Quasar Studios
quasar.gamestudios@gmail.com

Video Poker

Starter Kit for Unity Projects

OVERVIEW

The Video Poker Starter Kit package is intended for those who want to create poker games using the Unity engine. The project is complete and has a lot of options, the source code is fully commented and support is provided via email by using the address specified at the top of this document. Having more than 10 years of experience in the gaming industry, you can rest assured you can ask anything.

SPECIFICATIONS

The project contains a lobby and three video poker templates. The templates are: Classic, Style and Arcade, each one having its own flavour and feeling. Each template is a complete game with unique graphics and can be played like a separate game. If this project will be met with interest, multiple templates can be added in the future. Also the project contains a fully functional deck of cards that was specifically drawn for this starter kit. The graphics are done for a FullHD target resolution. It has a height of 1080p while on horizontal it extends beyond 1920p in order to look good on wider devices (like the 1024x600 tablets).

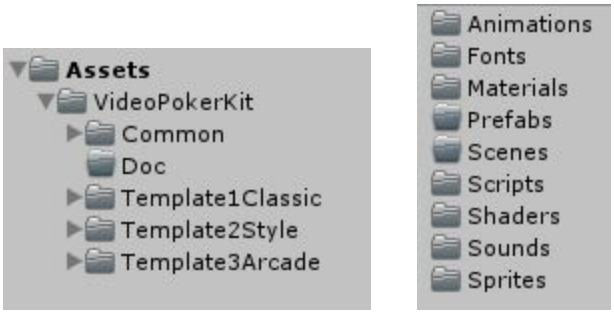
The code and assets are fully cross-platform compatible and the project was intended for mobile use. It should work on all platforms, although is was not tested on all of them. The graphics can easily be scaled down if needed, mainly the backgrounds and other larger elements. For the atlases we used the built-in Unity Sprite Packer.

PROJECT STRUCTURE

The project contains a common part that is more like a framework which contains the poker logic and some generic classes for buttons and other elements. It is important to mention at this point that most of the code for the project resides in the common part. This is a proof that it

is written very generic and it is not linked in any particular way to any of the templates. You can construct a new poker game that behaves differently than the templates with minimal changes in the code.

Along the common part each template has its own folder with specific assets, mainly graphics, some prefabs and very few extra scripts.



Inside each part (common or a specific template) a standard directory set was used with separate folders for animations, fonts, materials, prefabs, scenes, scripts, shaders, sounds, sprites. Each template uses heavily the common part and only if it needs special behaviours it has resources in its own folder.

THE CODE

The code used for this project is C# and all the source files are fully commented so that the mechanics are easy to understand. The complexity level is medium.

GAME SETUP

Each template has a similar setup:

- A cards zone that can hold 5 cards
- A paytable panel
- Deal button
- Change bet buttons
- Credit info with Add Credit button
- Info panel
- Exit button
- Win messages

GAMEPLAY

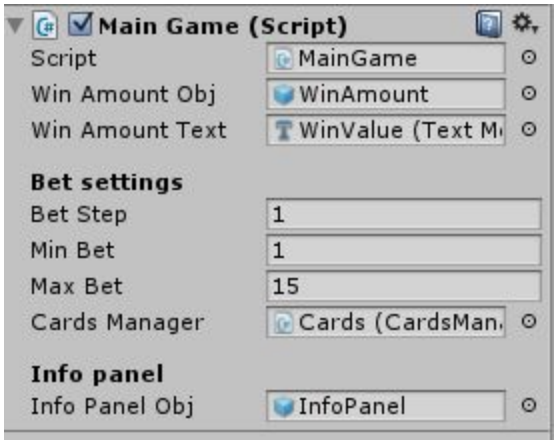
As you know, the classic Video Poker has a gameplay based on 2 stages: in the first stage you get 5 cards from which you can select any number of cards that you want to hold and carry over to the second stage. After holding, you press Deal again and you get new cards besides the ones you holded in the first stage. At the end the final hand is checked against the paytable for a win. In case of a win, the prize is awarded to the player while an overlay win message is displayed.

SCENE STRUCTURE

In each scene file from each template you will find a similar structure (they are VIdeo Poker games after all).

Main Game

The **MainGame.cs** script is the one that controls the main game flow. It has several settings that can be set in the Inspector:



We can change the min and max values for the bet and also the bet step that is used when pressing the Bet+ and Bet- buttons. For example, in the Arcade template the max bet is set to 5 because a lot of the classic machines for video poker had this behaviour. Also the paytable is different in the Arcade template and each bet has a different column that is selected at each bet change.

The cards manager

The cards manager has a deck array with all the cards used in the game, an array with the 5 slot cards of the player and also some settings related to the deal animation timings. The first value is the delay animation between 2 consecutive cards in the first stage of the game, while the second value is used in the second game stage. We have settings for each stage because usually you want your players to know which stage is which. When playing a lot ot sessions you want to quickly know if you can hold (you are after the first deal) or if the game is over (you are after the second deal). By using different values you give the player a hint about the current stage of the game.

Buttons

All of the buttons used are based on a common class named **GameButton**. The class represents buttons with 2 states (normal and pressed) and also permits a highlight state if needed (the Deal button usually uses this state). The class has a virtual method named **PressAction()** that is called when the button was pressed and all new button classes must implement it.

The rest of the elements found in each scene are: win messages, buttons, labels and sound. Each of them is pretty self explanatory and they can be inspected in the related code.

Happy coding