C++ Java GUI Interface

- Idea: Use C++ to create a Java GUI, that is, call functions in C++ to build a Java GUI
- Only should require a single .h file to be included by the C++ class and works assuming the Java program is also running
- Should be Windows and Mac OS compatible
- Seamless to the programmer using the interface with either OS
- Dr. Mood intends to use this project during CSC254 next year
- Contact: Dr. Benjamin Mood

Oracle Invocation API
JNI Tutorial
Graphical Java Front End to a C++ Program
Wikipedia JNI
Changing a Java GUI Dynamically
http://stackoverflow.com/questions/1843205/are-there-tools-to-integrate-java-and-c

Notes from Dr. Mood

- First assignment requires one single c++ header file
    - Doesn't matter if pc/mac/linux
- Can make calls, if i have a java window open, the calls I make do things in the java window
- If you write Java commands in c++, it will pipe those commands over to the java window to display the output
- Need to write a c++ library of functions, then use an IPC to transfer those commands to Java
- C++ program has two phases
- Overall idea is to be able to write commands in c++ and display this stuff in java
- **Don't use sockets**
- Java program will also be written
- Java program is not always being run
- IPC: Interprocess communication
- JPanel hierarchy
- Clicking things in Java will send back to C++ program, which deals with the event.
- Setup phase
    - Clicking on things in the Java program causes information to be sent to the C++ program
    - The C++ program will do something with it.
- Loop phase
- Make our own API with some kind of list
- First thing: probably graphics stuff
- Top level frame with graphics component
    - Draw line, change color etc

- Second level
    - After getting basic structure,
    - JButton?
- Make our own API
    - Pass in the name of the object and what you want to do
    - Another example: Clear command from C++ allows Java frame to be cleared
    - Ex: JPanel1, add it to frame
    - Button1, set text to something
- Dr. Mood - listeners and sending information will be the most difficult part **
- **Don't use JNI (Java Native Interface)**
- 1 way Socketing - network address
- **Dr. Mood recommend - Text file *****
- **Must be cross - platform**
- Java program never really closes
- One of the commands will be to reset
- "Hierarchically find all the children and remove them"
- Do experimentation
- **Refresh somehow - Dr. Mood**
- Dynamically change the Java GUI

User Stories

1) User opens JGrasp and loads the Java side of the program(.jar)
    a) Iteration: 1
    b) Priority: 10
    c) Time: 8 days
    d) Tasks:
        i) Write Java program that includes ways to create each type of Java graphics object.
            (1) Time: 1 week
        ii) Write functions and try/catch statements to respond to the user if an error occurs so that there is a simple way of debugging their program.
            (1) Time: 1 day
2) User opens IDE and includes the header file in the program they are writing.
    a) Iteration: 1
    b) Priority: 20
    c) Time: 10 days
    d) Tasks:
        i) Create .h file in C++ to be included in user's program
            (1) Time: 5 days
        ii) Research the best way to go about passing information from the user's C++ program to the already open Java program.
            (1) Time: 1 day
        iii) Write functions within .h file that either write Java code to a string or somehow connect directly to the Java GUI.
            (1) Time: 2 days
        iv) Write all Java code in strings to a .txt file to be read by the Java program.
            (1) Time: 2 days
3) User compiles & runs the C++ program and JFrame reflects the changes made.
    a) Iteration: 1
    b) Priority: 30
    c) Time: 9 days
    d) Tasks:
        i) Research and learn how to dynamically change a Java GUI (refresh the GUI?).
            (1) Time: 3 days
        ii) Make the Java GUI dynamically change so that we can adjust it as the user desires.
            (1) Time: 5 days
        iii) Make the Java program check for a .txt file with which it updates the Java GUI.
            (1) Time: 1 day

iv) Update the Java GUI with the new items that the user specified in their code.
    (1) Time: 2 days

4) User opens API file to be able to see the different functions that are available.
    a) Iteration: 2
    b) Priority: 40
    c) Time: 2 days and a ½ hour
    d) Tasks:
        i) Create a document which contains all possible method calls along with explanations and examples that the user can implement in their program.
            (1) Time: 2 days
        ii) Convert this document to a PDF, making it more convenient to use and look at.
            (1) Time: ½ hour
5) User can call clear function within program in order to clear JFrame.
    a) Iteration: 2
    b) Priority: 50
    c) Time: 3 days
    d) Tasks:
        i) Create clear function that clears the GUI and sets it back to its default state.
            (1) Time: 3 days

6) User can interact with JFrame and information is passed back to C++ program.
    a) Iteration: 2
    b) Priority: 60
    c) Time: 17 days
    d) Tasks:
        i) Research how to send information BACK to the C++ program from the Java GUI.
            (1) Time: 1 week
        ii) Implement within the Java program an event system that sends signals back to C++ based on what is done by the user in the GUI.
            (1) Time: 5 days
        iii) Create functions inside C++ .h file that enable the user to use any sort of input that comes from the Java GUI.
            (1) Time: 5 days