



Deuces Wild Poker

Casino Game Maker, Inc. - Game Development Project

SUU Software Engineering - Fall 2017

Overview

Casino Game Maker is participating with SUU's CSIS department by offering a project for current students to learn Software Engineering principles. The requirements for a Deuces Wild Poker Game development project is being provided by CGM.

Goals

1. Create math files for the poker game.
2. Create a configuration file.
3. Create a graphical asset package.
4. Create a sound asset package.
5. Create a functional poker game for a PC.

Specifications

Using the tools provided by CGM, the students will create a playable poker game that will run on a PC. The game will also run on CGM's target platform and will be used as a foundation for future poker games to be put on a casino floor. The poker game will be standalone.

The poker game will be a deuces wild game (2's are wild cards). The player will be dealt 5 cards. The player will then choose to hold 0-5 of the cards. Once the cards are held, the player will be dealt new cards for the unheld positions. The wins will then be processed.

The game will need the following:

- 6 buttons, 5 to hold the cards and 1 to deal/draw the cards.
- A wincheck to calculate card wins.
- An instruction message for the player at the beginning, hold stage, and end.
- Correct visual representation of 5 cards.
- Visual representation of the win.
- A message at the end of the game clearly stating how much has been won.

Tools Needed

Adobe Animate CC

TexturePacker

Notepad++

CGM Libraries and Example Template

Milestones

I. Understand Poker Game rules.

Gain an understanding of how the game is played and what screens, transitions, and images will be needed. This should include an understanding of how to read in values from the config file.

II. Setup and learning the tools.

Gain an understanding of Adobe Animate CC and the ActionScript programming language. Learn how to use TexturePacker and how it is utilized with the CGM engine. Get familiar with the CGM libraries and engine.

III. Create game engine.

Develop the program that will run the underlying game function and logic. This could include a state machine or event driven approach.

IV. Implement game math.

Create XML math files that are loaded by the game engine.

V. Incorporate graphics and layout.

Develop XML files that work with the engine and Texture Packer.

VI. Incorporate sounds

Develop XML files that work with the engine and map the correct sounds.

VII. Test.

Test all functions and features of the game.

VIII. Document

Document the game functionality and features.

Contact Us

Please contact us through email:

Amy Gifford agifford@casinogamemaker.com

Bryan Excell bexcell@casinogamemaker.com

Jared Ross jross@casinogamemaker.com