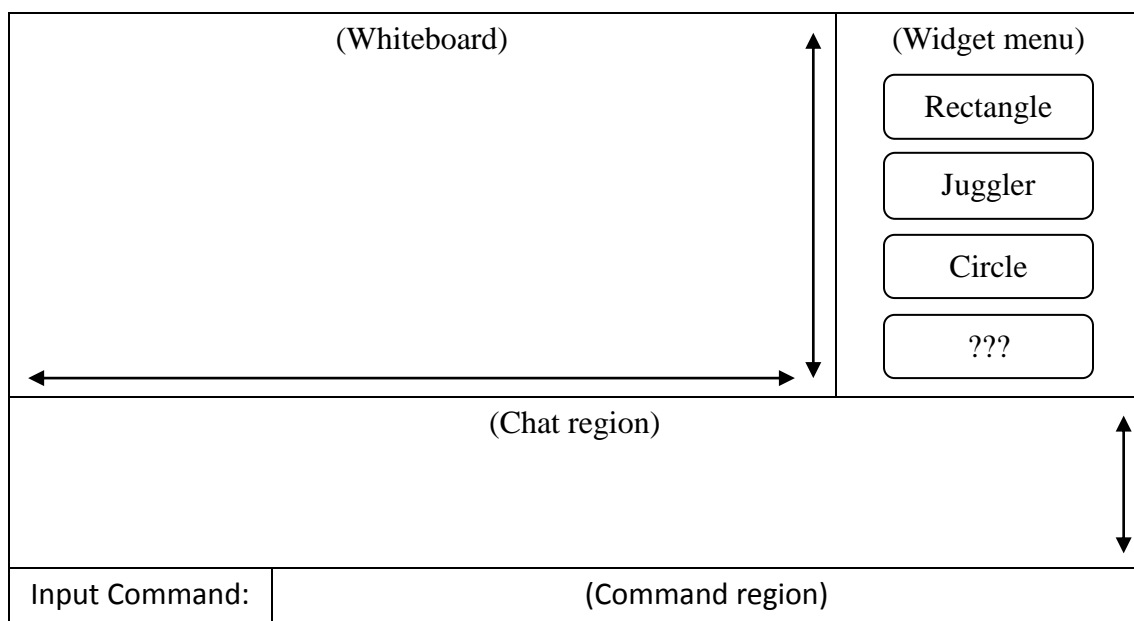


Advanced Java

Project 2

- Appearances
 1. **Whiteboard**
Place Widgets
Initial size is 500×300
Scrollable both horizontal and vertical
 2. **Widget menu**
List supported Widgets as button
 3. **Chat region**
Display chat messages from server (e.g. /msg, in project1)
Scrollable vertical
 4. **Command region**
Textfield to input command. Send command if user press <enter> on Command Region.



Specification:

- In project 2, you are asked to improve your chat system in project1 by providing GUI widgets.
- In this project, TA will provide the following four classes as widgets,
 1. `Widget.java`
 2. `RectangleWidget.java`
 3. `CircleWidget.java`
 4. `JugglerWidget.java`
- All widgets inherit an abstract class `Widget` which is provided by TA.
 1. `void parseCommand(String cmd):`
Parse a string "data1 data2 ..." into properties of this `Widget`.
 2. `String toCommand():`
Convert properties of this `Widget` into a string without <CR>, "data1 data2 ...".

3. `void destroy()` :

Destroy the widget when trying to remove it. This is used in the Juggler widget, since the widget uses a thread to handle the display.

- **Commands to/from the server:**

1. All commands to/from the server required in Project 1.

2. Create Widget Command Format:

`/post ClassName x y data1 data2 ...`

where “`x y data1 data2 ...`” is msg for `/post` in project1

Hint: use `toCommand` to generate “`data1 data2 ...`” in the sender side; and use `parseCommand()` to set properties from the string.

3. Move Object Command Format:

`/move id x y`

you change `/post` msg stored in server

- ◆ Note: clients can only move widgets which are created by themselves. Basically, we simply check the user names.

4. Remove Widgets:

`/remove id`

- ◆ Note: clients can only move widgets which are created by themselves. Basically, we simply check the user names.

- ◆ In this project, `/remove` should consider about privilege, only client who post message can remove it, including String type.

5. Commands related to widgets can be sent by user input.

In Project 2, you need to do the following.

- **Widget menu.**

1. Click a Widget and then click on whiteboard.

2. Record the clicking point (x, y) and associate this point with the Widget.

3. Post the clicking point and the widget content to the server, and then broadcast to all.

- **Whiteboard.**

1. Create Widgets by the above action on Widget Menu.

2. Move Widgets by drag and drop.

- ◆ Show the trace of moving Widgets locally. Do not show the trace to other clients.

3. Whiteboard should be dynamically enlarged if a Widget is moved out of its bottom-right boundary.

- ◆ Note: the size of the view for whiteboard should be fixed.

Other requirements:

- TA may ask you to add another new Widget during the demo.

- For wrong id or format, or invalid type, clients/servers should display error messages.

Good luck!

Due date: 5/3