ARTIFICE

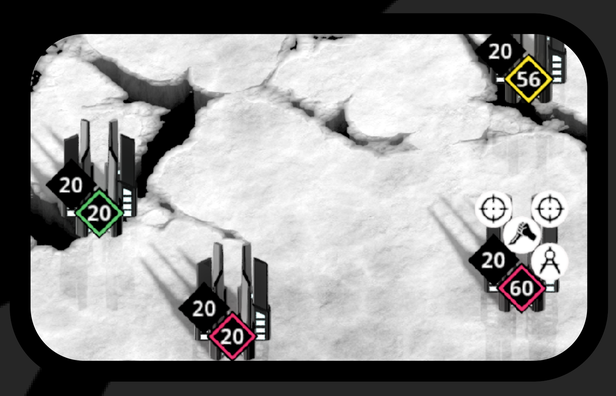
Artifice is a project that aims to recreate and improve upon the gameplay of the award-winning iOS game Subterfuge, which has been all but abandoned by the developers despite maintaining a dedicated user base.

Artifice is a mobile real-time strategy game. The premise is one of global conquest, where matches span days rather than minutes. Set at a pace realistic for a real war, the game leaves plenty of room for diplomacy rather than focusing purely on reaction speed or tactics.

How the Game Works

In Artifice, all players are placed on a map with their own outposts, units, and specialists. There are 10 neutral outposts outside of the 5 nearest outposts assigned to each player.

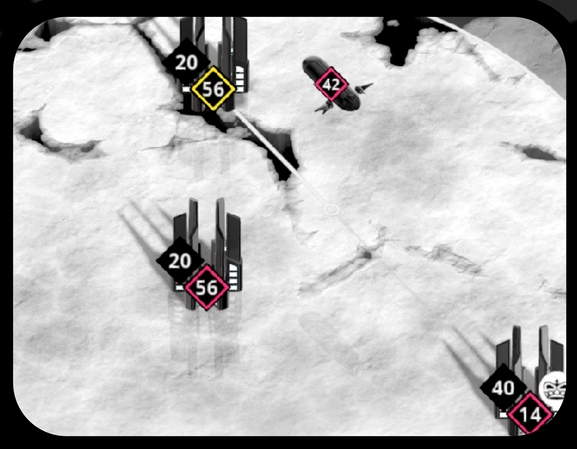
In order to win, a player must obtain a high amount of valuable resources (200 by default).



- **Factories make basic units. These units can be used to attack and take over other outposts, or to mine for valuable resources to progress towards winning the game.**
- **Generators power your basic units and factories. Generators increase the rate at which factories can produce basic units**
- **Mines produce valuable resources. At the start of a normal match, no one owns a mine, so you must find a way to obtain them by diplomacy, force, or drilling your own.**

From left to right: Factories, Generators, Mines

- **Players can attack their opponents' outposts and reinforce their own by transporting units in airships**
- **Battles occur whenever airships on the same route collide, or airships arrive at their targets**
- **Rather than the victor being determined by which player has the most units, specialists apply a variety of effects during battle to alter the outcome**
- **After a successful battle, a player can take control of another outpost**
- **When players lose their starting specialist, their Queen, they lose the game**



Milestones

Starting Point: at the beginning of the project for this semester, the vast majority of the game logic was already implemented through reverse engineering, as well as a sizable portion of the requisite server-side infrastructure . While the game logic supported player interaction, the user interface side was unimplemented and therefore users could not play the game.

September
- **Base multiplayer functionality**
- **Implementation of out-of-game UI (creating and joining games)**
- **Account settings**

October
- **Implementation of in-game UI**
- **Game logic cleanup and fleshing out game features**
- **Taking into account custom game settings for simulation**
- **Drag-to-send**
- **Battle forecast**
- **In-game player status screen**
- **Chat**

November
- **Map generation**
- **Ability to win a game and update ELO**

December
- **Consistent style for UI**
- **Animations for certain game events (e.g. explosions)**
- **Art assets**
- **Meet iOS app store requirements and submit for beta testing**

Details

The client for the game is implemented using the Godot Engine, which is an open-source game engine that supports both 2D and 3D development. The default language for the engine is GDScript, which is the language that we used for most of the scripts we wrote for Godot.

The actual game logic itself, which needs to be run both locally and server-side, is written in C++. Godot provides GDExtension, the 4.0 successor to GDNative, to use C++ libraries in Godot.

We plan to release to iOS and Android.

Going Forward

We are in dire need of 3D artists!

If you have previous experience with Godot Engine, or if you are very comfortable with C++, then we would love to have you in our team!

For next semester, our plans are to start running beta tests on controlled samples before releasing the game to the public. We'd also like to start experimenting with new features and additions to the game, since we've just been recreating existing functionality so far.