Final Report for RB\_C\_3\_Absolute\_Conquest

RB\_C\_3

Josh Brenneman, Jacob Aspinall, Harry Mitchell, Jeremiah Brusegaard

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 12/02/07 | SM | Initial Document |
| 0.2 | 12/02/17 | JB | Josh Reflection |
| 0.3 | 12/02/17 | HM | Harry Reflection |
| 0.4 | 12/02/17 | JA | Jacob Reflection |
| 0.5 | 12/02/17 | JB | Jeremiah Reflection |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Team Member 1: <<Harry Mitchell>> 4](#_Toc164665234)

[1.1 What went Wrong 4](#_Toc164665235)

[1.2 What went RiGHT 4](#_Toc164665236)

[1.3 Lessons learnt 4](#_Toc164665237)

[2 Team Member 2: <<Josh Brenneman>> 5](#_Toc164665238)

[2.1 What went Wrong 5](#_Toc164665239)

[2.2 What went RiGHT 5](#_Toc164665240)

[2.3 Lessons learnt 5](#_Toc164665241)

[3 Team Member 3: <<Jacob Aspinall>> 6](#_Toc164665242)

[3.1 What went Wrong 6](#_Toc164665243)

[3.2 What went RiGHT 6](#_Toc164665244)

[3.3 Lessons learnt 6](#_Toc164665245)

[4 Team Member 4: <<Jeremiah Brusegaard>> 7](#_Toc164665246)

[4.1 What went Wrong 7](#_Toc164665247)

[4.2 What went RiGHT 7](#_Toc164665248)

[4.3 Lessons learnt 7](#_Toc164665249)

Time for reflection! The purpose of this document is to capture student experiences – both positive and negative – during the semester long team project. What are the key events that are to remembered? What are the key lessons that you learnt that you can carry over to future projects. Think back starting from the time the teams were assigned, then project proposals were made, screenshots developed, requirements, architecture, framework demo, test plan, and finally the demo. What would you do different if you were assigned the same team and the same project again?

Each team member is to record his/her thoughts on a separate page. Thanks!

FOCUS ON LESSONS RELATED TO

1. Working in a TEAM
2. Software Development processes
3. Technical Issues

# Team Member 1: <<Harry Mitchell>>

## What went Wrong

It is hard to not make Java Swing look ugly. We spent more time trying to make our UI look nice than we should have. We also eventually had to override how UI objects were made to make our project more modular.

My variable naming conventions were very terrible at first. When I had to look back on code later, I had no idea what certain variables were storing.

We had to walk back from a meeting with our TA in heavy rain and Jeremiah's laptop didn’t work correctly for a while. We also had a power outage the morning before a meeting so we couldn’t charge our laptops or work on the project at all.

## What went RiGHT

We had very little problem with Git. We always worked on the project together in group code jams. This helped us talk to each other about problem we each were having and allow us to write code that interacted with each other’s code more efficiently. We were always able to meet at least one day each week and most weeks we meet multiple times. We had very good communication overall. Whenever I needed some new function from an object someone else was creating, I could talk to them about the functionality I needed and they would be able to add it for me. I was also able to do the same for all of my other team members. We created a game that will look great on all of a resumes and was fun to create.

## Lessons learnt

I learned how to use JavaSwing throughout this project. I learned how to use Git. I learned the basics of using sockets to pass information between two different computers. Using agile development allows developers to work very efficiently. Working with a team of people will make projects much easier.

# Team Member 2: <<Josh Brenneman>>

## What went Wrong

For the most part, all things when rather well.  The few hiccups that I personally encountered was setting up the git repository on my laptop.  The way that I did it I had to enter my username and password everytime I wanted to push or pull, that got pretty old pretty fast.  Ensuing certain functions of the game worked properly.  One such example is making the score update properly, it was fine at first, but when we added in the eliminating players element, it complicated things just a tad.  Another thing that took a while to figure out was how to combine JPanels, as the ingame screen is made up of two different panels, so getting them to work together was more of a hassle than I was expecting.

## What went RiGHT

I would say that a lot of things went pretty well.  For the most part, there were no big issues with creating the UI for both the game and the external menus.  Also ensuring that the different menus that i worked on worked properly also went rather smoothly.  For the most part i think we as a team got along really well, it helped that we all knew each other beforehand, so it wasn’t like that we were working on this project with complete strangers, which i think help with keeping a relatively open communication.  All team members all were very good at staying on top of the ball, we were never behind schedule for more than a day, every time that we would get together to work, we would be working the entire time, which helped to ensure that we would be able to finish the project by the deadline

## Lessons learnt

Some of the big takeaways from this class, for me, include learning java swing.  I had never really work with UI aspects in java, so that was a fun thing to learn.  Another big takeaway was working independently.  Aside from the demos, there was nothing else to ensure that you were on track, so it was a good experience to self motivate as a team to complete the project. Another thing that i learned was how to use git.  That was an experience to say the least.

# Team Member 3: <<Jacob Aspinall>>

## What went Wrong

The biggest thing that went wrong was when we designed the networking code at   the beginning. Originally, we communicated between the server and clients with strings using Java’s PrintWriter and Scanner classes. As soon as we started having to transport complex classes between the client and server, we realized this wasn’t sustainable long term. Eventually we found out about Java’s built in Serialization interface, and used that for sending and receiving information

## What went RiGHT

My group’s teamwork went very well. We decided to schedule meetings on every Saturday and Sunday afternoon for 4 or 5 hours. This made it very easy to be prepared for demos. Since we all met together, none of us were able to procrastinate and stress each other out.

Also, we all got along as a team very well. There were no members with grudges or hatred towards other members.

## Lessons learnt

The biggest lesson I learned is that the best way to develop code is in a room with the rest of your team. The reason being is that you can easily ask and discuss ideas about the project. Also, whenever we had code issues or misunderstood concepts, the other team members are able to help you right away. Working next to each other means a problem that could have lasted for hours, can be solved in only a few minutes.

# Team Member 4: <<Jeremiah Brusegaard>>

## What went Wrong

Setting up the database was challenging because I didn’t really have any experience with SQL. So, deciding what tables should be created and what should be in them was difficult. So, in the beginning there was a lot of revision work on each of the tables and the content in them. This took a long time to fix and refactor tables. For example, we originally stored ID’s for cards and then referenced the IDs in the table where the players had their card collections. Which it was hard to figure out what cards had from MySQLWorkbench. I then changed it to stored it by name since none of the cards shared the same name.

## What went RiGHT

For the most part everything went well. I think that having group code jam sessions for all the meetings we had helped a lot. I think that we communicated availability well and we were still able to keep working on the project even if someone couldn’t make it to the meetings. Personally I am very happy that Jacob was able to figure out all the Server code because I know that was a very complex part of our project and for the most part worked pretty smoothly. Another thing that worked well was our demos and being prepared for them. We would always test the code very harshly to make sure that our project would be functional for all the demos.

## Lessons learnt

I think we all learned to do more research about design choices and organizational patterns. Many times we had very cluttered code that could have been avoided if we had more time to plan out what we were doing. I also learned a lot more about SQL and how to use it properly rather than just using it to get things to work. I also learned about JDBC and how to use it in Java. Git was another topic I wasn’t as familiar with so learning how to use it and use it well was a good skill to learn from this class. Another thing I never thought we would run into was memory management in Java. So I learned that loading images is very expensive in terms of resources and the images should be pre-loaded as a last resort.