

Village Ventures is a lively dance game set in a vibrant Wild West saloon. Players take on the role of a charismatic dancer aiming to dazzle the saloon crowd and earn commissions to upgrade their outfits and enhance their performances.

In this game, players start each session by entering the local saloon. They must navigate to the dance floor to start performing. The more they move and perform, the more they earn. For now only basic performance (walking) is set up but can easily be upgraded (like adding bonus for jumps, etc.) but was not for time sake.

Earnings from performances can be used to purchase new outfits from the saloon's tailor. These new outfits not only change the dancer's appearance but also increase the commission earned from each performance. The more stylish and up-to-date the outfit, the higher the commission rate.

Each performance is time-limited, encouraging players to maximize their moves and showmanship within the allotted time.

In the development front I started by creating a basic scenario for testing and used a character movement script from the internet with tweaks, to simplify and make it fit better in the game.

From there I started to work on the 'GameSingleton' class, that keeps track of the overall state of the game, and the 'GameInterface' singleton class, that keeps track of information relevant to the player and shows it on screen when necessary.

With the basic systems in place I made a ScriptableObject to hold the Outfit (item) data and another that holds a list of all of them to facilitate consulting information about them throughout the game systems.

A class called 'ItemStack' individualizes a single item in the Inventory class, that then can be used (worn) in the game. The ItemStack class holds a reference to the item data as well as keeps track of its quality (the more you use the outfit the more it loses value).

Another role of this class is to be used by the custom 'Animator'. Which is responsible to refresh the sprites and equip (visually) items in the player character. The idea was to facilitate the scaling of the game by automatically getting the correct sprites using a default sprite layout and custom animator that takes advantage of it.