John Grillo

Kurtis Pittman

Gabriel Wahrmund

**BLACKJACK DEVELOPER NOTES**

Our chosen project was for our “board game” was to do the classic card game of Blackjack. The rules are well known and straightforward. Each face card is worth 10 pts., an ace is worth either 1 pt. or 11 pts. And otherwise, each card is worth its number value. If the player meets 21 pts., they win and if they exceed, the player automatically loses. And if the dealer has a “soft 17,” then the dealer must draw another card (“hit”) regardless of their own desire. Otherwise, if the dealer has a higher point total than the player, the dealer wins.

Pretty straightforward. But simple things for simple fun lead to many, many lines of code and long hours.

The hardest part of this project, according to our lead designer Kurtis, was writing the code for shuffling the deck. The deck is first sorted by thirds, and then shuffled as a whole. C++ was challenging to all of us, but we were able to plod our way through it. Otherwise, pulling the “top card” rather than a card from random in the pack was a challenge.

Kurtis handled the shuffling and dealing system, Gabe handled the banking system—which is a completely separate element from just the game mechanics—and I handled the menu options. I choose a do-while loop because each set of options would repeat themselves until a particular set was chosen or the quit option was set to “false,” as the do-while menu loop’s initial state is set to true. My challenge was to keep the loop from executing forever, or if an incorrect key would plunge it into a fail state. Fortunately, the intrinsic nature of the loop would repeat options covered that.

There were a couple errors we found upon initial compilation: At first, two aces amounted to 22 and wouldn’t “go soft” and return a value of two. Second, the initial if-win statement made so that if the player had more points than the dealer for any reason, even if they should have busted, the player would win [e.g. because 25 > 21, the player wins, when it should be a zero or null statement.] These were quickly corrected.

Otherwise, the project went swimmingly well. Everyone pulled their weight and their fair share of the work, we met on a regular basis to brainstorm and troubleshoot. We’re confident in our program’s success and have faith in its coded graces. The only thing the program really lacks is graphics, but otherwise, functions exactly as it should.