Caravan Developers Guide

Caravan card game

Alex Studniarz
CSUID: 2824959

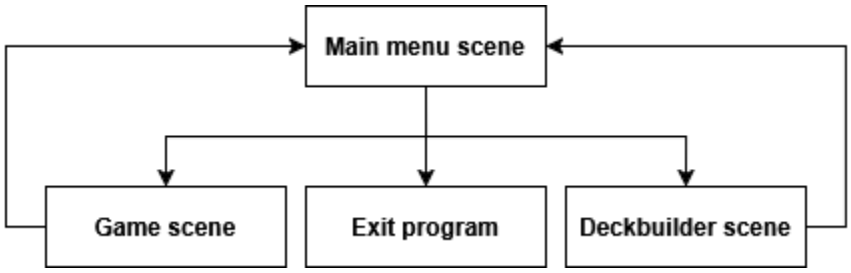
Bilal Haider
CSUID: 2726111

Gabriel Surovey
CSUID: 2798110

Preface

Caravan is built with Godot, an open source free game engine. All the code for this project is done in GDscript, a programming language made specifically for godot that is syntactically very similar to python. This guide assumes you are familiar with Godot.

Highest level architectural view



Main menu scene (main.tscn)

main_menu.gd

This is the only script native to this scene and its function is to change the current scene to other scenes and terminate execution of the program based on received signals from the button nodes. This scene also calls a global script called save_load_manager.gd to check if the player has a save file, when a save file is not present it disables the “Play” button.

Deckbuilder scene (deck_builder.tscn)

deck_builder.gd

This script is attached to the root Deck_builder node. It serves several functions. The first is to attempt to load a save file by calling the global SaveLoadManager, copying the saved deck to a local deck for editing. Next is to instantiate 53 card_button.tscn objects for each grid container, setting the parameters for each card with values defined by the card_database.gd script and connecting signals between the cards and this script. It also pairs the scroll function to each scroll container so that they move in unison. There is also a method to receive signals from cards called “_on_update_deck”, that updates

the deck and CardCounterLabel and emits a signal to the cards to update their visuals. Deck_builder.gd also controls the MainMenuButton node, disabling it if there are less than 30 cards in the deck and enabling it if 30 or more cards are in the deck.

main_menu_button.gd

This script is attached to the MainMenuButton node. This script saves the local deck in deck_builder.gd to a savefile by calling the SaveLoadManager global. After the deck is saved it changes the scene back to the main menu scene.

card_database.gd

This script is not attached to any node in this scene and is called in the game scene as well. It is a massive dictionary that defines all the parameters that a card of a specific type should have.

card_button.tscn

This scene is instantiated by the deck_builder.gd script. It is a TextureButton node that sends a signal to update the deck in deck_builder.gd when left clicked.

Texture_button.gd

This script is attached to the TextureButton node. The purpose of this script is to update the visuals of the card when it receives the updateCard() signal and to emit the updateDeck() signal when clicked. Parameters to identify what kind of card the object will be are defined at the top of the script.

Game scene (main.tscn)

The game scene is the most complicated scene in this project. It contains scripts to create card objects, create Caravan objects, track game rules and much more.

main_game.gd

This is the second largest script in the project and is responsible for many different things. It is in desperate need of a rework.

- Player and CPU assigned caravan.tscn objects are created here
- Turns are managed here along with the visuals of the turn_display_panel

- The setup round is handled here
- The CPU deck is shuffled here
- This script tells the decks to draw cards to the hands
- This script sends card objects to the Caravans
- This script handles jokers (this is needed because jokers effect all Caravans on the board)
- Caravan selection and removal is handled here
- The win loss conditions are handled here
- The game ending panel is handled here
- The CPU's decision making is handled here
- The forfeit button panel is handled here

hand.gd

This script manages card objects that are parented to it. It receives these card objects from the deck.gd script and has code to visually order the cards within its boundary. A function inside this script, "on_card_clicked()", receives a signal from the card.tscn objects currently in the hand when they are clicked. This function highlights the card object clicked, sets the selected card variable, disables the discard_tract_button, and enables the discard_card_button. The function _discard() is called from main_game.gd. When called it deletes the selected card object. The function play_card(), works in a similar way, it is called by the function add_card_to_caravan() in main_game.gd, except instead of deleting the card it removes the selected card as a child and returns the card to the calling function.

deck.gd

This script contains functions that load and shuffle the deck from the savefile by calling the SaveFileManager global, update the deck card counting label, instantiate card.tscn objects, and set their appropriate parameters by calling the card_database.gd script, and finally send card objects to the hand when requested to do so by the main_game.gd script.

opponent_hand.gd

Contains most of the same functions as hand.gd, with the highlighting and signal catching function removed. Additional functions are included to select the lowest value card in the hand, discard a random card from the hand, select the card closest to a given value, and select a card with the same suit as the card given.

opponent_deck.gd

The code for this script is nearly the exact same as deck.gd, with one exception. This script does not load a deck from the save file and instead has a pre-defined deck that it draws from.

Card.tscn

This is the card object, it is made of sprites and a collision box to detect mouse inputs. It is instantiated by the deck.gd, opponent_deck.gd and caravan.gd script (to create temporary duplicate card for projection).

card.gd

This script is responsible for keeping the parameters of the card, toggling the highlight around the card, and sending signals to other scripts when clicked based on its current state.

Caravan.tscn

caravan.gd

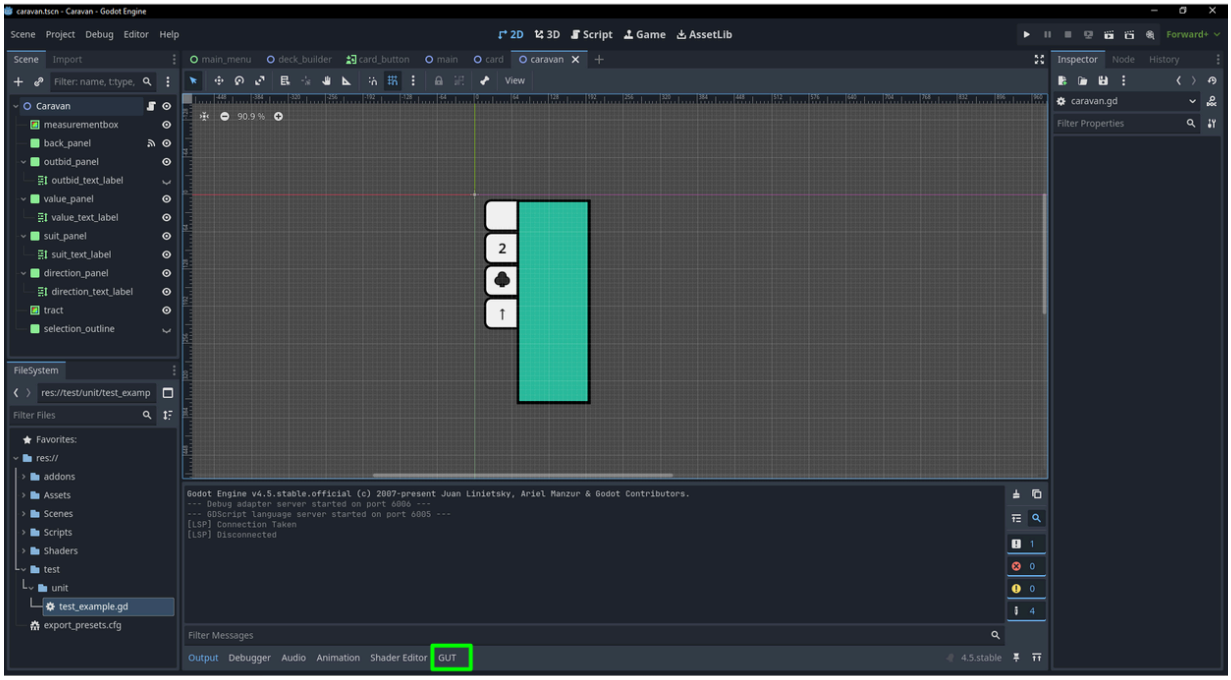
This is the most massive script in the game, and like the main_game.gd, it desperately needs to be reworked. The caravan.gd script is responsible for the following functions:

- Resetting the parameters of the caravan and deleting all children from its tract node
- Specific functions for removing cards from its tract node
    - Remove cards of a specified value
    - Remove cards of a specified suit
    - Remove card at a specified index
- Displaying, setting and keeping track of all of the parameters of a Caravan
    - Outbidding
        - Outbidding is actually NOT set here, instead it is being set inside the win_loss_conditions() function in the main_game.gd script
    - Value
    - Suit
    - Direction
- Adding cards to the tract node, and applying face any face card effects
- Code for visually organizing and displaying the cards in the tract node

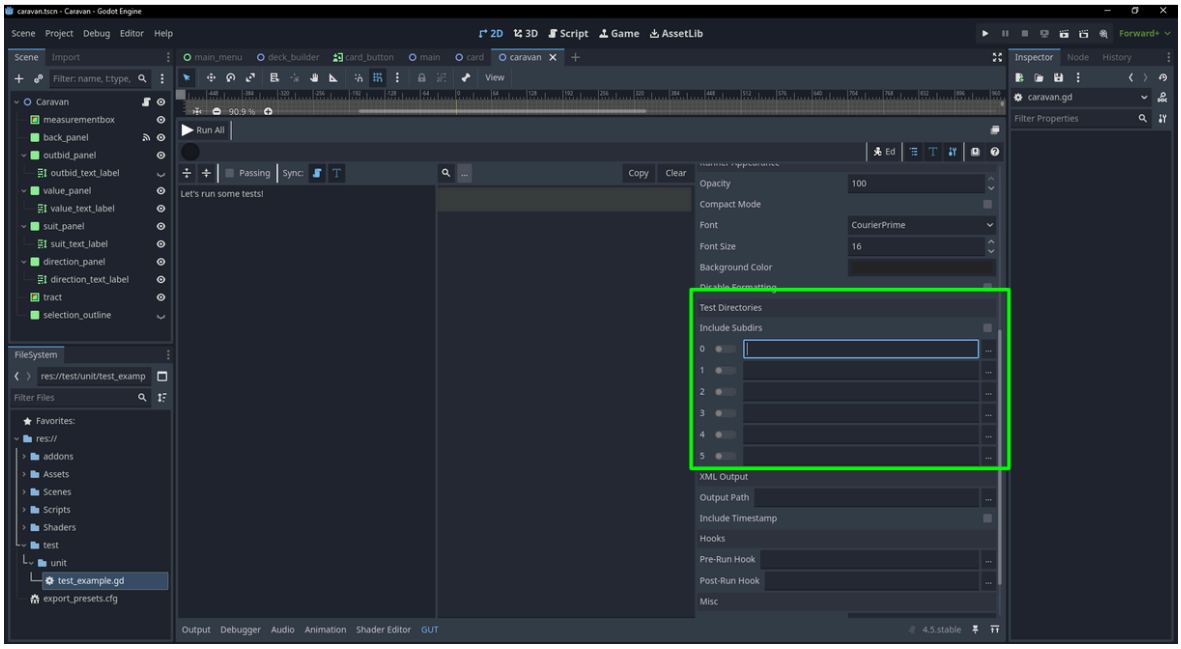
    - This code is very similar to to the code in hand.gd and opponent_hand.gd
- Checking the validity of a players selected card placement and visually projecting this placement
    - Copying the selected card in the players hand (this is for projection)
- Handling the collision of the mouse and card objects that are children to the tract node

Creating and running test cases

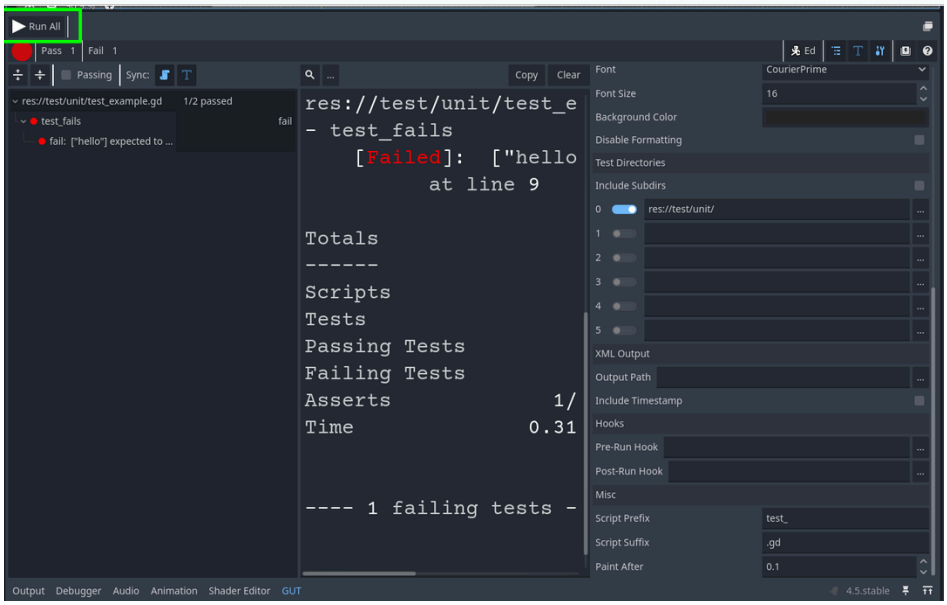
This project comes with the Godot Unit Test (GUT) extension installed. To set up GUT for this project, click on GUT on the bottom center panel.



Next scroll down the right side of the GUT window until you get to the section labeled “Test Directories”.



This is the area where you will paste the path that contains any test cases you have or will create. A test case example is currently in the path “res://test/unit/”. Once the path is pasted, you can run test cases by pressing the “Run All” button in the top left hand corner.



The GUT documentation can be found here: https://gut.readthedocs.io/en/v9.5.0/