

Project 19 Report

Elba Chimilio, Curtis Hiscock, Edan Canning, Rebecca Loreda, Stephen Slogar, Katie Harrington, Logan Peck

Dr. Jonathan C.L. Liu

COP 3503

December 1, 2016



Casino Royale: A Modern Take

Our team's initial idea when designing our project was to implement a fun gaming interface through which players could simulate two card games, both of which require and allow for betting amounts and award a payout to the user. Overall, the project's development was a success, but as a team we learned a few lessons in the process. The most important lesson we learned regarding software development is underestimating the time required to build each facet of the interface and of both games since we had consider various possible test cases. In addition, continuing off of this trend, we learned the importance of not blindly writing code. Testing functions regularly and as they are created makes the debugging process so much easier.

“Everyone should learn how to program a computer, because it teaches you how to think.”

- Steve Jobs



We began development by creating our Unified Modeling Language diagram. We attempted to create a schedule for when each portion of the program would be completed, but we had not anticipated the time it would take to develop each function and properly test them. Throughout the development process, we came up with additional test cases and parameters that required us to change the overall structure of our program slightly. Through this, we learned that in the software development process, it is almost impossible to account for every single test case.

In order to make a more visually appealing game interface, we learned about and made consistent use of ASCII art for all cards in each deck. In addition, ASCII art played a role in livening up the overall interface user experience.

The project required consistent team communication. Every team member needed to be aware of how each part of the program works. This helps with the general learning process and contributes to a more seamless development process. It also allows all members to help throughout the debugging process by having an overall understanding of the program’s functionality, including desired inputs and outputs. Many of the classes and program sections were created and implemented using multiple team members as well as interfacing the rest of the team when it came time for the convergence of both games into the same user experience.

There are ways in which, if given more time, we would like to do to improve the overall program. For example, adding more games would add to the overall functionality, the overall entertainment value, and allow for additional ways to make money from this software through in-application purchases. Additionally, creating a Graphic User Interface could make for a more visually appealing user experience and would engender more downloads if this were to become a mobile app or greater use if this was created as a web application. Finally, implementing a way for multiple players to interact with the interface versus just having the opponent be the computer, would add a new level of depth to our overall program in terms of entertainment and software development.

References

Links for Images:

(http://comesingapore.com/pic/load_op/2097)

(https://lh4.ggpht.com/-fd1T1ZGYi01Aohv_f96h_wXr9iJvz8u9A9CtrlTVE-wjNy5wlb8OPO6618ii_qcTU0=w300)

(<http://kwbridge.com/art/hand35-b.gif>)

([http://pad1.whstatic.com/images/thumb/a/aa/Play-War-\(Card-Game\)-Step-6.jpg/aid675276-728px-Play-War-\(Card-Game\)-Step-6.jpg](http://pad1.whstatic.com/images/thumb/a/aa/Play-War-(Card-Game)-Step-6.jpg/aid675276-728px-Play-War-(Card-Game)-Step-6.jpg))