

Battlecode 2020 Postmortem

Created By	Ⓜ Michael Hahn
Last Edited	@Feb 12, 2020 8:54 PM
Tags	

Battlecode 2020 Postmortem

Introduction

This was our first year competing in Battlecode as team **confused**. While we unfortunately did not qualify for the finals, we still had tons of fun and learned quite a lot through this endeavor. We also placed 2nd in the high school bracket, and ended up 6th in the scrimmage servers. You can check out our code [here](#), where our final submission was in src/rushdef.

The 2020 Game

This year, in addition to the enemy, we had to fight off climate change as well, which brought rising water levels and ever-increasing pollution. The only solution? Gathering soup. Soup was the currency of the game.

There were two types of robots: units, which could move, and buildings, which could not move.

Unit Types

- Miner: Basic economy unit, could move and mine soup, which was not added to your global soup pool unless deposited onto an adjacent refinery or HQ. Also can build all buildings except HQ.
- Landscaper: Could dig and deposit dirt on adjacent tiles. Each tile in the game had its own elevation, and digging brought down the elevation by 1, and depositing increased the elevation by 1.

- Delivery Drone: Can fly over any terrain, including water. It can pick up adjacent units (not buildings) and carry it to somewhere else and drop them in an adjacent position. Drones cannot pick up other drones.
- Cow: A neutral unit that produced a significant amount of pollution within a certain radius by farting. Moves randomly.

With the exception of drones, units could only move to tiles which had an elevation difference of within 3.

Building Types

- HQ: Builds miners, has an in-built refinery and net gun.
- Refinery: Refines soup deposited by miners, increasing pollution.
- Vaporator: Produces more passive income per turn, decreasing pollution.
- Design school: Produces landscapers.
- Fulfillment center: Produces drones.
- Net gun: Can shoot drones, killing them in one shot.

This year, there were actually very few ways units could die, and there was no direct combat between the units.

- If the tile the unit was on got flooded, then the unit dies. (with the exception of drones)
- If the building gets a certain amount of dirt on it (50 for HQ, 15 for other buildings) then it gets buried and dies.
- If the robot throws an unhandled exception or called `rc.disintegrate()`, it also dies.

Because there were so few units, we felt like all of them were very important to the game and the units were well balanced.

Win Conditions

The main goal of the game was to destroy the other HQ. In case both teams' HQ were destroyed at the same time, tiebreakers were implemented:

1. Last surviving HQ
2. Greatest number of units
3. Greatest production cost (current soup pool + unit costs)
4. Most transactions successfully minted in the blockchain
5. Highest robot ID

Other Game Mechanics

- There was a global water level, which increased very slowly in the early rounds and very fast in the later rounds. (after round 2700+)
- Except for the HQ, when robots are spawned, they have an initial action delay of 10.
- Pollution affects robots' vision range and slows down their actions.
- Landscapers could fill in flooded tiles with dirt. If the elevation went over the current global water level, then the tile gets unflooded.

Communication

Robots had local vision, and could only see a small radius around itself. To communicate, robots had to use blockchain, which allowed you to send 7 integers in a block. This was quite a lot, since it was basically 224 bits per block, and you could see up to 7 blocks per round. There was also a bidding system in place, and only the 7 most expensive bids were displayed for that round, but we felt like this feature was not well explored since message spamming cost a lot - to drown a 10 cost message, you needed to spend 77 soup just to drown that message. This made message spamming highly unprofitable until at least mid game where a team could have a bunch of vaporators. Even then, most teams preferred generating their own units rather than spending it on spamming messages, since it could also potentially block their own messages.

Funny note: Before the sprint tournament there was a conflict between the documentation and the specs, so we sometimes sent 2 integers in the blockchain, which caused an `ArrayOutOfBoundException` and crashed some other bots. We fixed this after the tournament by padding the message with zeros later on.

Pre-Sprint Tournament

Our team thought that drones would be quite powerful, so we mainly worked on drones. But first we worked on the essentials: pathing and communication. For pathing we implemented bug navigation, which worked fairly well for the initial simple maps that we were given.

For communication, we sent out an information type combined with coordinates of locations. This allowed us to communicate HQ locations, soup locations, water locations, and many other useful information. We had an information queue that all units had, and if they encountered new information, they would add it to that queue. Then on the $0 \bmod 10$ rounds, they transmitted 6 information from their queue, with the last integer being a hash for verification. This worked quite well, since it allowed us to save slightly on soup, and when units were initialized, they would only have to read blockchain on $1, 2 \bmod 10$ rounds.

We quickly realized sending all of soup and water locations were not viable, since a lot of the soups and water were grouped together in clusters, and if we had a list of all soup and water locations, then this list could easily go over 20 or 30, chugging up all of the bytecode just to iterate over the list. Therefore we only added soup locations if it was a certain distance apart from all of the current soup locations, and same for the water locations.

As for drones, we implemented sending cows over to the enemy HQ, and also drowning enemy units by picking up a nearby unit, travelling to the nearest water location and drowning them.

However, working on miners, drones, and infrastructure meant that we had zero landscaper code, which was pretty essential to survive after the whole map flooded. So we frantically hard-coded a base formation.



This formation allows us to send drones out even after everything gets flooded.

Sprint Tournament

Throughout the game, there were mainly three strategies.

Rush

Team **Battlecode** had an overwhelming rush strategy: they would make a miner run to the three possible enemy HQ locations due to symmetry, and when it found the HQ, it built a design school right next to it, which immediately spawned landscapers and buried it.



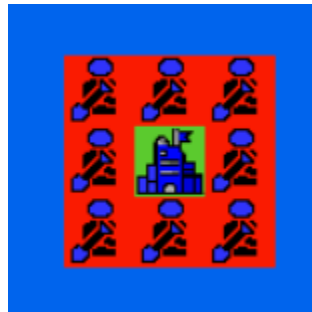
The rush successfully burying the enemy HQ

This placed them first in the scrimmage server by an overwhelming margin. While most teams were turtling with the formation of 8 landscapers surrounding HQ,

digging a wall, some of the top teams switched over to imitate **Battlegaode's** rush strategy.

Turtle

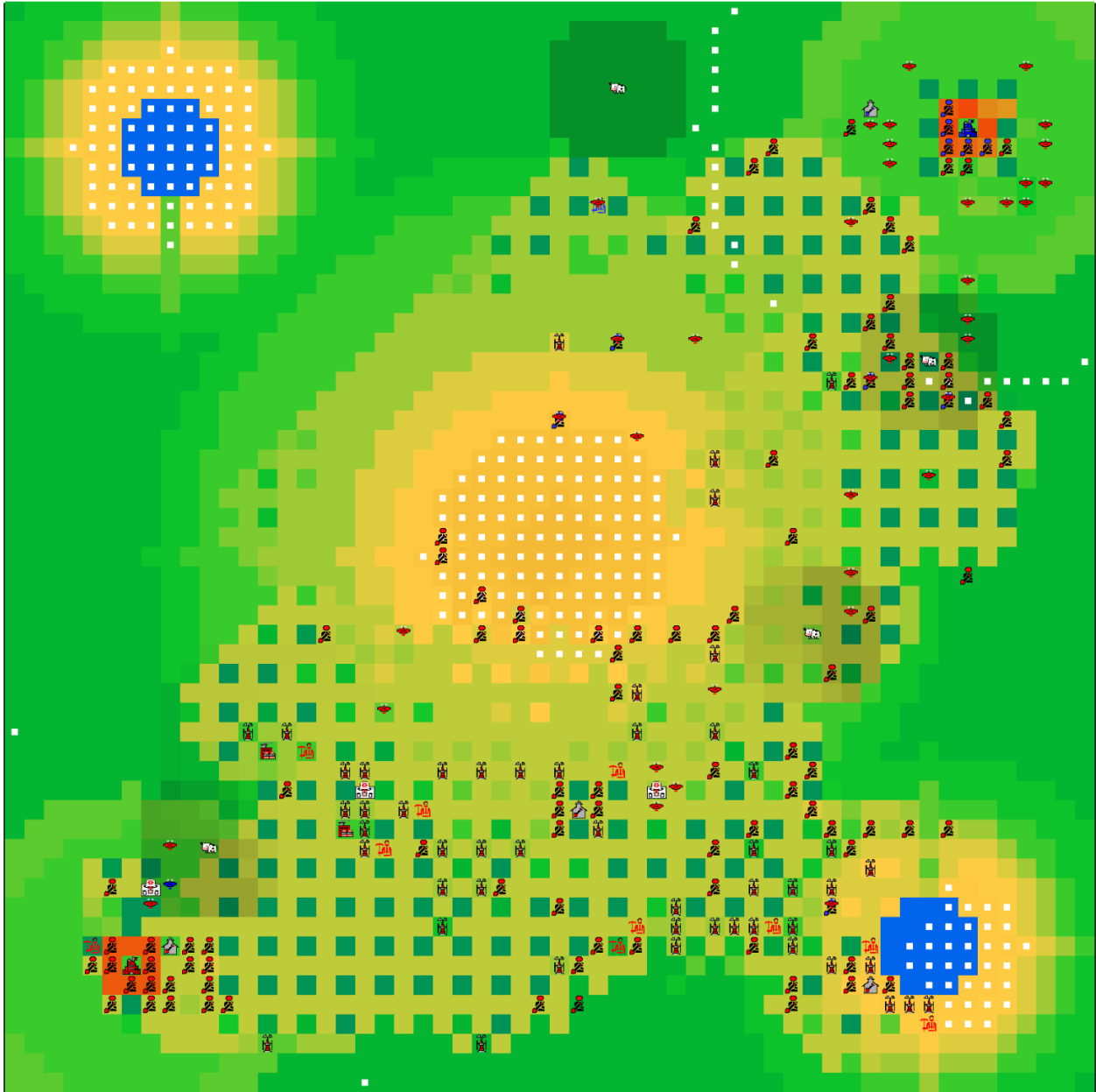
This strategy was implemented by most teams: you needed to surround your HQ with some kind of wall in order to survive when all of the land would be flooded. Most teams built a 3×3 turtle by surrounding the HQ with 8 landscapers which dug a wall.



The standard 3×3 turtle

Lattice/Waffle/Terraform

This was one of the more interesting strategies. Team **Super Cow Powers** and **Bruteforcer** spawned many landscapers to form a lattice shape, with many vaporators to sustain the massive amount of units required.



Bruteforcer (red) vs. CitricSky (blue). Bruteforcer builds a lot of vaporators to build an overwhelming number of units.

On our last submission for our sprint tournament, we submitted a pretty fatal bug that caused our base to have holes and get flooded anyways. So we got eliminated pretty early.

One thing to note was that the sprint tournament maps were much harder than the released maps - in some of the maps the HQ spawned in a corner, which

broke quite a lot of bots, and there were some maps where the HQ was surrounded completely by infinite water tiles.

In the end, **Bruteforcer** won the sprint tournament with a lattice bot, with **Battlegaode** coming in second place.

Pre-Seeding Tournament

After the sprint tournament, many people expected nerfs to rushes in some shape or form. They got a vaporator nerf instead. This caused a lot of complaints in the discord community, and combined with the fact that the top two latticing teams, **Bruteforcer** and **Super Cow Powers** were no longer eligible, rushes dominated the top pages of the scrimmage ladder.

We decided that our previous configuration of the base was not very robust; there were so many edge cases we had to deal with. So we decided to ditch the base formation and copy the terraforming strategy. (Note: we misspelled terraform as teraform initially and after that we were too lazy to change it back to the correct spelling)

Additionally, our code was getting longer and longer, with [RobotPlayer.java](#) reaching over 2000 lines of code. We were spending a lot of time scrolling through pages to find a function, so we decided to revamp our code and structure our code similar to the lectureplayer.

Rush

```
| we spend 1 hour coding a rush bot  
| top 10 within 2 hours  
| :/
```

After hearing this success story of **Anomalous Pandas**, we also added rush to our strategy. And like they said, we also got into top 10 with a few scrimmages, even though we were around 40th place before.

We assign our first spawned miner as the "rusher", which will do the following things.

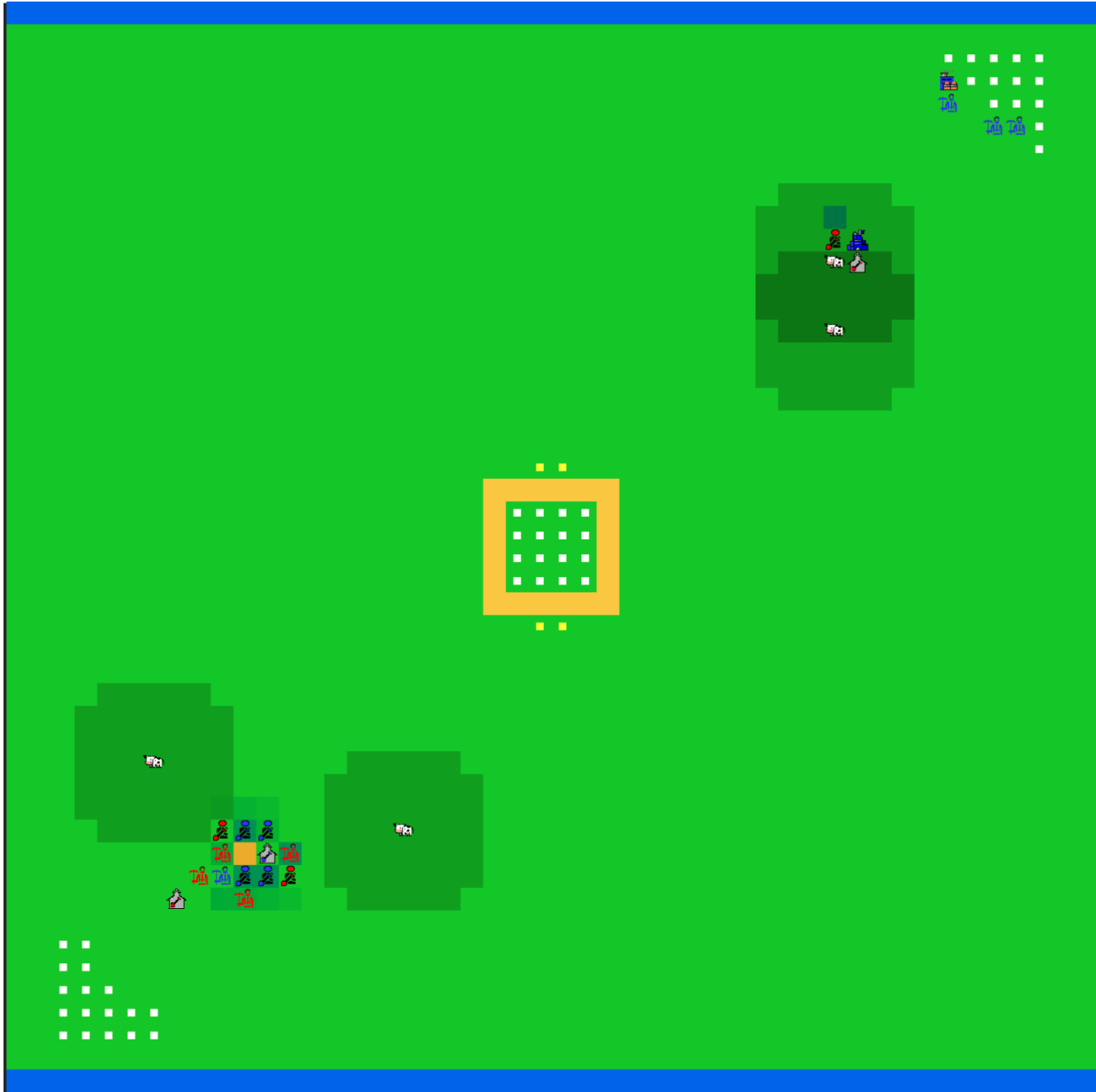
1. If it does not know the enemy HQ location, it goes to the unchecked possible locations, since there are only three possible symmetries (horizontal, vertical, rotational)
2. If it sees the HQ location, it tries to go to a position adjacent to the enemy HQ and build a design school.
3. After building the design school, it tries to go to a position adjacent to the enemy HQ but farthest away from the design school so that the design school can maximize landscapers right next to the enemy HQ.
4. If the design school has already been built and it spots a drone or a drone factory, it builds a net gun.
5. If after 180 rounds it still doesn't see the enemy HQ, it gives up on the rush and joins the rest of the miners and do normal miner stuff.

If a landscaper can see the enemy HQ when it's spawned, it goes into "rush mode", and will do the first thing in the list that it can do:

1. If it sees an empty spot next to the enemy HQ that's accessible, (elevation not too high) and it's not adjacent to the enemy HQ, then it attempts to go to the closest empty spot.
2. Depositing dirt at enemy design school
3. Depositing dirt at enemy HQ
4. Depositing dirt at enemy drone factory
5. Depositing dirt at enemy refinery
6. Digging dirt from own design school
7. Digging dirt from own net gun
8. Digging dirt from under enemy miners (so they get stuck and can't move)
9. Digging dirt from water (in some cases digging underneath itself caused it to drown by a nearby water tile)
10. Digging underneath itself
11. Navigate to nearest enemy design school
12. Navigate to nearest enemy drone factory

13. Navigate to nearest enemy refinery
14. Find the nearest enemy miner that's next to enemy HQ, navigate to it, and hope it moves so you can occupy its spot

We also noticed that in a battle of rush vs. rush, it was mostly better to not defend, since defense meant that you couldn't spend more on offense.



confused (blue) vs. Bagger288 (red). Notice how both team rushes, but we have no defense while Bagger288 tries to defend our rush. Because Bagger288 spends all their soup on defense, they can't go ham on offense and we end up burying their HQ anyways.

Because of this, we implemented a rush cost: if it's still before round 250 and the rush isn't over (our rush miner signals that the rush is over when our design school is destroyed) then everything non-rush gets an additional rush cost of 250. This meant that we would always prioritize offense over defense.

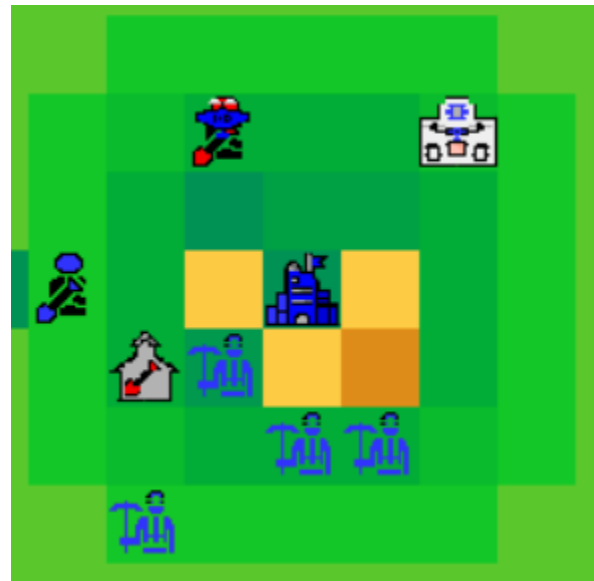
One peculiar thing is prioritizing killing the enemy design school rather than the enemy HQ. We made this choice ~~to assert dominance~~ because killing the enemy design school faster usually meant that there would be less enemy landscapers, which meant a higher chance of the rush succeeding, especially with the fact that the design school can't do anything for 10 turns after it spawns. (ok the real reason is because we were copying the rush from **Team Barcode** and we noticed them doing it and we never tested which one is better because we're lazy)

Rise of Java and rush defense

Overnight, team **Java Best Waifu** rose up and took the first place in the scrimmage ladder. What was most surprising was that they were not a rush bot like virtually every other top team, but they were a lattice bot.



confused (red) vs. Java Best Waifu (blue).
Java Best Waifu builds an early drone.



With the early drone, Java Best Waifu successfully defends our rush.

The key to their success was their early drone: they built a drone factory and drones very early in the game that allowed them to defend rushes with ease.

Seeding Tournament

We were pretty confident going into the tournament that we would get a pretty good seed. However, we lost to **CookieBot** on round 4 because our pathing

broke on all of the maps and our units couldn't do anything. Teh devs had put pretty anti-rush maps as well. Then on the losers bracket, we also met **Bagger288** who was a top rush team at the time. We got knocked out of the loser's bracket against them, so we didn't get that good of a seed as we were expecting.

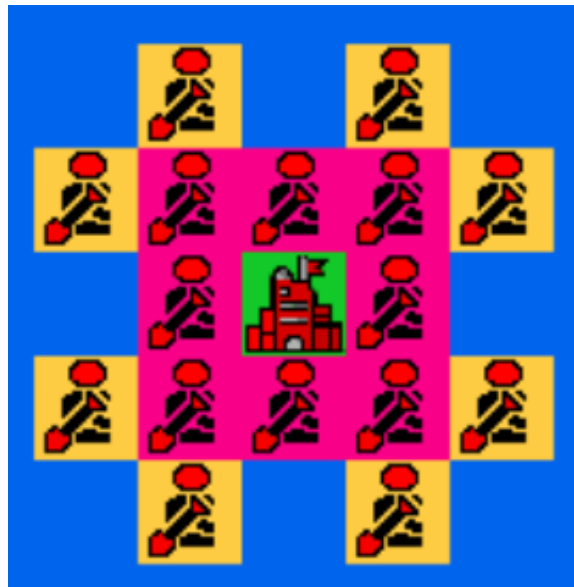
smite and **Java Best Waifu** were in the finals, and **smite** was a top rush team. While **smite** came up from the loser's bracket, they managed to beat **Java Best Waifu** two times and win due to **Java's** bot suiciding on some of the maps.

Pre-Qualifying Tournament

We were in the international pool, so we got a very short window to work on our bots after the seeding tournament. At the time, the top international teams were **Anomalous Pandas**, **Bagger288**, **NP-ez**, **Prasici**, and us.

Initially we were just planning to do small bug fixes, but we ended up doing way more and implemented a lot of additional features.

'Hashtag' turtle



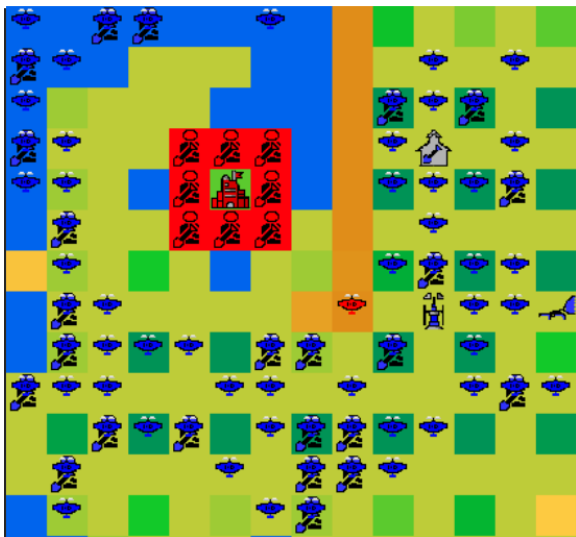
An improved version of the basic 3×3 turtle

We were losing a lot of matches simply because our turtle was getting outclassed by other turtles. So in addition to the landscapers on the 3×3 turtle, we added reinforcement landscapers at distance 5 away from the HQ. Teh devs coined it 'hashtag' turtle. These reinforcement landscapers would put the necessary amount of dirt on their own tile to survive, and mainly focus on building the inner wall.

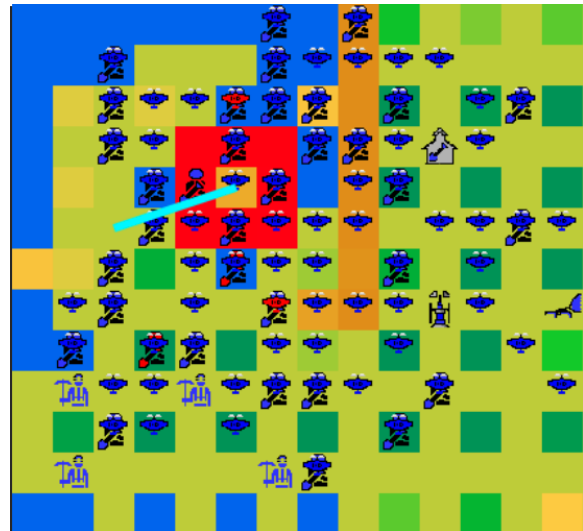
Some other turtles were greedy and also put landscapers on the corner of the hashtag (distance 8 away from HQ) but these landscapers were vulnerable to being picked off by enemy drones. Since many of the team were utilizing drone harass, we decided not to place landscapers there.

Drone crunch

A drone crunch is when you gather a large amount of drones, with the option of carrying some landscapers, and 'crunching' the enemy HQ by rushing it all at once, picking up the enemy landscapers on the wall and if you're carrying your own, placing your landscaper down so they can bury the enemy HQ.



confused (red) vs. Java Best Waifu (blue).
Java Best Waifu is preparing the drone crunch.



Java Best Waifu successfully crunches and buries our HQ.

We had also implemented this before Seeding tournament, but we didn't use it that often because we tended to turtle our HQ pretty quickly, and our eco stagnated and we couldn't get enough drones. But this is important as a followup to the next feature:

Drone wall

If the game was lattice vs. turtle, then basically the game was decided if the lattice bot didn't get rushed and was able to put up a decent number of vaporators, since they could just drone crunch. So a countermeasure was developed.



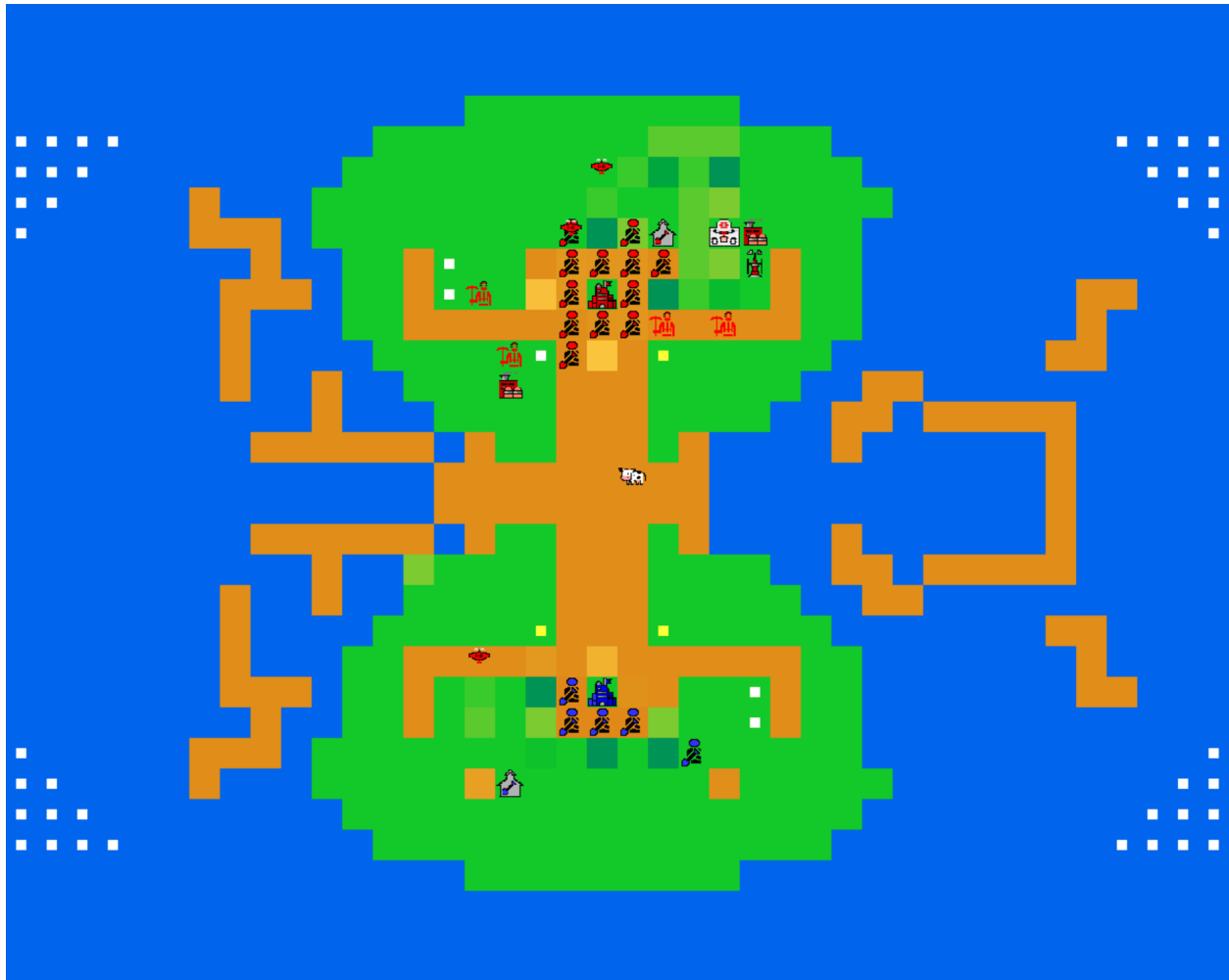
confused (blue) vs. Blue Dragon (red). Our drone wall successfully wards off red drones attempting to crunch.

Since units couldn't move onto tiles where there was another unit, if you surrounded yourself with drones, then the only way for the enemy drones to get to your HQ was to kill the drones, which required a net gun. But if the whole map was flooded, the only way to kill those drones was to either terraform the entire way and place net guns, or to drop your own landscapers filled with dirt to create mini platforms, and build net guns on them. The latter was very hard to coordinate, and only **Bowl of Chowder** successfully implemented it, and according to them, it worked very rarely.

Drone placement

While we didn't have any rush defense coded, sometimes we managed to defend rush by virtue of having enough landscapers. Then after our drones picked off the enemy units, there were many ditches and hills left around the HQ that was difficult to maneuver and left a lot of edge cases. This was another advantage of

rush: even if the rush failed, it was still very damaging to recover from the rush if it's even halfway successful. To prevent this, we used drones to pick up stray landscapers and placed them on the hashtag turtle formation. This even had the added benefit of being able to form a complete turtle around difficult terrain.

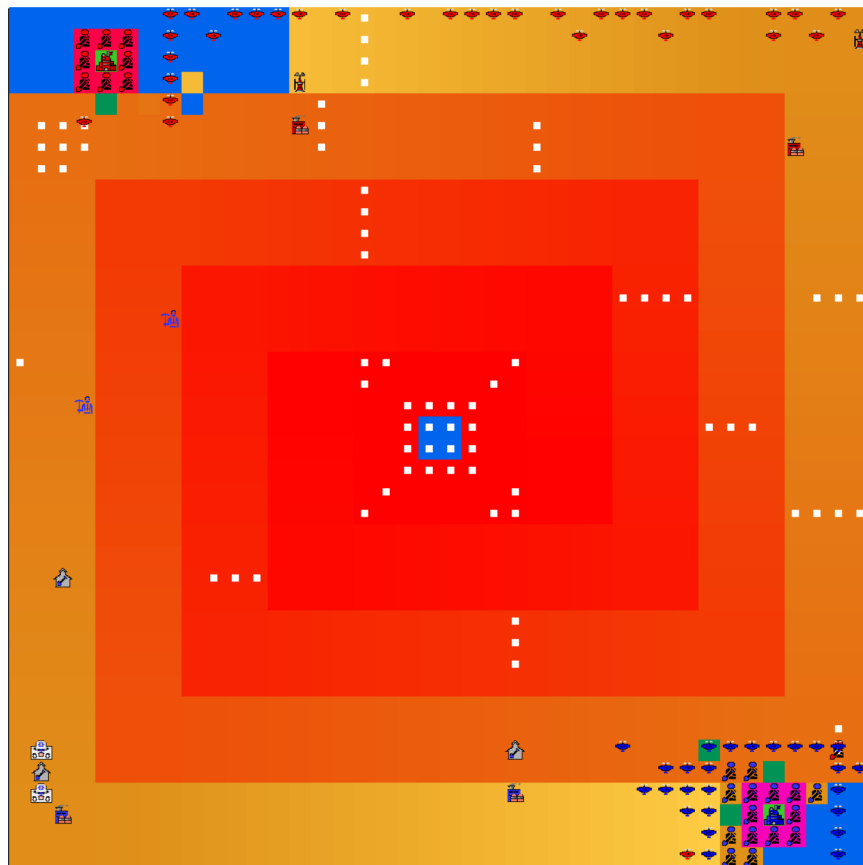


confused (red) vs. vvvvv (blue). This map is notorious for being difficult to set up a turtle because the dirt levels around the HQ is 100. Notice how we use drones to put landscapers while vvvvv has only a few landscapers.

Qualifying Tournament

Our seed was terrible; even though we were 5th seed in the international pool, our first round was against **Anomalous Pandas**, who was the first team in the scrimmage server at the time. This was because they did very badly on seeding tournament for a peculiar reason, and had 37th seed. We lost against them 4-1.

But we were still optimistic; if we managed to win all of the loser's bracket, and win against the loser of **Prasici** or **Bagger288**, we would still have been able to qualify. While we had a terrible winrate against **Bagger288**, we had a pretty good winrate against **Prasici**, hovering around ~70%. However, we got even more unlucky and faced off against **Rael Tähvend**, a team that had improved a lot between seeding tournament and qualifying tournament. They had a very greedy turtle combined with a drone wall, so it was very hard for us to win against them, since we were mainly countering lattice teams with drone crunch.



confused (red) vs. Rael Tähvend (blue). Even though we have a lot of drones, so does Rael Tähvend, and they have a drone wall so we can't crunch. They also have a greedier turtle formation.

We ended up losing 3-2 against them, and unfortunately got knocked out of the qualifying tournament.

Pre-HS Tournament

Now that we didn't qualify for the final tournament, the only tournament we had left was the high school tournament. Initially, we were the only top 10 HS team, but then another strong team rose up: **Kryptonite**. They were a hard rush team, where their rush miner would spawn a drone factory and drone if it took too long for them to get to the enemy HQ.



confused (red) vs. Kryptonite (blue). Even though this map is impossible to rush with just a miner, Kryptonite sends a miner with a drone to reach our HQ.

Because we didn't have to worry about other finalists, we decided to basically overfit onto **Kryptonite's** team. Our pathing was definitely worse than Kryptonite, but it seemed our turtling was slightly better. So we put more focus on rush defense.

Drone defense



We decided to copy **Java Best Waifu**'s rush defense strategy. We spawned a drone factory as early as possible, and made one or two drones circle right around the HQ before round 225. However, we made the grave mistake of actually testing this feature against **Kryptonite** by requesting a lot of scrimmages.

While we managed to implement this feature in time, **Kryptonite** responded by implementing building net guns even before they reached the HQ, preventing our drones from picking them up.



confused (blue) vs. Kryptonite (red). While we have defensive drones in place, their miner preemptively builds a net gun, nullifying our defense.

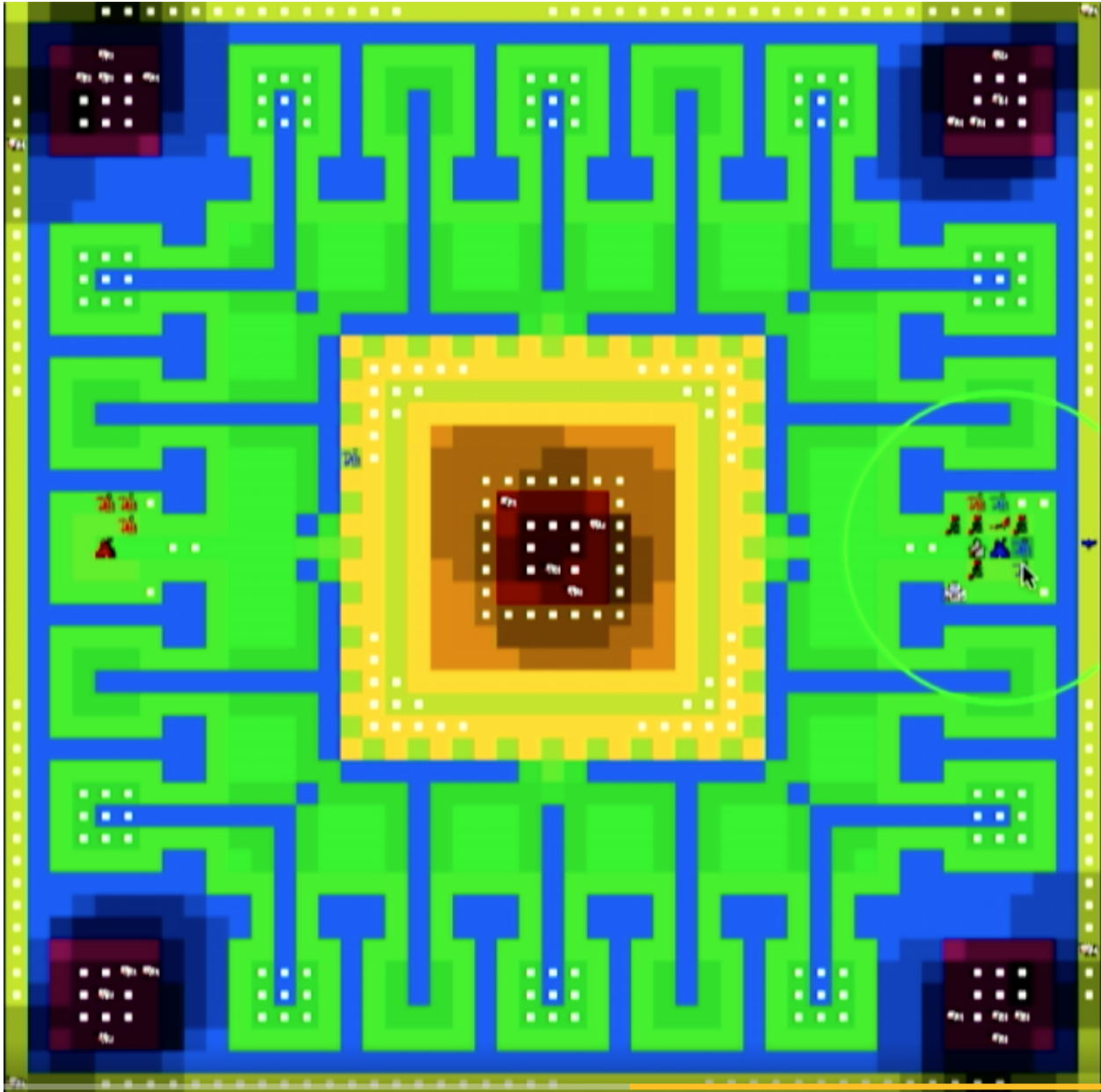
What was worse was that they submitted their change right before the deadline so we didn't have any time to respond. We learned our lesson of not requesting scrimmages to test out the feature that was supposed to be countering that team.

HS Tournament

We proceeded smoothly into the finals of the winners bracket against **Kryptonite** as expected. We lost 3-1 due to our rush pathing and **Kryptonite**'s greedy turtle. We went down to the loser's bracket, won against **blair blezers** and made top 2.

We would have to beat **Kryptonite** 2 times in a best of 5 to win the high school bracket.

Our hope were the easy rush maps: in the maps where both of us were able to rush, we were more likely to win because building a net gun was 250 while building a drone factory was 150, leaving us more advantageous in economy. In the final round, we were 2-2, and while it was an easy rush map, due to our pathing, **Kryptonite** successfully rushed while our miner was stuck.



confused (blue) vs. Kryptonite (red). Our miner is stuck while Kryptonite is already on our base.
Sorry for the bad quality of the image :/

With that, **Kryptonite** won first place in the HS tournament, and we took second place. Congrats to **Kryptonite**!

Final Tournament

While we didn't follow the final tournament meta very closely, a subcategory of the lattice strategy popped up.

Cookie

Both **smite** and **Bowl of Chowder** adapted to a version of a passive lattice instead of an aggressive lattice like **Java Best Waifu**, who terraformed all the way to the enemy's base.



smite (red) vs. Bowl of Chowder (blue). The name cookie comes from the fact that the land turns yellow and orange. Both teams lattice only up to a certain amount.

The idea was that to sustain an aggressive lattice, the wall height couldn't be very high because it was too much area to work with. Both **Java Best Waifu** and **The High Ground**, who used aggressive lattices, had a fixed dirt height of 8 for this reason. But if the area is restricted, then you can put more dirt, and with defensive net guns you can outlast the enemy's crunch, and also still have economy in the late game.

The final round was **Java Best Waifu** vs. **smite**, and **Java Best Waifu** won this year's Battlecode. Congratulations to them!

Reflection

One regret is not spending enough time on one of the most important things: pathfinding. We never really got to improving pathfinding because we relied heavily on our drones to carry units around if they were lost.



Our drones are carrying the miners who want to get to the soup on the higher elevation.

Pathfinding was one of the major causes for virtually all of our losses. Relying on drones for pathfinding had a lot of detrimental effects.

- Our initial rush miner doesn't use a drone, so it was very possible that our miner could get stuck during pathing. Our assumption was that if it was rush vs. rush, then we would also be able to rush their HQ so we shouldn't worry about defense so much. This logic broke down when our miner got lost but the enemy's miner successfully found our HQ.
- Sometimes miners got super lost in the early rounds due to troll wall placements by teh devs. This was a dire problem, especially when we didn't have any drones. This meant we wouldn't be able to do anything until our HQ got eventually flooded.

Final Thoughts

I wasn't expecting much when I initially signed up for Battlecode, but I ended up having way more fun than I ever expected. A big thank you to teh devs for organizing the competition without any major issues, and also to the Battlecode community for being friendly and super collaborative. I'm definitely going to be a part of Battlecode 2021 next year, and hopefully we perform better! I encourage everyone to try out Battlecode next year!