

Final Project Proposal

15-112
Section P

Eugene Lee
eleehua.j

Project Description

Ctesiphon - World in a Bottle Ctesiphon will be a **sandbox civilization simulator** on a procedurally-generated world. In particular, it will be a mostly hands-off experience focusing on the interactions and evolution of civilizations and cultures over time. Players will be able to view the simulation as it evolves over time, as well as slightly influencing events through various means. Specific facets that will be focused on will be discussed further under "Structural Plan".

Competitive Analysis Several grand-strategy games are the inspiration and the closest comparison for this project. In particular

- **Civilization** - Civilization is probably the most well-known game in this genre. Civilization tends to be more focused on being a game, with the growth and evolution of civilizations and technology being more of a theme than the centralizing focus, and the underlying dynamics being shallow or absent. Ctesiphon will be similar in that it begins with a mostly-empty, dynamically-generated world.
- **Europa Universalis (EU)** - EU is another extremely popular grand strategy game. The main focus on EU is mainly around combat and economics, while Ctesiphon will be more focused on the 'softer' aspects of politics and culture. Furthermore, EU, as a historical game, is tightly linked to real-world civilizations and the world itself, while these will be dynamically-generated in Ctesiphon.

In general, Ctesiphon will be more of a simulation than a game, like watching ants in a glass box, and hence can have more realistic processes dealing with cultures / politics etc., than these existing games.

Structural Plan Each large feature to be implemented will be segregated into a separate file.

- **init** - The main file, calls the other files as required.
- **Mapmaking** - The methods to generate the world, which will be dynamically generated. This will include two submodules:
 - **Voronoi** - The map will be generated using a Voronoi diagram, to introduce irregularity and variety. Since this is pretty complicated, it will be in its own module.

- **Terrain** - Dynamically generate altitude, climate, rivers, and other geography.
- **Culture** - Cultures will be an integral part of the dynamics of Ctesiphon. Cultures will dynamically interact with the environment, and each other, in different ways. They will also merge, evolve, and interact over time.
 - **Language** - Like Culture, Language will also be a dynamic system. Language is independent, but correlated with Culture, and can affect relations and interactions. They will also evolve, merge, and diverge over time.
- **Polity** - Politics (like countries, empires, duchies etc.) will be the bodies through which political actions (like wars, communication, etc.) occur. They will also be the main visible aspect of the simulation. There can be super-polities and sub-polities, alliances, civil wars leading to separation etc.
 - **City** - Cities will be the smallest fundamental unit of the map, and polities. Cities can grow over time (due to technological progress of the cultures within the city), and shrink (due to war, disasters etc.)
- **Interface** - Essentially just the view methods, displays the screen and interface.

Other additional features (like individual armies for better simulations of war) will be implemented after these.

Algorithmic Plan Here we will briefly discuss the implementation of each of the features.

- **Map Generation:** The map will be generated using Fortune's algorithm, producing a list of Cities, each storing lists of neighboring Cities (this will help later). Terrain will be first generated by doing a BFS floodfill at random points of random sizes to create land / sea, then adding noise to create altitude. Rivers will be spawned at high points and flow downhill, and then the 'wetness' (fertility) of each city will be calculated based on its proximity to water. Temperature will also be calculated, and this will determine the biome of each city.
- **Culture Generation:** Cultures will be essentially a set of 'traits', which influence the way polities / cities dominated by that culture will act (like a Culture predisposed to city-building will build up their cities, a xenophobic Culture will try to remove other cultures from cities they dominate). These traits will be loosely based off the initial random spawn.
- **Cultural Evolution:** Cultures will spread to cities adjacent to cities that they're present in. If a culture is in a polity in which it is a sizeable minority, it may splinter from the main culture and form a new one, and diverge from there. Cultures can pick up new languages / traits from other cultures that are in polities with it. Cultures can also merge (diverging from their ancestor culture) with sizeable minorities it shares polities with.
- **Language Evolution:** Language will be mostly visible through city names and polity names. They'll be characterized by a set of letters used, typical syllabic structure, and some Markov chain probabilities that determine their 'look'. These can shift over time, and merge with other languages (like when parent cultures merge).

- **Polity Generation:** Each City is automatically in its own Polity. Larger Polities can be formed via mergers (stored as a tree, with the 'capital' as the root node of the tree) with alliance partners, other polities that share cultures / languages or defeated war enemies.
 - **City Mechanics:** Cities have their own *Fertility* and *Economic Output*, which are functions of the underlying geography and improvements made by the polities that own the city.
-

Version Control Plan I'll be storing my project in a Github repository.

Module List I won't be using any external modules (just Tkinter).