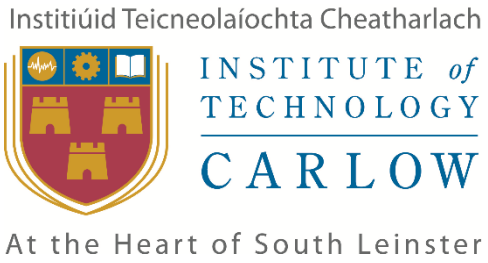


Computer Games Development CW208
Technical Design Document
Year IV

Szymon Piskorz
C00242808
28/04/2022

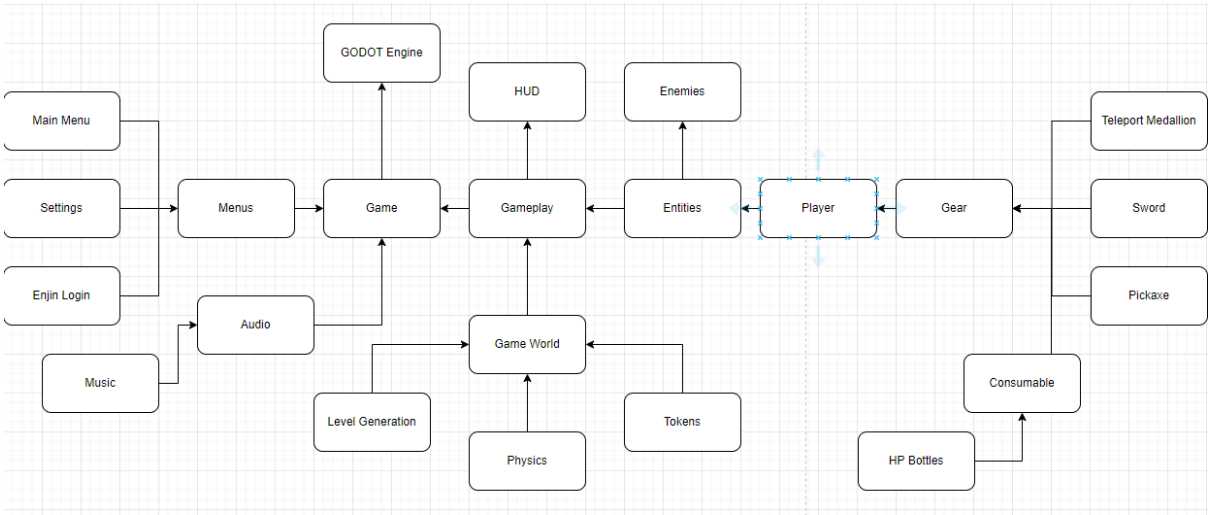
Contents

Table of Contents

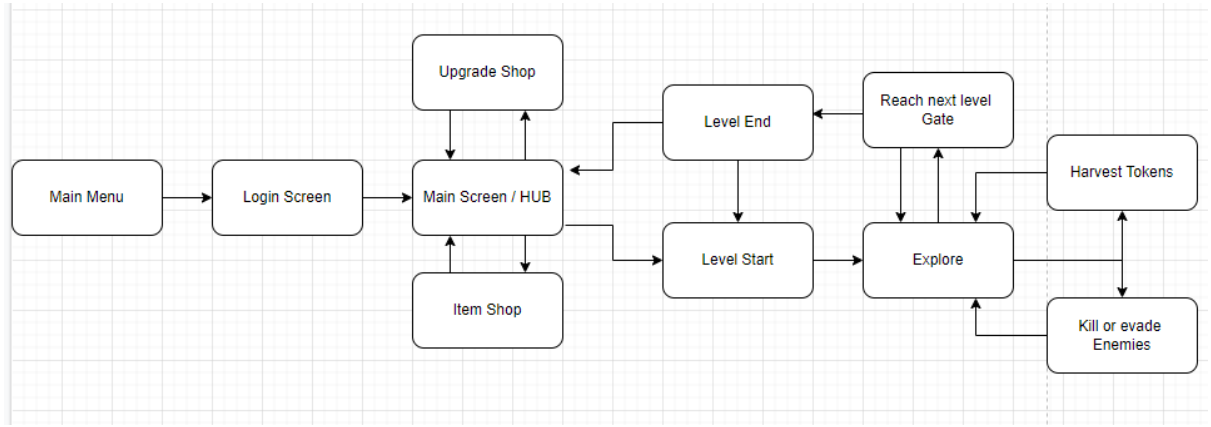
Technical Design

UML Diagrams

Overview Diagram



Simple Game Loop Diagram



GraphQL

For the GraphQL requests and receiving of the results, I used a GraphQL addon for Godot.

The addon was slightly modified for this project changing of the way some of the objects were serialised into GraphQL requests to make it easier to write mutations and queries. A modification was also made to attach access tokens.

Blockchain

Enjin

Enjin is a blockchain service designed for video games, with features such as minting and sending Fungible tokens and Non-Fungible tokens as well as authentication.

Enjin provides a general GraphQL API. The official documentation is out of date so everything I learned was through a website I came across which had the descriptions of queries and mutations, the schema references/definitions in Banana Cake Pop and some trial and error.

There are several different levels of management and authorization in Enjin.

App/Project

The app/project contains all the token, asset and NFT data as well as the user data for a specific game/app/project.

All API request can either be authenticated with the app and / or user access tokens except for creating and / or minting new assets/tokens/NFTs and creating users.

Identity/User

The user refers to an Enjin account that can be used in multiple different apps/projects or a user specific to an app made through an API.

Users contain multiple different fields of information like a name, number of identities and all tokens owned.

Identities share relationships with both apps and users and are mainly used for linking wallets and taking care of transactions. And are also used as an extra cover for specific wallets.

Access Tokens

These tokens are used to authenticate all requests made by Enjin with different levels of security / authorization. These have different sources and different ways to be received.

The app access tokens are received with the AuthApp query when given the APP IP and APP SECRET. These tokens can be used to authenticate any requests made by users and identities specific to the app.

The Enjin user tokens are received with the EnjinOAuth query when the given the email and password for the Enjin account. These tokens can be used to authenticate and request by the apps they own and some requests in other apps that they have identities in.

The player user tokens are received with queries that return the EnjinUser objects using an appropriate Enjin user or app access token. These tokens can be used for specific requests of the current player user in the current app.

Wallet

Enjin uses its own wallet app for managing assets, cryptocurrencies, and authorizing requests. This app allows for having as many Ethereum wallets/wallet addresses which can be linked to different identities for transactions. This wallet address is no different from other Ethereum addresses but the app is used for confirmation and to authorize most of Enjin requests.

Assets

Enjin apps/projects can have many different assets for each app/project which are Fungible and Non-Fungible tokens. Each asset uses the Enjin coin (ENJ) in order to be created and can be converted back into them.

Authentication/Transaction

All Enjin transactions are used to control all blockchain related things like; Minting and Spending tokens, requesting spending of crypto.

All the transactions must the authenticated and authorised in the user/identity specific wallet.

Enjin API

This project implements an API class for Enjin so that it can be used in any of the other classes its needed without having to make new GraphQL requests each time.

This class provides the following:

- Connecting to Enjin.
- Logging in.
- Logging out.
- Getting User data.
- Minting tokens.
- Sending tokens.
- Getting Token balance.
- Creating Identities

This class was originally based off a Godot addon that another classmate started making this year which can be found here. In this project it was modified and developed more while I was working on integrating Enjin with Godot.

References

Web-site

https://kepithorstudios.com/graphbook

https://docs.godotengine.org/en/stable/index.html

https://github.com/Dracks/GodotGraphQL