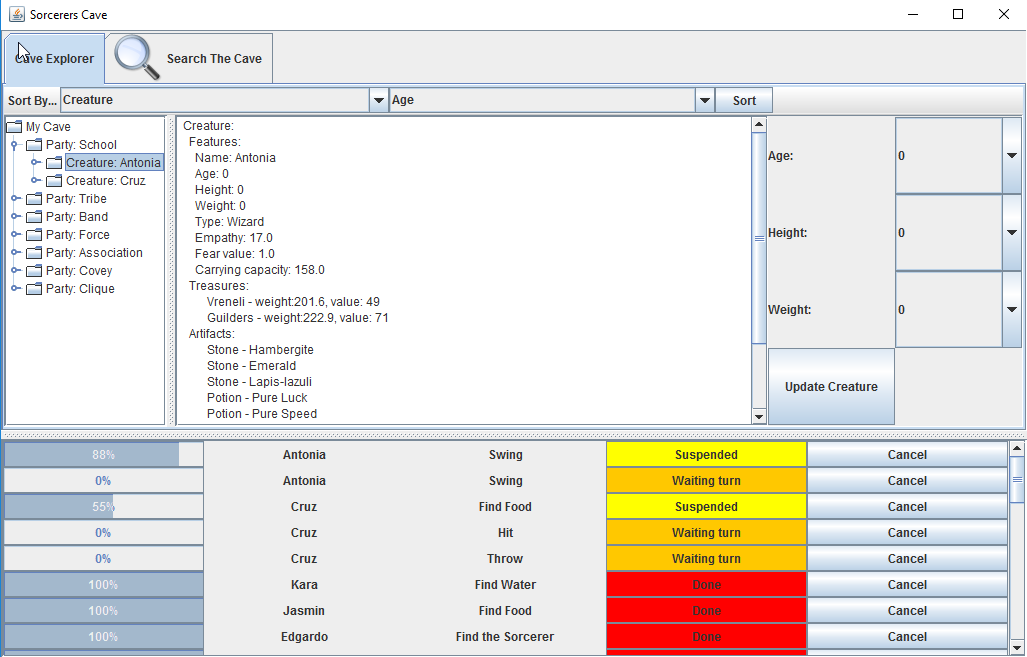
// a:<index>:<type>:<creature>[:<name>]

a : 40001 : Wand : 20001 : ElderWand

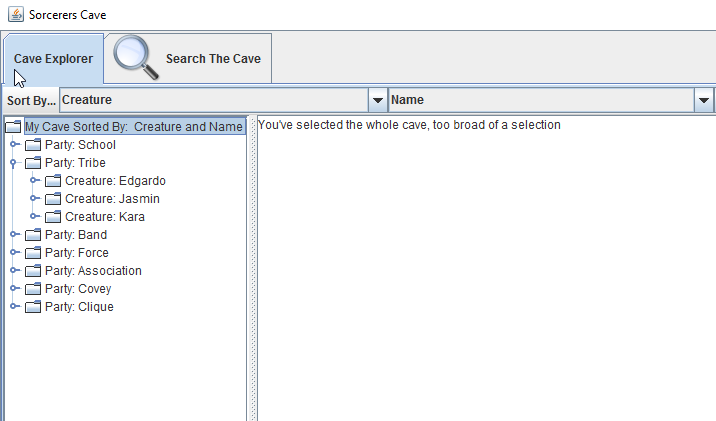
// j:<index>:<name>:<creature index>:<time>[:<required artifact:type>:<number>]\*

j : 50001 : Swing : 20001 : 5.00 : Stone : 0 : Potions : 0 : Wands : 1 : Weapons : 0

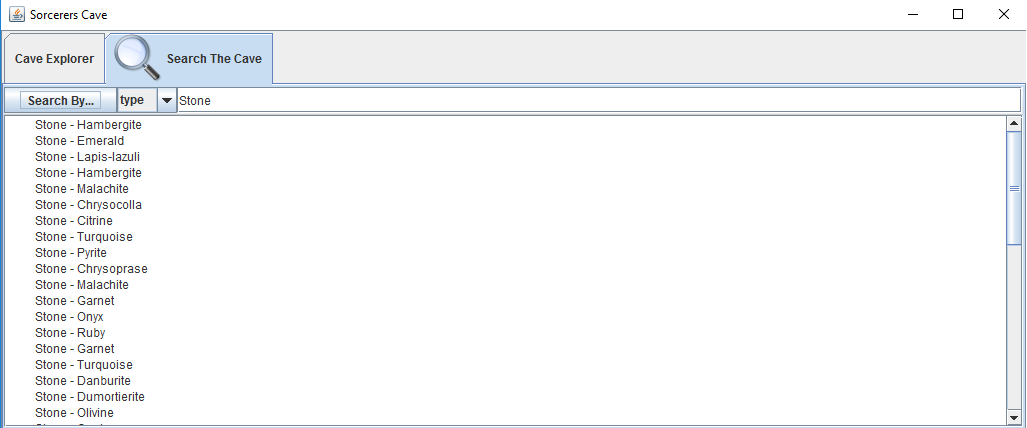
* + 1. The expectations are as follows:
       1. One Cave is created
       2. One Party is created
          1. Must belong to the one cave
       3. Two Creatures are created
          1. Both must belong to the one party
       4. Three Treasures are created
          1. One owned be each creature
          2. One must be undiscovered
       5. One Artifact is created
          1. Must belong to the first Creature
       6. One Job is created
          1. This Job must have and index of 50001
          2. This Job must be assigned to Lucy
          3. This Job must require the correct resource
       7. Before sorting Lucy should come first
       8. After sorting Jane should come first
  1. Result
     1. Assertion tests using Java JUnit testing framework succeeded in 0.480 seconds
  2. GUI Test Cases:
     1. Picking LargeCave.txt using JFilePicker. File contains no undiscovered items. After file loaded, all parties are present in tabs with no undiscovered items or creatures present as expected
     2. Clicking within empty space on tree does not return exception, instead prints statement in console
     3. Clicking on the My Cave element returns: “You've selected the whole cave, too broad of a selection”
     4. Single clicking on element within tree returns expected element



* + 1. Sorting Creature by Name re-renders the tree view showing “My Cave Sorted By: Creature and Name” as the root instead of “My Cave”
    2. After sorting above completed, Eduardo should be listed as first creature in Tribe



* + 1. Search by name for character named Kara returns expected creature.
    2. Search by Name for foobarbang returns expected standard output. Pressing
    3. Search by type of Stone returns all Stones as expected.



* + 1. Search by Type for foobarbang returns expected standard response.
    2. Search by index of 20002 returns Creature Kara as expected.
    3. Search of non-existent index returns standard response.
    4. Search of index (below lower bound) returns standardized response.
    5. Search of index (above upper bound) returns standardized response.
    6. Search with empty string returns correct standardized response.
    7. Threads maintained in all possible states using buttons in Job section to control a creatures’ ability to work on jobs

1. Comments:
   1. I did have a little trouble getting my locking working, and did decide against a semaphore for this application, as it was overkill and unnecessary. I instead used a Mutex strategy to lock a resource to a single thread, and used the tryAcquire method to check whether it was available. Turned out really nice!
   2. I had a blast taking this course! Thanks for all the great pointers!