Zachary Waters

Professor Waters

Objects and Design

17, April 18, 2018

Thousand Parsec

Introduction:

Thousand Parsec is a framework designed for multiplayer turn-based space empire building games. Originally started in 2002 by Tim Ansell and Lee Begg, with the initial goal of simply being a clone of the more popular space empire games at the time, the project evolved into becoming a framework, which would allow users to design and play their own space-spanning empire games on top of it by modifying Thousand Parsec's central rulesets. Thousand Parsec's game creation framework is designed to produce a specific group of games, known at the time as 4X games; named from the four main phases of gameplay: eXplore, eXpand, eXploit and eXterminate. The rise and fall of Thousand Parsec speaks to time sensitivity of the public's desires, Thousand Parsec had a slow start and by the time it was first playable in 2006, the landscape around it no longer resembled that from once it had begun. The Gaming had long since moved beyond the genre of space empire games, and into the realm of 3d graphics powered by console hardware. Despite this Thousand Parsec persisted for another six years till 2012, when the project was officially discontinued.

Description:

Thousand Parsec operates on Client-Server architectural model, which has three major components. The first is the client, which operates on the user's side and acts as their conduit through which the player can interact with universe. The client then uses the second component the protocol, which allows it to communicate with the final and most important component, the server. The server is what holds all information about the current game and performs all the necessary operations to determine the state of the game's next turn. This gets relayed back to the protocol, and then the client where the user can see and respond. Thousand Parsec is packaged by layer instead of by feature, divided between the server, protocol, and client classes. The design pattern that Thousand Parsec uses resembles that a of command behavioral design patterns which we learned in class, but in more advanced terminology it uses a multi-client server application design pattern. This is what allows it to keep track of the many players partaking in the multiplayer games.

Analysis:

The process in which a game of Thousand Parsec begins, is by probing the metaserver, a supporting server held by the game creators themselves. The metaserver acts as an aggregator for other Thousand Parsec servers, collecting information about the various games that are taking place. The "soon to be gaming" client is provided with a list of this information and is prompted to pick whichever game they wish to join, they are then redirected to that particular game's server, where once a connection is established the client then uses the protocol to request the entire Catalog from the server. The Catalog holds all the objects, components, designs, categories, resources, boards, messages, properties, and players that define the state of the game.

Once the client has downloaded all the of the necessary information the client now begins plotting the information into a representation of the game for which the player can understand. Now the default starting scenario for any new player in a server is determined by whichever ruleset the game happens to be selected. The "Missile and Torpedo Wars" ruleset starts the player with a single randomly generated home planet, and two automatically generated scout fleets. Every turn each player is given an allotted time to decide for how they wish to spend their turn, once time is up everyone's turn happens simultaneously. From then on, it's up to the player to eXplore, eXpand, eXploit and eXterminate their way to victory.

The way in which the game allows the players to interact with, and design their own ruleset is through a rather extreme form of object-oriented design. Everything in the game is designed to be an object, from the equipment on the ships, to the ship themselves, to the galaxies and planets they play in, all the way up to the universe itself, its all objects. Thousand Parsec provides five default game objects; Unversed, Galaxy, Star System, Planet, and Fleet, but the system allows players to extend these classes to create entirely new objects for which to populate their own games and rulesets with. Thousand Parsec also allows any would be game designers to create something called "Orders" which can be attached to Fleet and Planet objects. Think of Orders as the various actions a player can undertake during their turn. For example, a universally used order is the "Move" order, which is self-explanatory. Despite the mandatory nature of Orders Thousand Parsec does not define any by default, and thus it's up to each individual ruleset creators for establishing a system of Orders. The reason Thousand Parsec doesn't provide some default orders was because some rulesets like "Risk" require an entirely different way to move fleets around than say "Missile and Torpedo Wars" and providing a default move would result in one of the games either being ruined or having the developers try to find a way to turn off basic functionality. Personally, I feel that if Thousand Parsec, which calls itself a 4X gaming system isn't willing to provide default Orders, for fear of locking down players to a specific gaming ruleset, then it shouldn't call itself a 4X system. Players go about designing their own creations through the Thousand Parsec Component Language, or TPCL, which is written for the language of Scheme due to its simplicity.

The client-server architectural framework for which Thousand Parsec is built upon relies heavily on the link between the server and the client, known as the Protocol. Think of the Protocol as a phone operator balancing the many communications going back and forth between the Server and the Client, in a cycle where every turn the server sends the game state information to the Client, and the Client sends back a list of Orders and Designs to be processed. The way the Protocol handles all this information is through a system of frames, which you can imagine as a packet of information. All these frames begin with the Header frame acting as a parent class. From the Header frame comes the following two most important frames, the Request and Response frames. These two types of frames are what are the key for how the protocol allows communication between the server and the Client. Thousand Parsec has also done something else which is particularly interesting, it externalizes its AI players in the same fashion as players, separating them from the games hidden information preventing them from "cheating." Now despite how pleasant this idea sounds, I am against it for a variety of reasons: firstly, allowing AI to cheat is an easy way to create challenging enemies for competing players without wasting precious time trying to create a machine capable of playing the game fairly, and being good at it.

Conclusion:

In conclusion, Thousand Parsec is both a story of success and failure. The project was ambitious, free, and open source, spearheaded by their incremental design process which allowed seamless and organic growth to the framework. The client-server architectural model was also a vital key in the early success of the project, as it allowed development of clients away from any of the game's logic. Despite all of its successes Thousand Parsec failed at its most important job. Being a fun game. At some point in its development Thousand Parsec stopped being about a game and became more about being a software. Thousand Parsec was brought to coding conventions and was being used to teach students and interns design principles, features that could have improved the games marketability to the general public were ignored in favor focusing on perfecting the project's framework and design. And now after the project's termination we can see exactly what that got them, a program with some really great code, but a game that was forgotten due to being out-competed, and out-enjoyed by every other game on the market.