MiniDayZ RE (Reverse Engineering).

Mod Creation

Introdução

The entire study on the internal functioning was done based on version 1.2 of MDZ+, any previous versions may contain different internal content, so it may not work when exporting mods made based on this edition.

This document will be the starting point before we delve deeper into the things that interest us.

All or most games are developed by a "Game Engine".

This game engine is responsible for: Managing Resources such as Assets (Audio, Sprites, Textures, Scripts, etc.), Rendering System, Managing physics and implementing several other things that will make the programmer's life easier.

This way, the programmer does not need to write functions from scratch, as the engines offer resources in a safe and efficient way.

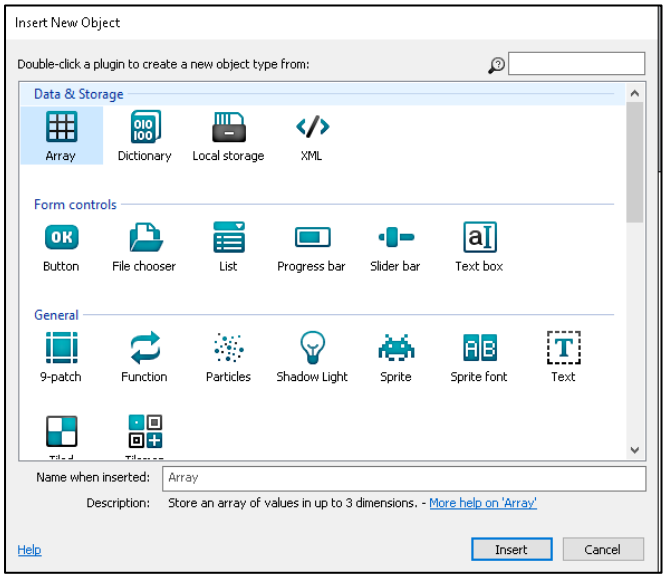
There are several game engines on the internet such as Unity, Unreal Engine, Game Maker 2 and Godot. Both are free, but may require some type of paid license to add extra features.

MiniDayZ was also created on a game engine, known as c2 or rather, **Construct 2**.

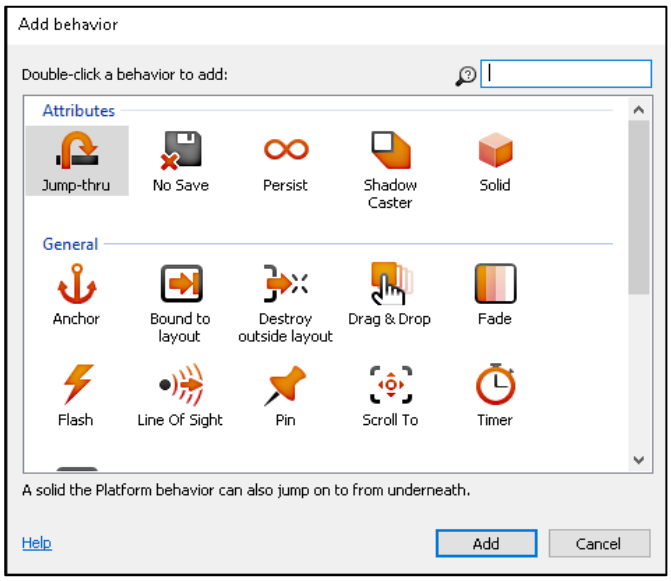
Construct 2 is a very easy to use engine, suitable for those who do not have much or no experience with programming languages, as it uses **event sheets** internally for game logic.

Construct 2 logo. [Image 0.1]

It has implementations of **Plugins** and **Behaviors**, objects that will give actions and functions to items in the game.



Some plugins found in the engine [Image 0.2]



Some object behaviors [Image 0.3]

Construct 2, as an engine, also manages resources such as collisions, rendering, audio, physics and peripheral interactions.

-But how does she do it?

Simple! c2 manipulates and uses an arsenal of “WEB APIs”.

-Web APIs?

Web APIs *(Application Programming Interfaces)* are sets of rules and definitions that allow different software to communicate with each other.

Web APIs allow different systems and services to communicate with each other in a standardized way. They define how requests and responses should be structured, facilitating interaction between applications.

APIs allow developers to leverage existing functionality without having to fully understand the inner workings of the underlying system.Tudo isso em um Web Browser!

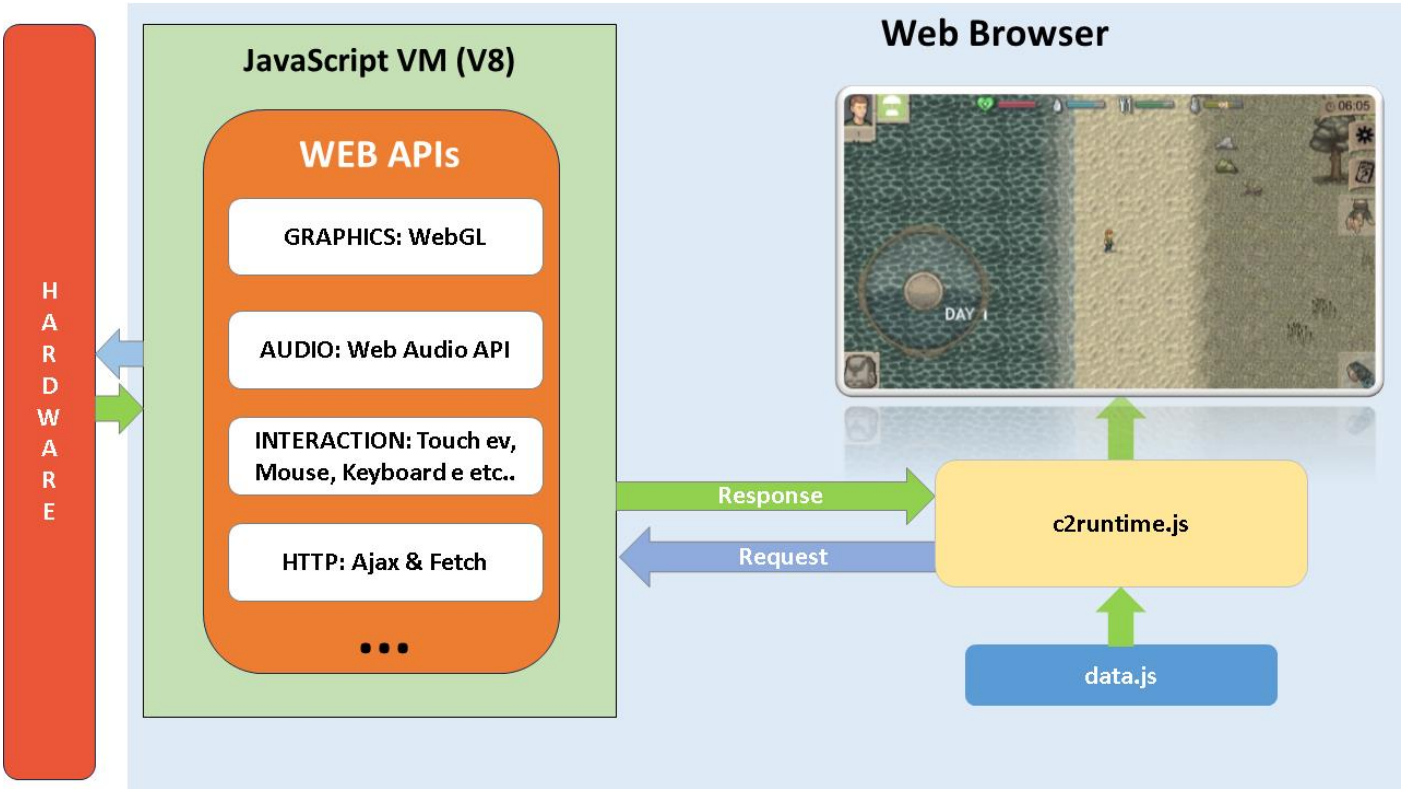
Construct 2 is 100% based on web technologies (**HTML5**), which is why MiniDayZ can be played in a browser just like in https://nextdev65.github.io/MiniDayZ/ exported by NextDev65. There are certain advantages and disadvantages with this type of implementation.

Some Advantages:

- **Multiplataform**. With HTML5 export, people will be able to play on different platforms such as Computers (on the WEB or systems that simulate this) or mobile devices through one implementation Web *View*.
- **No Download:** Web games do not require a client-side installation.
- **Easy to implement additional features.**

Some Disadvantages:

- **Reverse Engineering Can Be Easy**. That's why I made this simple documentation XD.
- **May not be optimized**. The implementation of web-based games still has challenges such as performance, because the game may not take advantage of a native system, as it is running on a virtualized system and each implementation has to make *"expensive"* queries to perform a specific task. Fortunately, engineers develop increasingly optimized techniques, such as the WASM (Web Assembly) system that runs code compiled in C/C++ through an API that accesses native resources.



APIs receive a request and send the response to the runtime. The runtime will be responsible for handling the response. [Image 0.4]

This image is simplified as much as possible, to show in a simple way how a game exported by Construct works.

You can see Apis in "operation" and some additional layers have been removed, such as the **Operating System**.

Attention: WEB APIs may not access hardware directly, for security reasons.

It is through these APIs that the runtime can render, handle user interactions, save data and pull information via http.

Resumo

- We saw why game engines are so widely used in the production of thousands of games around the world and how they facilitate work that would be very difficult to do without them..
- We also know the engine in which MDZ was created (Construct 2). **This is quite important**.
- Then we also saw the APIs, fundamental elements that together with the applied logic, give life to the game.

Knowing the Game Engine on which the game was made is a rule for creating Mods! It's the first step and one of the most important things to do before reverse engineering..

That's why I wanted to pass on this very important information.

- **Links**
  - **Construct Site** (Construct 2 has been discontinued, but binary versions are still available for download): https://www.construct.net/en.
  - **The JavaScript Construct 2 SDK Manual:** (Very important to know, as some analyzes will be done based on this documentation): https://www.construct.net/en/construct-2/manuals/construct-2-javascript-sdk
  - **Web Apis:** https://developer.mozilla.org/en-US/docs/Web/API
  - **The MDZ Mods Site:** https://minidayzmods.com/