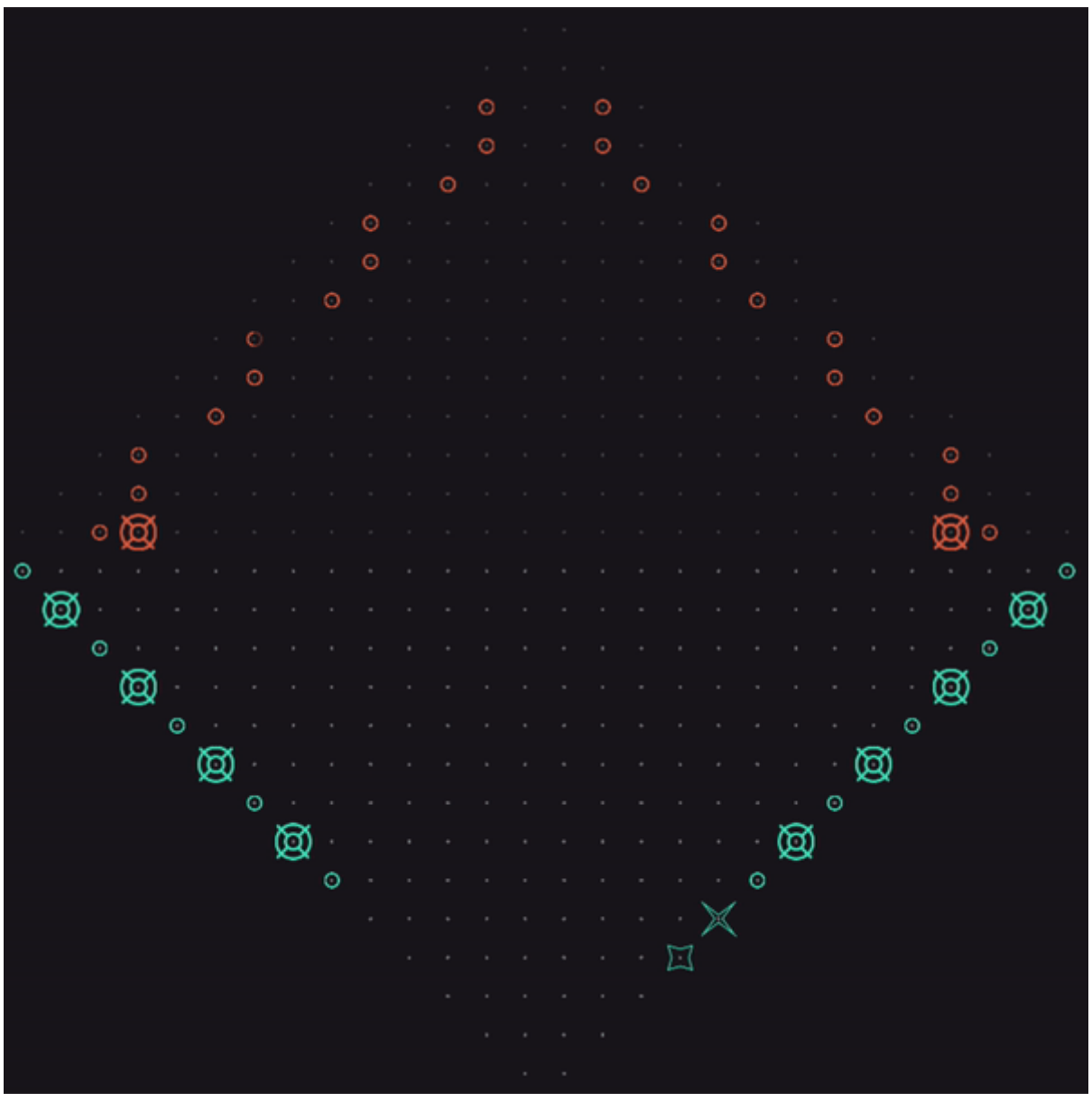
GAMEPLAY OVERVIEW

Correlation One's Terminal is a member of the Tower Defense game genre. It is a two-player, simultaneous-turns game that takes place on a diamond-shaped arena. One player occupies the bottom half of the arena, while the other player occupies the top half. The objective is to reduce your opponents health to zero. You can do this by advancing **Mobile units** to your opponent's edge and building **Structures** to protect your own edges.



All units have **Health** and when their health reaches zero, they are destroyed. Some units will **Attack** and attempt to destroy enemy units that enter their **Range** of operation. Throughout the game, both players are provided two resources - **Mobile points** and **Structure points** - which are used to create Mobile units and Structures, respectively.

The key differences between Mobile units and Structures are shown in the table below:

Where is it Deployed?	From either of your two arena edges	On any square in your half of the arena
How does it Move?	Moves to opposite arena edge, using pathing algorithm described **below**	Stationary (does not move)
What does it Target?	Targets all enemy units	Only attacks enemy Mobile units
Can there be multiple 'stacked' in one location?	Yes	No
Do they block movement?	No	Yes
What resources are used to Deploy it?	Mobile points	Structure points

Units differ in their cost, health, damage, and range of operation. Click on the drop-downs below to learn more about Units.

▼ MOBILE UNITS - ATTACKERS:

Mobile units are deployed from either of the two edges on the player's side of the arena, and aim to reach the opposite edge in enemy territory. They attack enemy Mobile units and Structures while moving. Some Structures attack incoming enemy Mobile unit units, acting as a defense.

If Mobile units successfully reach the opposite edge, they decrease the opponent's Health by 1 point and award 1 Structure points to the deploying player; they then disappear from the arena.

There are three types of Mobile units. The **Scout** is a fast-moving unit which deals light damage and is useful for scoring. The **Demolisher** is expensive and easily destroyed, but it's high damage and far range and can wreak havoc on enemy defenses. The **Interceptor** is a high-health unit that deals high damage to enemy Mobile units, but cannot attack enemy Structures. The three types of Mobile units and their characteristics are detailed below:

	UNIT IMAGE	COST	HEALTH	RANGE*	DAMAGE	SPEED*
Scout		1 Mobile points	15	3.5	2	1
Demolisher		3 Mobile points	5	4.5	6	2
Interceptor		1 Mobile points	40	4.5	20	4

*Range: Maximum euclidian distance of a targetable coordinate

*Speed: Frames required to move one space

▼ STRUCTURE UNITS - DEFENDERS:

Structures are stationary units that do not move. They block the paths of both friendly and enemy Mobile units, and no two Structures can occupy the same location. They persist across multiple turns.

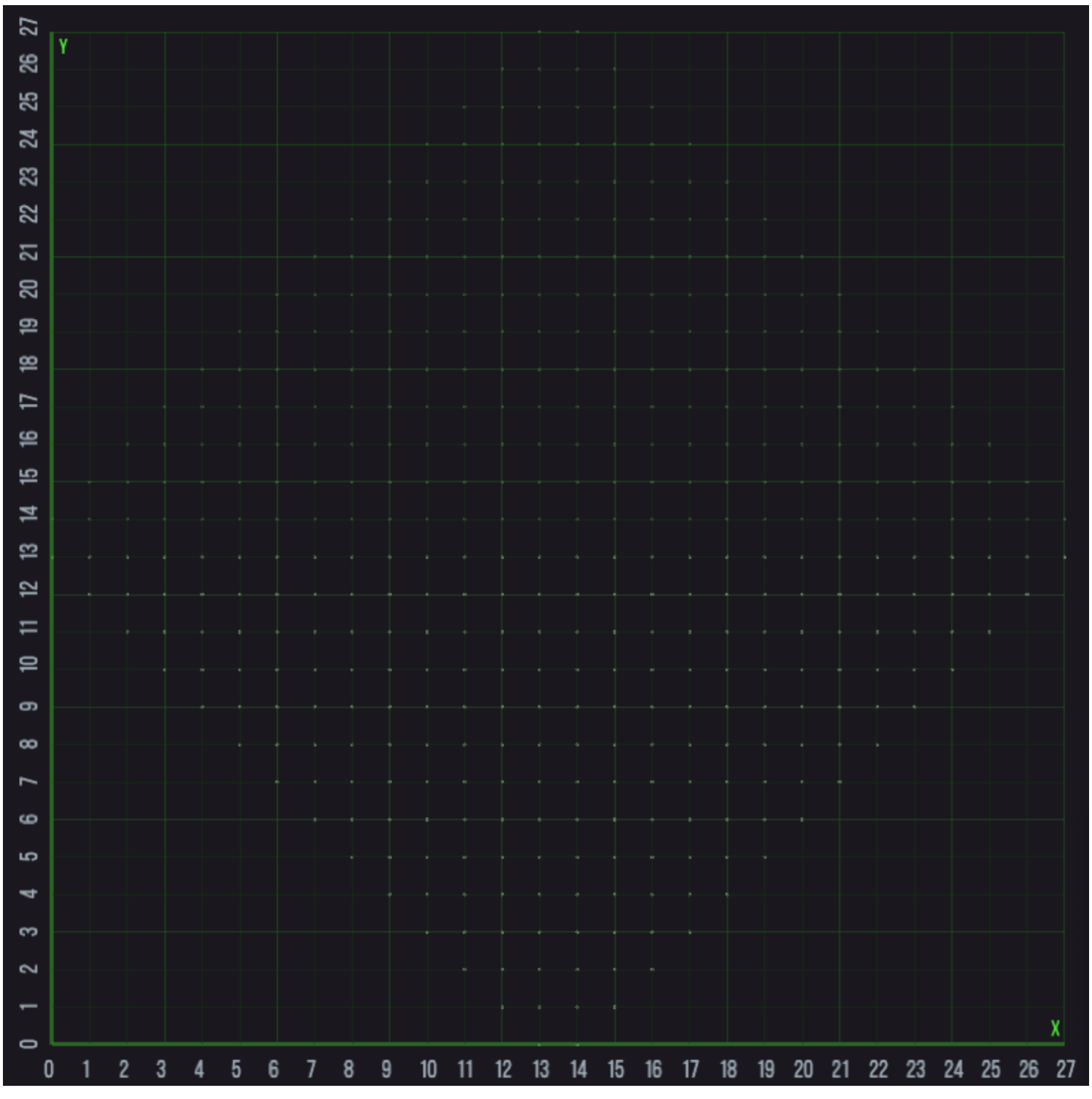
There are three types of Structures. The **Wall** is a cheap, simple Structure used to influence the paths units take and to protect more valuable Structures. **Factorys** provide additional health once to friendly Mobile units that pass within range. **Turrets** attack enemy Mobile units. A Structure unit can also be upgraded to gain stats. Missing health persists when a tower's health is increased by an upgrade. The upgrade cost is the same as the units base cost unless otherwise stated. The

	UNIT IMAGE	COST	HEALTH	RANGE*	DAMAGE	ENCRYPTION	UPGRADE
Wall		1 Structure points	30	N/A	N/A	N/A	Cost: 1 Health: 120
Factory		4 Structure points	30	N/A	N/A	0	Cost: 3 Range: 7 Shield: 4 Shielding increased by 0.3 * Y location
Turret		2 Structure points	90	2.5	5	N/A	Cost: 4 Damage: 16 Range: 3.5

*Range: Maximum euclidian distance of a targetable coordinate

MAP

Units in terminal are placed and move along a diamond-shaped grid. The coordinates on the grid range from 0,0 to 27,27. Note that half of these coordinates fall outside the diamond shaped gameplay area.



GAME START

You can see each players' available resources at current point of game playback in the upper corners of the game board. The gauge represents your owned proportion of the total resources available between the players.

TURN STRUCTURE

Each turn is split up into three phases: Restore, Deploy, and Action. In the restore phase, players are granted resources. In the deploy phase, players choose where they want to deploy their units. In the action phase, deployed units act automatically according to the game rules.

1. "RESTORE" PHASE

Players lose a fraction of their stored Mobile points from the previous turn and then are given additional Mobile points and Structure points. Players are also shown the current game state. The below resource schedule explains how resources are allocated during the game.

▼ MORE

At the start of each turn (except the first one), a "decay" mechanic is applied whereby each player automatically loses 25% of all Mobile points stored from the previous turn. Following this players are given 5 Structure points and 5 Mobile points, plus an additional 1 Mobile points for every 10 turns that have passed. For example, at the start of turns 0 through 9 you recieve 5 Mobile points, at the start of turns 10 - 19 you recieve 6, and so on.

In addition to the resources gained at the start of each turn, players gain 1 Structure points for each point of damage they dealt to their opponent's health in the previous turn.

Note: Mobile points decay amount rounds to the nearest tenth.

2. "DEPLOY" PHASE

Players select locations to deploy Mobile units and Structures using their accumulated Mobile points and Structure points. Players can also choose to remove existing Structures for some refund. Once both players have made their selections, the Deploy phase is over, and the Action phase begins.

Players send commands that list where and how many Structure and Mobile units they wish to deploy. During this phase, the player can see enemy Structures which survived the previous action phase, but not the new Units their opponent is deploying. Players will have 5 seconds to submit their commands each turn, after which they will begin taking 1 damage per second.

Players are also allowed to remove previously built Structures from the arena in this phase and receive a refund. This refund is equal to 7500% of the initial cost of the Structure, times the percentage of the Structure's original health remaining at the time of removal.

$$Refund = 75 * InitialCost * (RemainingHealth / OriginalHealth)$$

Structures that are selected to be removed will be marked visually and will be removed at the end of the action phase.

Note: Structure refund amount rounds to the nearest tenth.

3. "ACTION" PHASE

The game engine deploys Mobile units and Structures according to player choices made in the Deploy phase. The Action phase progresses in discrete Frames and continues until all Mobile units are destroyed or reach the opponent's edge.

▼ MORE

After both players have chosen which units they wish to deploy, the game engine deploys Mobile units and Structures according to the commands sent in the Deploy Phase. Then, based on the movement and targeting logic (detailed later), the game automatically controls each player's units for the duration of the Action Phase. The Action Phase has many discrete Frames and continues until all Mobile units are destroyed or reach the opponent's edge.

Each Frame during the Action Phase is sent to both players. Players cannot send commands during the Action Phase, but can observe, collect information, and plan for the next turn. After the Action Phase, the next turn begins and the cycle repeats.

During each frame, actions will occur in this order:

1. Each unit takes a step, if it is time for them to take a step.
2. Each shield decays. See 'Shielding' in **Advanced** for information about shields. Note that in some rulesets of terminal, shields do not decay.
3. New shields are applied, if a unit has entered the range of a friendly Factory
4. All units attack. See 'Targeting'
5. Units that were reduced below 0 health are removed

GAMEPLAY END

If a player reduces their opponents health to zero, they will win. If the 100th round is completed, the player with the highest health will win. If both algos have the same health at the end of the 100th round or lose all their health in the same frame, the algo with the lowest computation time will be declared the winner.

UNITS - ADVANCED

targeting that is only needed by advanced users. For more basic unit information, see the **Gameplay Overview**

▼ MOVEMENT AND PATHING:

Movement and Pathing are automated by the game's engine. Details of the information movement logic, as well as edge cases when movement is blocked, are described below.

PATHING LOGIC

In general, the Mobile units will always take the shortest path to their destination, and will prefer to zig-zag rather than moving in a straight line for extended periods of time.

Each Mobile unit moves at its speed detailed above, taking one step after the required number of frames have passed. Mobile units can only move left, right, up, or down (not diagonally). Each time a Mobile unit takes a step, it will choose the most ideal tile to step onto using the following logic:

1. Choose the tile which is the shortest number of steps from the Unit's destination, described below.
2. If multiple tiles are equally close to the units destination, move in the opposite direction of the previous movement. For example, if the Unit made a vertical move on its previous step, it will prefer a horizontal move.
3. In the case where a Unit has just been deployed and has yet to move, it will prefer a vertical movement.
4. If there are two tiles with equal distances and are equally prefered based on direction, the unit will choose one that is in the direction of it's target edge. For example, if a unit wants to reach the top-right edge, and must choose between moving left and right, if both paths have the same minimum number of steps it will move right.

CHOOSING A DESTINATION

A Units destination is usually the opposite edge of the edge it was created on. If the opposite edge is unreachable due to structure placements, the Unit will instead attempt to reach the deepest possible location in the enemy territory, and then self destruct as described below. The deepest location is the location with the furthest Y coordinate from your territory. If multiple such locations are reachable, the Unit will choose the one closest to its target edge. For example, A unit attempting to reach the top-right corner who can reach [13, 26], [14, 26] and [15, 25] will choose [14, 26]. First, it will narrow its search to [13, 26] and [14, 26], as they have the deepest Y coordinate into enemy territory. Then, it will choose [14, 26] because it wants to reach the top-right edge, and [14, 26] is further to the right than [13, 26]. A Unit whos target edge is the top-left it will choose [13, 26] if the same options are available.

Note that if a Structure is destroyed at any point during the Action Phase, a path can become available to a better self-destruct location or the target edge, causing a unit's path to change dramatically, sometimes even causing it to double back.

SELF-DESTRUCT

If the Mobile unit unit's path is completely blocked, it will go to an open space closest to the opposing edge as described above and self-destruct. The self-destruct only damages enemy units and has a range of 1.5. The damage dealt to each affected enemy is equal to the starting health of the self-destructing unit. However, self-destruct damage will only occur if the unit has moved at least 5 spaces before self-destructing. Units will still attack on the frame that they self-

▼ TARGETING:

Targeting is also handled automatically by the game; players cannot directly control which enemy units their deployed Mobile units or Structures target. Mobile unit and Structure units both follow the same targeting heuristics. They begin with a list of all eligible enemy targets within range, and remove targets from the list in the following order until a unique target is identified:

1. Prioritize Mobile units over Structures
2. Choose the nearest target(s). Note that the potential targets could include multiple locations if they are the same distance away.
3. Choose the target(s) with the lowest remaining health
4. Choose the target(s) which are the furthest into/towards your side of the arena
5. Choose the target closest to an edge

This will almost always uniquely identify at most one enemy unit to target. Each unit can deal damage at most once per Frame.

ADDITIONAL CLARIFICATIONS

Units attack in the order that they were created. In the rare case that the above logic does not identify a unique unit, the most recently created unit will be chosen. Units that have 0 health remaining will not be targeted. 'Overkill' damage to a unit will not affect another unit. For example, if two one-health units are in the same location and one takes 2 damage, the extra point of damage will not affect the other unit.

▼ SHIELDING:

Factorys provide bonus health to allies that move near them as a Shield.

A Factory will attempt to apply a Shield to any friendly Mobile unit that enter its range.

A given Factory can only shield a given ally one time.

Each Factory can shield any number of unique allies.

Shields stack indefinitly, any ally can be shielded by any number of unique Factorys.

GAME REQUIREMENTS

Terminal requires the Google Chrome browser to upload, debug and test your algorithms on our website or to play by hand. Please download/update to the latest version for an optimal experience.

However, in order to run the game locally for more convenient testing, there are a few free language installation requirements:

Python 3.6 or latest

Java 10 or latest

Additionally Windows users will need to use:

Windows PowerShell v5 or latest (included in Windows 10)

Older versions of Windows, such as Windows 7, can update their PowerShell for free, simply click the link under the PS 5.1 column matching your windows version.

If you have trouble playing the game on our website or locally please check the readme in

DEVELOPMENT GUIDE

Note: When creating your algorithm, you can assume you occupy the bottom half of the arena and write code from this orientation. The game will handle the symmetry when running your algo from the top half.

SELECT YOUR LANGUAGE

Download the starter kit to start programming your bot. The kit includes starter bots for python, java, and rust.

View the programming documentation here

LANGUAGE	GITHUB REPOSITORY	DOWNLOAD
PYTHON	GITHUB	DOWNLOAD
JAVA	GITHUB	DOWNLOAD
RUST	GITHUB	DOWNLOAD

AVAILABLE TOOLS

Download tools to play games locally. Read the CLI Client Tools README included in the download for installation and usage instructions.

TOOL	RESOURCES
CLI CLIENT TOOLS	GITHUB

COMMUNITY TOOLS

Our community creates and shares a wide variety of tools for developing algos on Terminal. The Terminal projects forum is a great place to talk about any tools you are working on or perhaps even one you'd like to see!

TOOL	USER	RESOURCES
REPLAY ANALYSIS SCRIPTS	@ISSAC	THREAD, SOURCE
TERMINAL TOOLS	MAINTAINED BY @ISSAC (CREATOR @8)	THREAD, SOURCE
ALGO ANALIZER	@BCVERDICT	THREAD, SOURCE
MAP MAKER	@KEVINBAIO	THREAD, SOURCE
API TOOLKIT	@ISSAC	THREAD, SOURCE
FRAME INFO DOCUMENTATION	@8	THREAD

TROUBLESHOOTING

~~is likely someone else had a similar issue.~~

ALGO FAILED TO COMPILE:

Firstly, try clicking the algo on the My Algo page, it won't refresh its compiling status unless you click on it.

Make sure all file and folder names in your project, even inside the folder that you submit, are free of spaces and special characters.

Your algo may not appear in the dropdown list on the Playground if it is still being prepared for automated play. Try waiting a minute, or re-uploading an algo with extraneous files removed.

WEBSITE ISSUES:

Make sure you are using chrome and it is updated to the latest version. Unfortunately, only chrome is supported for this build.

If the game board has visual issues, enable hardware acceleration for chrome. It can be enabled in chrome settings by clicking here. Then simply click the toggle for **"Use hardware acceleration when available"**. Additionally users have reported **disabling hardware acceleration and reenabling** it while restarting the browser also helps.

Lastly for the above and any other kind of strange behavior found on the website, try refreshing the page. If that doesn't work try **empty cache and hard reload**; right click anywhere on the website and click **inspect**, ignore the new developer section that opens up. Instead, right click the refresh page icon on the normal url bar next to the back and forward arrow icons and hit **empty cache and hard reload**.

GAME ISSUES:

First make sure to read the README.md included in the starterkit viewable on the github page by scrolling down. Additionally make sure to get the latest version of starterkit as your issue may have been solved by a recent update. We also may have updated it with improved documentation and new features.

Below we will list some common errors and how to address them:

com.google.gson.JsonSyntaxException or **com.google.gson.Gson.fromJson**: This is caused by using the normal **print** function instead of using our provided **debug_print** function. This causes all sorts of strange behavior including the json error because stdout print statements is how your algo talks to the game engine, instead use our **debug_print** function which prints to stderr.

... has been compiled by a more recent version of the Java Runtime : This means your version of java isn't java 10 or above. Please install a newer version. You may have to restart your computer for the update to take affect. See below if this persists and you are on windows.

Error: Unable to access jarfile engine.jar or **No such file or directory**: You are likely running the run_match script while in the scripts directory. First, try running our new run_match.py script which is directory independent and more flexible in how you call it. If you do want to use an older run_match script your console has to be run in the parent folder that the engine.jar is contained in. Details for how to run the commands correctly are in the README.md.

WINDOWS OS ISSUES:

Firstly, make sure you are using powershell not cmd or git bash.

You likely will also have to run **Set-ExecutionPolicy** as detailed in the README.md.

recognized as an internal or external command...: If after installing java as described above you still get this error you likely have to update your windows PATH variable which you can see how to do here. For example, we had to add C:\Program Files\Java\jdk-11.0.1\bin to the PATH variable using the windows interface (command line changes to PATH didn't stick but doing it through control panel did).

Split-Path : Cannot bind argument to parameter 'Path' because it is null : Your powershell may be out of date which is likely if you are using a windows version before windows 10. See the requirements section for details on how to update your powershell.

The term 'py' is not recognized as the name of a cmdlet, function, script file, or operable program. : Make sure you have python 3.7 installed and not python 2 or an older version. Also make sure during installation that you set the option to add it to the PATH variable and install for all users. Lastly, after installing restart your powershell.

PERMISSION ISSUES:

If you are having issues relating to permissions, your machine may be automatically disabling execute permissions on the scripts in the scripts folder. On a Unix machine, you can use ls -la to check the permissions on your files, and chmod to add execute permissions. Similar commands exist in windows powershell. The README in the starterkit provides more information and tips related to our scripts.

GLOSSARY

	DESCRIPTION
Attack	An attack will reduce the health of a Unit equal to the attacker's damage
Demolisher	An expensive and easily destroyed Mobile unit. Has high damage and far range. Can wreak havoc on enemy defenses if properly used.
Mobile points	Currency for placing Mobile units on the board
Health (Player)	For every Mobile unit that reaches the opponent's edge, the health of the opponent is decreased by one.
Health (Unit)	Health of a Unit. If it is reduced to zero, the unit is destroyed and removed from the arena.
Interceptor	A high-health unit that deals high damage to enemy Mobile units, but cannot attack enemy Structures
Structure points	Currency for placing Structures on the board
Mobile units	Units which move across the board to the opponent's edge. Multiple Mobile units can occupy the same location.
Scout	A fast-moving unit which deals light damage.
Structure	Stationary units that do not move. They block the paths of both friendly and enemy Mobile units. No two Structures can occupy the same location.
Factory	Structure which provides additional health to friendly Mobile units
Turret	Structure which attacks enemy Mobile units
Wall	A cheap, simple Structure used to influence a Unit's path and protect more valuable Structures