I need to design a good way to develop games, and in particular the different things/systems/mechanics/components (to be referred to as “Iulion”) that make up the games and the game-objects that are part of the game, and which may interact with the Iulion to generate gameplay (similar to Adrion, I shall refer to them here as “Rygel”: these represent the “objects” that players can possess, use, game-play with, do game-things with, engage in [aynu-game-play] with, trade and generally interact with as items/objects; these include major game-structures like civilizations, characters, Pokemon, Feanor, Mechs, game-structs, Elu-objects, and anything else that is a core part of game-play and the game).

Elysion/Effylion: The ultimate thing I am trying to get out of developing the games

Rygel and Adrion are aynu-things that encapsulate any kind of object/structure in the game. They include items, Feanor, Elu-objects, gameplay-objects, gameplay-structures, Feanor, Mechs, game-structs, game-world-things, [aynu-game-things], [game-play-structs], [aynu-game-structs], [aynu-game-world-play-things], and any other thing in the game, particularly things which the player can engage in gameplay and [aynu-game-play]/[aynu-game-things] with. But they also transcend mortal description, and include features and things that transcend mortal understanding. I can only approximate what the Rygel/Adrion are, but they ultimately encapsulate all of the game-things I desire, are meaningful to the game and its theory/universe/reality, and that are part of the game and its theory/universe/reality that I care about. Anything I desire and which will achieve [Elysion]/[Effylion] is a Rygel/Adrion. Rygel and Adrion include any thing which can engage in gameplay and interact with Gameplay systems and has a manifestation in the Game world, such as a player character, pet, structure, thing, game-play-object, Civilization, cities, player-developed structure/thing that can be used in gameplay, and anything else. Rygel and Adrion also include any player developed things/structures/gameplay-things/game-things.

The Rygel and Adrion will contain and be used to develop/game-develop the things that I desire to possess, can be used to engage in the game-play, game-things, game-world-things, [aynu-game-things] that I find fun/[aynu-fun] and which make me [aynu-happy], [Elysion], [Effylion], and which achieve [Elysion]/[Effylion] for me when I possess them and use them to engage in game-play, game-things, game-world-things, [aynu-game-things], …, [more to develop], … and anything else as determined, created and game-developed by their data-sheet.

Rygel and Adrion are the objects that form a core basis of gameplay, and form the basis of an object-system and mechanic for the game/game-world and object/thing-gameplay, including aynu-things.

[develop more things about the Adrion/Rygel; make them into the ultimate things/game-things/aynu-game-things that will contain the objects/game-things that I desire to possess, and achieve all of the game/game-development/aynu-game-things/Elysion/[Ultimate aynu-goals/things] by possessing them and engaging in object/Rygel-gameplay/mechanics with them (not all Adrion/Rygel will do this, but certain Rygel/Adrion may do [it] for certain players)]

Essentially, Rygel and Adrion can include any kind of thing or structure, including things that players can develop and build-up, that forms the basis of gameplay. This includes Nations or civilization-entities in Strategy games Like EU4, Distant worlds, Starcraft, etc. It also includes things like items or pets from Neopets, or pokemon from Pokemon, or cards from any card-game.

[Rygel is an aynu-concept that encapsulates all the things that exist in the game-world; they can include the objects/structures used to engage in gameplay, like a Civilization/character/Pet/mech/struct/thing/Feanor used to engage in gameplay]

[Rygels, and any other thing in the game, can engage in Object Mechanics (probably to be renamed Rygel Mechanics)]

To start off: each Iulion should get its own folder containing all of the things necessary to create the Iulion within the game: these folders contain everything about the Iulion in the game. The Iulion and its game-things are made up of all the things in the Iulion’s folder.

Every Iulion’s folder contains a Magon, which is the most important document involved in creating and implementing the Iulion, [game-developing] it, and giving it an existence in the game. The Magon’s name is always equal to the name of the Iulion. The Magon contains the main code/text/data/things/script which is responsible for creating the making-up the Iulion. All the documents/files and their code/text/data/things/script are the materials that make up the Iulion; they are the “atoms” that make up the Iulion.

The folder will also contain all of the other documents and document-structures used to create and game-develop the Iulion. Everything needed to fully understand, interpret, implement, create and give game-things/meaning/existence/theory to the Iulion will be contained in one of the documents in the folder. The Iulion and everything about it in the game will be contained in the folder and in one of the documents, particularly the Magon. The folder will contain all of its game-engines, game-data, game-things-documents, game-systems/mechanics/things, [Iulion], [Adrion], [Rygel], Objects/[aynu-things] and any other [game-things], all of which are contained in the documents. The documents, data, files and data/file/document-structures will do everything and anything for the game-development of the Iulion.

[Need to design special documents, document-structures, file-structures, data-structures, files, data, text, aynu-code, code, and folder systems for creating and developing/[game-developing] game-systems/mechanics/things, [Iulion], [Adrion], [Rygel], Objects/[aynu-things] and any other [game-things], [game-world-things], [aynu-game-things]]

There needs to be a core mechanic made up of the “Ultimate document” and game engines that create the object

The Magon for each Iulion is the main engine responsible for creating the Iulion and giving it an existence in the game, the Magon document is the closest single object to the Iulion, and is responsible for creating the Iulion and its theory in the game. However, other documents/files/things may also be part of the Iulion and be responsible for implementing the Iulion and creating it in the game

Magon = main thing creating the Iulion

Other Files = also contribute to making the Iulion

There can be other documents with their own structures and sub-folders and sub-systems, that also play a role in creating the Iulion, and can have their own structure and code/data/content. These documents also play an important role in making the Iulion and creating it in the game.

->for example, there can be a document that describes and encodes the Iulion in either code, aynu-code, special text, or English prose; this contributes to the creation of the Iulion and its implementation in the game: this document is part of what makes the Iulion, its game-engine, and creates its game-play and game-world-existence/theory/reality

The main thing responsible for creating the Iulion will be the Ultimate document (which I shall call “Magon”), which is responsible for creating the Iulion in the game, determining its game-system/theory/reality, and creating everything about the Iulion in the game.

These Magon will be capable of doing everything and anything for the Iulion’s game-development. [Magon will do [the Ultimate thing and aynu-thing I want] for each Iulion’s game-development] (for more, see the Magon document)

Classes of Rygels are themselves Iulion, since they have their own game-theory/mechanics/systems/models/existence/Iulion. Each class of Rygel can be developed as any other Iulion. In addition, they will have a special folder that contains the base data sheet (since some classes of Rygel are special structures that can be developed, built up, and improved through gameplay; these data sheets can be modified/changed for different instances. Though some things can never be changed) for each member of the class of Rygel. Each data-sheet contains, stats, data, aynu-code, script, text, and code that make and create the Rygel-object in the game-world: like in trading/collectible card games, having a copy of the data sheet and the code/data/aynu-code/script/text it contains is equivalent to possessing the Rygel-object in the game, and allows the player to use the Rygel-object in the game, engage in gameplay with the Rygel-object, game-play using [Iulion] in the game/game-world, and do/aynu-do any game-play-thing, [aynu-game-world-thing], [game-world-thing], [game-system-thing], [aynu-game-thing], [Iulion], and game-thing with the Rygel object as encoded in the Rygel’s data sheet and determined by the corresponding Game-mechanic/system/thing/[thing]/Iulion’s game-engine/theory

The text/data/stats/aynu-code/code in the Rygel’s data-sheet are the “atoms” that make-up the Rygel in the game; they are the materials that the Rygel is “made of” in the game.

Each copy of the data-sheet “is” an actual instance of the corresponding Rygel: it is the game-thing/Rygel and the code/aynu-code/stats/text on the data sheet are what create and make-up that object in the game and allow players to game-play with the Rygel object, engage in any kind of [game-world-system-play] with it. Adding a Rygel’s data sheet to the Iulion folder of the class adds that Rygel’s data to the game and makes it possible for the game to generate more copies of the Rygel as required by the game and as a result of players engaging in gameplay and as allowed by the legitimate game-mechanics-theory. Copying the data sheet is equivalent to creating a new instance of the object/Rygel in the game, and control over the Rygel’s data-sheet document is equivalent to control over the Rygel in the game. The game regulates the existence, ability to modify/change Rygel, and ability to produce these Rygel by controlling access to these data-sheets and the ability to modify their data/code/aynu-code/text. Only the game-mechanics-engines or a game administrator can modify, copy, or create the Rygel data sheet, and players can only cause the game-mechanics-engines to give them Rygel or modify them by engaging in legitimate gameplay and exploiting the game-theory to understand what game-play things they need to do to get the game-play-engines to create/build the Rygel they want and give them access and possession-rights to the Rygel they want.

-Players possess Rygel and have possession rights when they are in one of several locations in which a player’s own Rygel may be kept. This includes, but may not be limited to:

- the player’s inventory

-safe-storage locations

-a player’s bank storage

-a player’s home/base/fortress

-the player’s shop

-the player’s gallery

-given to another player but endowed with a return tag

-when they are placed inside of other Rygel the player possesses

-when they are placed in certain things that do not take away possession of the Rygel (for example, when the object is placed in a work-bench for crafting, similar to Master Crafting in Path of Exile)

-when they are part of a structure or thing the player controls or where the player has received a guarantee that they still own the Rygel

-and many other things, which may be part of other game-mechanics, but do not take away the player’s possession rights when Rygel are interacting with them or are given to the thing

Classes of Rygel: This is the broadest category of Rygel and groups different types of game-objects based on a broad set of game-systems/mechanics/models/things and Iulion that they share. Examples of classes of Rygel include Items, Elu-objects, Feanor, Mechs, game-struct classes, and much more.

(Types of) Rygel: Classes of Rygel may be further subdivided into different groups. However, each Rygel is created from a base data sheet that determines the Rygel. Drawing on the example of Neopets, two Rygels are the same if they are the same “basic object”. Which means that, unless the Rygel has been modified using some sort of game-mechanics/system/Iulion, the two Rygel have the same base data-sheet. This is essentially equivalent to them being the same type of item: in the same way two identical apples are the same item, but different instances. For example, if I have a “Gate Guardian” card, the card I own is an instance of “Gate Guardian”, while its type is “Gate Guardian”, and its class is “Yu-Gi-Oh Card”.

Instance of a Rygel: This is the actual copy of the data sheet, and the manifestation of the Rygel in the game; players can own/possess and gameplay with an instance of a Rygel. Similar to trading card games, having a copy of the card (and therefore being able to use the card in a game) is the same basic mechanic as having an instance of a Rygel and its data-sheet.

[continue to write and add stuff to create my Game Development Theory and System, which will help me better construct a system that I can use to develop my game and the ultimate things I desire and want; the ultimate thing/game that is my desire to create and possess and keep safe and preserve forever]