PulseCraft

"The only choice is war. The only allies are enemies."

ECE Pulse 2013 Software Competitions Instructions

Background

The year is 2405.

War has ravaged the land. The landscape torn up to a husk of its former self. We didn't know the war between Google and Bing would turn out this way but I guess life is full of surprises. Ever since Bing realized they wouldn't beat Google in the western hemisphere Bing decided to market to the East. We all knew that Google would take over the world but Bing was another story. Now here we are. In the middle of the war. The world has been scorched. Weather changes at random. The Earth's people are searching for a stop to the fighting. Searching for an end.

In the field of war and destruction, you, an untried untested commander, have been given the privilege of destruction. In a world of capricious slaughter and meaningless violence, you decide to make a difference. Google/Bing has given you a chance and you will not disappoint. The world *will* know peace at your hand.

Objective

Competition teams are tasked at battling one another in the PulseCraft arena. The goal is to implement a competent artificial intelligence (AI) to play the game. The AI is tasked with building and directing an army to sweep through the land and destroy the enemy. An advanced AI will take into account various game mechanics and be adaptable to the many different opponents you will face.

Gameplay

The game is a 2D turn-based strategy game visualized in a top-down view. The goal of each AI player is to destroy the other player's base. The first player's base is located in the top-left corner of the map, 5 squares from the top, 5 squares from the left. The second player's base is located in the bottom-right corner of the map, 5 squares from the bottom, and 5 squares from the right.

There is *no* fog of war. All units are visible at all times. All bases are available at all times. The health of each unit and each base is available at all times.

Maps

Maps are 2D arrays of tiles. Map tiles can be either Ocean tiles, Mountain tiles, or Plains tiles. Map files are specified in a simple ASCII format and will be included in the code distribution. Note that maps may be of any size, so ensure your AI can handle different types of maps!

Units

You are allowed to create and organize armies containing 3 types of units. All units have the same base health and range, but their effectiveness in battle depends on the opposing unit's type.

- Mechs (Unit Type 0): Infantry, Specialized in the destruction of vehicles

- Tanks (Unit Type 1): Mechanized, Armored units with short range bombardment.
- Artillery (Unit Type 2): Mechanized, Ranged bombardment unit.

Units can move either across two plains squares or a single mountain square per-turn. Units *cannot* move across an ocean square. All units cost the same amount to purchase per turn. There is *no* concept of range in the game. Units can only attack units that are located in adjacent squares.

Despite the unit description mentioning range there will be no concept of range in the competition. Range is only something we're using in our lore and its mostly something for your imagination

Operations

AI players can perform four types of "operations" or actions in the game. The operations are as follows: Move, Attack, Buy, and CheckWeather.

- Move: Allows an AI player to move a unit to a target square on the map.
- Attack: Allows an AI player to specify a unit to attack an adjacent unit.
- Buy: Allows an AI player to purchase more units.
- CheckWeather: Allows the AI to query for the current state of the weather. This is explained in detail in the Weather section.

Players are penalized for making invalid operations, such as trying to stack units during a Move or Buy operation or attacking a player that is out-of-range. More information on the effect of invalid operations is explained in the Operation Queues section; more complete information on what operations are invalid are in the API documentation.

Operation Queues

Each player is given a queue of operations that they can issue operations to. Each player can only issue as many queue operations as the length of his or her queue allows.

If a player issues an *invalid operation*, such as a Move to a location not on the map or occupied by a unit, the Operation will fail and half of a queue slot will be lost on the next turn. This discourages extremely aggressive clients. Every operation except for CheckWeather occupies a single queue slot. CheckWeather takes up half of an AI player's queue.

Weather

Central to the concept of attacking in this game is the concept of Weather. Weather affects the ability of a unit to damage another unit. Under normal weather conditions the Artillery shreds the unarmored Mechs, Tanks dominate slow moving Artillery, and Mechs can outmaneuver short ranged Tanks.

Diving weather is expensive. Due to the heavily altered state of the landscape in this time of total war, highly skilled meteorologists must be employed to obtain the status of the capricious weather. Player AIs must be conservative in their desire to check the state of the current weather, since checking the weather comes at the cost of performing other operations.

During a rainstorm, Artillery lose their accuracy to hit smaller Mech units but can hit Tanks slowed down by mud, Tanks however now have an advantage over Mech units who are exposed to the harsh rain environment. Mechs being harder to target in the rain now can approach Artillery without being blown away from a distance.

When faced with a Heatwave, Mechs gain a dominance against both Tanks and Artillery due to crew members being baked alive in literal "hot boxes."

When a Blizzard hits, Tanks and Artillery gain dominance against Mechs who are exposed to the harsh winter environment.

To sum up the weather phenomena and their effects on units, refer to the following list

- Normal (Weather Cond Normal): Mechs > Tanks > Artillery > Mechs
- Rainstorm (Weather Cond Reversed): Tanks > Mechs > Artillery > Tanks
- Heatwave (Weather Cond Dominate): Mech > Tanks = Artillery
- Blizzard (Weather Type Weak): Tanks = Artillery > Mechs

The organizers of the competition will not, at any time reveal the nature of weather generation. The only way for a player AI to query the status of the weather is to issue a CheckWeather operation. The CheckWeather operation occupies half of the player AI's queue.

Buying Units

Each AI player receives a fixed amount of gold per-turn. The AI player can choose when and how to spend the money it receives per-turn. Gold can be used to purchase units. Units must be built with a starting position that is within 2 spaces of your own base; they may not stack, including with the base itself.

The Arena

In order to facilitate player testing, the organizers will be hosting a set of Arena servers. Each Arena server allows individual player AIs to play against each other. The Arenas are capable of keeping stats of each individual team.

It is *highly recommended* to enter the Arena as soon as possible, even without a fully implemented AI. Entering the Arena will give the player a much-needed look into the logic of the other players. Different Arenas will be configured to serve different maps or different weather patterns.

Local Testing

To further facilitate player testing, each player will be allowed to locally run a copy of the server. The server code we distribute will not keep track of stats and will come with a configuration file that exposes tweakable settings such as weather patterns. Note that you may construct whatever conditions you like locally, but have no control over the arena configurations.

Rankings

The ranking is based on your client's match history in the Arena servers. We will have 3 Arenas up, an Easy Arena, a Medium Arena, and a Hard Arena. Based on your performance Arena, we will place you in the Software I, II, or III bracket. You must play games in the Arena you wish to compete in to compete in that particular bracket. Note that we will be analyzing the Arena standings by hand. If you initially competed several times in the Medium Arena but later moved to the Hard arena where your performance was okay, we will consider you to be entered into the Software III bracket.

The two GPUs will go out to the top 2 winners of the Software III or Hard bracket, there is no distinction between Software III.1 and Software III.2. Make sure to compete in the Arena corresponding to the bracket that contains the prize you wish to compete for.

FAQ

Thrift

What should I do next?

Read the Thrift protocol definition. The protocol definition can be found in game.thrift. The game designers highly suggest reading the game.thrift file for detailed design considerations.

What is Thrift?

Thrift refers to Apache Thrift, a cross platform abstract language which allows us to specify and deploy a language agnostic description of the communication protocol. The following definition is taken from the Thrift Website.

> The Apache Thrift software framework, for scalable cross-language services development, combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml, Delphi and other languages.

Why did Pulse choose Thrift?

Thrift is language agnostic. It eases the burden on us for supporting multiple clients in different languages. Thrift also allows programmers comfortable in languages we do not support to interface with our protocol description.

I wish to develop in a language that is not officially supported. What should I do?

Read the Apache Thrift tutorial and try to implement the services we define in game.thrift. If you are having trouble connecting to our server with your Thrift distribution, consider interfacing with our wrapper script, which is how C++ is supported.

We will only officially support C++ and Python and teams using those languages will receive priority in our support. *If someone knows Java well and helps us out we will be able to officially support it.*