Barrel Joyride

Overview

Barrel Joyride is a trimmed down version of Jetpack Joyride by Half Brick, it is meant as a template and as such will still need to be fleshed out a bit for a proper release. That being said it is still pretty playable at the moment.

It was designed around the dimensions of the iPad, as that is the device I own, however naturally being made in Corona it will work on plenty of other devices.

A general overview of its features are:

★Infinite scrolling background
★Collectible items
★Obstacles (moving, rotation and stationary)
★Jetpack mechanics
★Very basic particles for the jetpack
★Data saving
★Sound
★Looping music playlist
★OOP structure
★A wizard

This documentation is not meant to be a full step-by-step tutorial but rather an explanation of all the various classes that are included. All the code is documented as well and hopefully is pretty easy to understand or at the very least use.

Classes

class_Player.lua

The player class doesn't really do all that much, it is essentially just a visual image that has a physical body attached. It has the main collision handler function and a Jetpack object attached to it which will be explained later. Things that could be done to this Class to expand it would be to allow for different visuals and also to separate the jetpack image from the player image and place it in the Jetpack class.

class_Jetpack.lua

The jetpack class is another simple class that just acts as the thrust part. It simply uses "applyForce" to move the player and also plays a sound and creates the particle objects.

Ways to expand this would be to move the actual jetpack image to it and allow for variations.

class_Sparkle.lua

The sparkle class is the particle that is chucked out from the jetpack, it has a plain white visual which is then coloured in code randomly to make it look all magical. It just uses a transition to chuck them out and a bit of randomness to make them more fluid.

class_Coin.lua

Yet another simple class here consisting of a visual, a physics body set to "isSensor" so that the player doesn't get physically affected. The only function that really does any resemblance of work is "playerHit" which is called when the player object collides with a coin. All it does is play a sound, fire off an event and then destroy itself.

class_Zapper.lua

I see a pattern here, this function is basically the same as the Coin class so I don't really need to explain much.

class_Missile.lua

The missile does a little more work than the rest of the classes as it actually has to move, but it is still pretty simple. It still has the regular image and physics stuff but it also has an "alert" image that pops up on the right hand of the screen just before the missile is visible and then removes itself after a short while.

class_Level.lua

This class does a little more work then most others, but don't worry it's still pretty simple. It loads up some json data which describe each of the level sections (explained later), creates a roof and floor so the player can't escape and sorts out the scrolling.

It has an update function that moves the two active level sections and then another one that creates a new section and destroys a previous one to allow for infinite scrolling.

class_LevelSection.lua

The levelSection class is created through json data and sets the position of all the zappers, missiles and coins.

class_Utils.lua

Just a class with a couple of helper functions for reading in files.

Overview

I hope you enjoy this template, you are free to do whatever you want with it. You can modify the art, code anything just have fun and if possible give a shout out to Glitch Games if you do use it - @GlitchGames