Shrugs

Shrugs Help Regular Users Generate Sites

Higher-Order Testing

The two types of higher-order testing that we chose, other than usability testing, are function testing and volume testing.

Function testing:

Why we chose function testing:

We chose function testing because we spent a lot of time creating strict external specifications that the Shrugs application should adhere to. We wanted to make sure that any discrepancies between the behavior of Shrugs and our specification be caught as to ensure the programs runs as expected. We also wanted to ensure, since we have been able to view our code, that the program behaves as expected from a standpoint that is oblivious to the underlying code (i.e blackbox testing).

How we will perform function testing:

We will perform function testing by carefully going through each and every user case story in our product backlog to test all possible variations of input to see all possible variations of output. We will perform boundary value analysis on ranged input to ensure that all possible types of input result in desirable output. We will also use equivalence class testing on ranged inputs/graphical user input to make sure that we can cover all possible inputs in reasonable amounts of time. We may even spend some time creating more specific documentation to test against.

Why function testing is the best fit for our project:

Our project has a lot of graphical parts to it, so that when testing, it is very apparent if something is wrong. It is also beneficial to us to use black box testing because, if we find problems via black box testing and then fix those problems, the user is very unlikely to run into any errors that they might be capable of experiencing.

Volume testing:

Why we chose volume testing:

We chose volume testing because our application graphically displays data. It will be obviously noticeable to a user if Shrugs does not handle data correctly so we want to ensure that all amounts of data, even exceptionally large amounts, are handled correctly.

How we will perform volume testing:

We will perform volume testing by creating a large amount of elements within our editor. We will also perform modifications to some of the elements within the editor so that we can determine how many elements can be created and modified without compromising the functionality or integrity of our program. Since the editor can save/load to JSON, we can create extensively large JSON files containing huge amounts of different types of Shrugs box objects and verify that the application can handle this amount of data.

Why volume testing is the best fit for our project:

Our project was designed to allow for a lot of user customization through input. This customization includes allowing for no limit of box objects to exist within one project. Thus we use volume testing to determine if our project will be capable of handling a project with huge amounts of objects. Our GUI handlings many draw events that are critical to user experience, we want to make sure that with huge volumes of data, Shrugs can still handle the drawing and interpretation of this data.

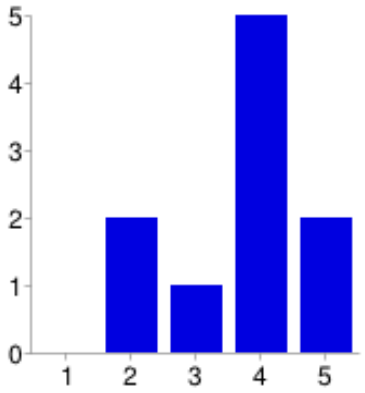
User Testing

User Survey Results

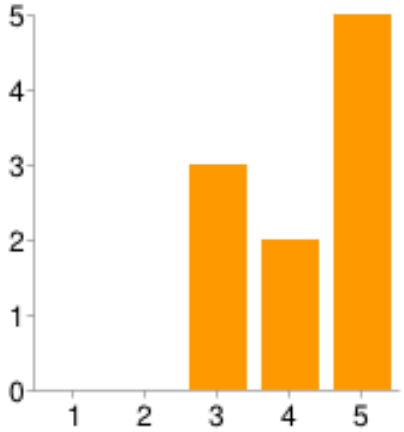
Users were directed to create and then export a website of their own design. They were instructed to include a textbox, image, div box, and link box. They were instructed to delete at least one box, and update at least one box. They were also instructed to import an existing project. We did not want to give them too much instruction on how to perform these tasks as we wanted to see if our user interface was intuitive or not.

The survey we gave them was the following google form: https://docs.google.com/forms/d/1LtZfGQYVWAvNW5wDdnF1AxBt2m8KMK9o_s79VhJIMEo/viewform

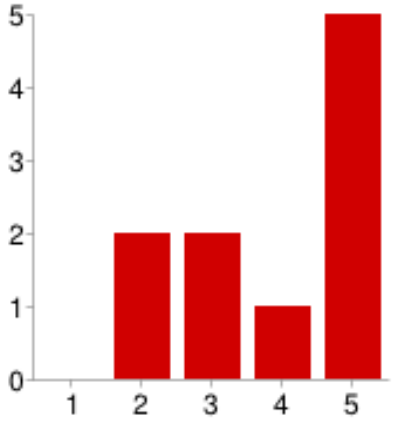
Please rate the user interface



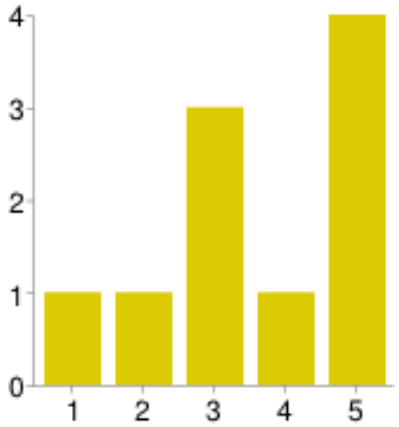
How easy was it to make new boxes?



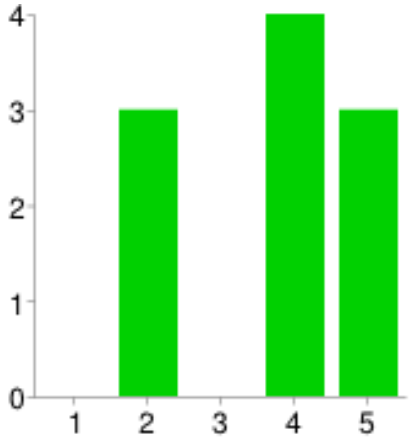
How easy was it to customize your project with boxes?



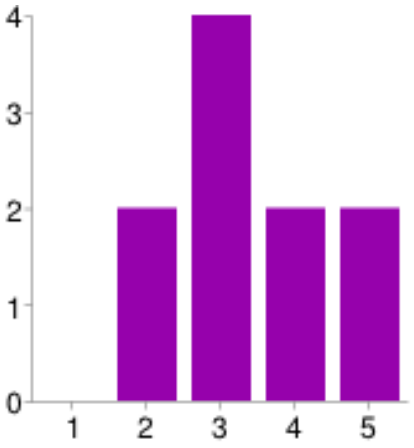
How easy was it to customize a box?



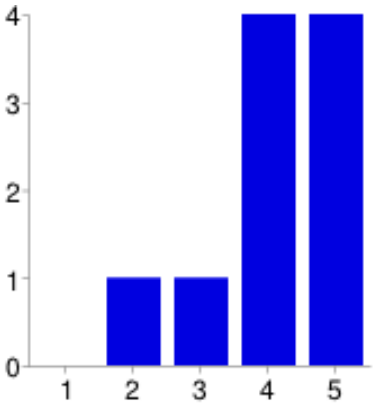
How easy was it to gain information about a box?



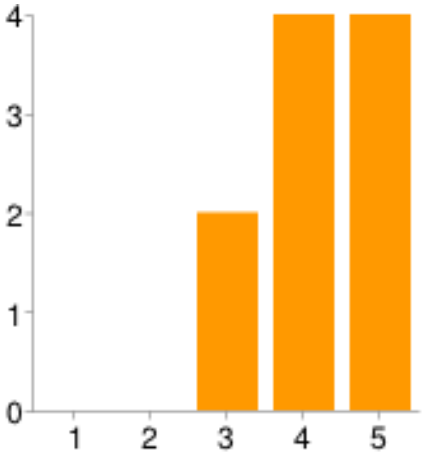
How easy was it to delete a box?



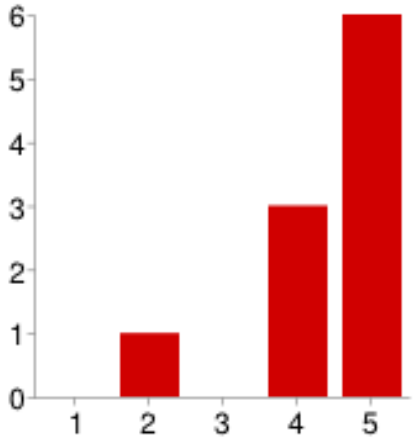
How easy was it to make a new Text box?



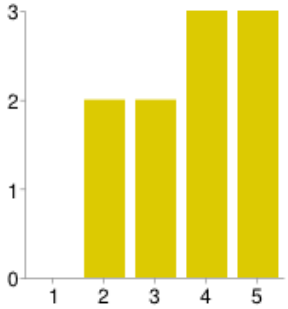
How easy was it to make a new Image box?



How easy was it to import and export your project?



How easy was it to use the program?



What did you like about the application?

- It gives the user the freedom to add any essentials a web page would need to their own page in an easy manner.
- It has few buttons and simple to use.
- It is very easy to become comfortable with the interface. I
- like the order of it
- It was user friendly
- It's decent for the amount of time that you said it took

What didn't you like about the application?

- Some sort of tutorial is needed.
- Didn't know what I was doing
- The inability to change the text in a box that you have already created.

User Survey Analysis

The Survey was actually very informative. Most of the candidates that used our application didn't have much of a programming background or even knew how a web page worked. The methods that they went about attempting to use in the application that we thought were intuitive were not intuitive to them. For example, most users did not know that in order to create a new object on the project they had to click and drag on the screen. The users simply attempted to click on the screen to make a new object or to edit currently existing objects. With this we can improve our project by:

1) adding a tutorial
2) changing the interaction so that it will be more intuitive to the user
3) changing some of the toolbars so that the user will know that it is a tool bar
4) make it so the user never has to find information about objects
5) changing it so that the objects can be customized in an easier manner

Usability Testing

Defect #	Defect	How to correct	Severity
1	Users are unable to resize boxes after they are drawn and they'd like to be able to.	Use the grid to proportionally resize children of a box being resized.	3
2	Toolbar is not readily identifiable.	Place a divider in the JMenu to set the toolbar apart.	3
3	Users didn't readily understand each object.	Add tooltips for more hints on mouseover of objects.	3
4	Help menu was not descriptive enough.	Add more detail, drawing, and diagrams to the help menu.	2

Function Testing

Defect #	Defect	How to correct	Severity
1	Shrugs does not ask user if they want to resume an existing project when they first start the application.	Add a dialogue that asks the user if they want to resume a project. If user selects yes, prompt JFileChooser.	3
2	Shrugs does not recognize when user runs Shrugs for the first time ever, and does not automatically prompt help menu.	Create persistent boolean flag 'firstTimeUse' and initialize it to true. Prompt the help menu on startup if true and persist the value to be false afterwards.	3

Volume Testing

Defect #	Defect	How to correct	Severity
1	Large numbers of same-level boxes causes extreme lag.	Separate drawing and interaction into separate threads	3