

# Battlecode 2020 Postmortem

Ivan Geffner (XSquare)  
Java Best Waifu

## 1 Introduction

I'm a Math PhD student from Cornell and I have been participating in Battlecode since 2015: Team Potato (13th-16th in 2015), Felix and The Buggers (12th - 9th in 2016), Oak's Disciples (3rd in 2017 and 4th in 2018) and Oak's Last Disciple (33rd-48th in 2019). This year I teamed up with Daniel Hopping, who is a CS Sophomore at University of Michigan [and also my cousin!] and competed as **Java Best Waifu**, winning First Place in the Final Tournament.

The name of the team was intended to be a [not so] lowkey suggestion to **Teh Devs**. Initially, our team name was going to be *Pls Devs, Stick With the Java Engine*, but it was too long and explicit; it later morphed into *Keep Java Engine, Battlecode is Best in Java, Java Best Engine*, and it ended up being *Java Best Waifu* as a reference to the following Reddit meme.

## 2 Game Overview

Battlecode 2020 consists of surviving longer than your opponent on a map that gets flooded little by little. The flood works as follows, the water level increases each turn and, if a tile with lower elevation than the water level is adjacent to a flooded location, it gets flooded as well. Note that this means that even low-elevation tiles may take very long to get flooded as long as they are surrounded by high-elevation ones. The water increase is slow but exponential, which guarantees that the entire map will eventually flood (and therefore that the game will eventually end). There are a total of nine different types of units that a player can control:

- **HQ:** The most important unit of the game. Each team starts with a single HQ unit, and it wins if and only if the enemy HQ gets destroyed, which means that it gets either flooded or buried by landscapers. It produces **miners** and has a large vision range, but it can't move. It can also shoot nearby **Delivery Drones** and kill them.

- **Miners:** They can gather and refine soup, which is the main (and only) resource of the game, and also build **Design Schools**, **Fulfillment Centers**, **Refineries**, **Vaporators** or **Net Guns**. They have a medium vision range and it can move to adjacent locations such that the difference in elevation is not too steep and such that they are not flooded<sup>1</sup>.
- **Design Schools:** They produce **Landscapers**. They can't move and have a small vision range.
- **Landscapers:** They can dig adjacent locations (decreasing their elevation) and deposit the dirt gathered either on other locations (increasing their elevation) or on top of buildings<sup>2</sup>, which buries them. They have a small vision range and can move to adjacent locations such that the difference in elevation is not too steep and such that they are not flooded<sup>3</sup>. Their main use is to bury enemy buildings, saving your own buildings from getting buried and to shape the terrain in such a way that water takes longer to get to your HQ.
- **Fulfillment Centers:** They produce **Delivery Drones**. They can't move and have a small vision range.
- **Delivery Drones:** They can move over any tile that doesn't contain another unit and may carry a single non-building unit and drop it wherever it wants. It is the main utility unit, since it can carry your Miners and Landscapers to places that they couldn't easily reach otherwise, and also your main offensive unit, since it can drop enemy units over water! They are slower than Miners and Landscapers, but difference in elevation and water makes them a real threat since they can get to them following a straight path. They have a small vision range.
- **Refineries:** Miners have to deposit their soup on either the HQ or on Refineries for that soup to go into the team's shared resources. They are usually built near soup deposits when they are far from the HQ or other Refineries. They can't move and have a small vision range.
- **Vaporators:** They produce a constant income of soup every turn (which is double of what you get naturally!). They are a big investment but are certainly worth if you're aiming to reach the late game. They can't move and have a small vision range.
- **Net Guns:** They can shoot nearby Delivery Drones and kill them. They can't move and have a small vision range.

Besides these units there are also neutral **Cows**. These cows create pollution that shrinks the vision range of nearby units and makes them slower and clumsier.

---

<sup>1</sup>Okay... they can, but they die! :(.

<sup>2</sup>Units that can't move.

<sup>3</sup>Same.

## 3 Bot Development and Meta

### 3.1 Release - Sprint Tournament

Since this game was pretty different from any previous game, we were very uncertain about how the meta would develop. We decided to begin with the tasks that would be absolutely necessary regardless of the strategy: soup gathering, navigation, communication, an easily modifiable building order, etc. During this time we implemented the BugNav<sup>4</sup> that we used throughout the whole tournament, also we implemented the Soup gathering and Refinery construction core methods (we would build a refinery if we didn't find any nearby one and there is at least 1000 nearby soup), and we also implemented a way to check all visible tiles using a minimal amount of bytecode instructions: instead of hardcoding the offsets of the coordinates of all visible cells into two arrays  $X$  and  $Y$ , we hardcoded the sequence of directions that a unit from  $(0,0)$  would have to move to pass through all visible cells. This saved us more than 2000 bytecodes per turn since performing a *Location.add(Direction)* instruction was only 2-3 bytecode, while doing *new Location(loc.x + X[i], loc.y + Y[i])* was more than 15. We literally didn't play a single scrim before the Sprint Tournament so we don't know exactly how the meta was developing, we later sent an incomplete bot that lost in the very first round. However, this allowed us to check the strategies of the most successful bots, which we classify into three categories:

- **Turtle bots:** This was by far the most common strategy. Turtle bots basically gather the nearby soup and then spawn eight landscapers that build a  $3 \times 3$  wall around the HQ. If you didn't do anything about it, your HQ would flood before them. (Also note that drones cannot grab any of those landscapers without entering into the HQ's shooting range!)
- **Rush bots:** Team **Battlegaode** found a way to rush the enemy base early on that was really hard to counter. They would send a miner to your HQ and then build a Design School next to it. This factory would spawn landscapers that immediately bury your HQ.
- **Lattice bots:** **Bruteforcer** and **Super Cow Powers** implemented a revolutionary strategy that consisted on building a lattice of elevated terrain and build Vaporators on top of it. It was fairly simple to build such lattice since landscapers could dig the dirt from the holes of the lattice and deposit the dirt on top of it. The lattice allowed both to reliably build Vaporators without the risk of them getting flooded early on, boosting their economy by a lot, and also to move anywhere on the map. With this strategy Bruteforcer won the Sprint Tournament for the third year in a row!

---

<sup>4</sup>See the following article.

## 3.2 Post-Sprint

After seeing the Sprint Tournament we really liked the lattice approach and we began to implement it. We made our HQ to spend the free bytecode at the end of its first turns computing where the wall should be placed. We wanted to leave enough space for at least a Design School and a Fulfillment Center, and possibly a few Vaporators, but we definitely didn't want to hardcode where the wall should be since we knew that Teh Devs were waiting to give us the trolliest map with super deep tiles close to our HQ. We ended up running a BFS to not include any deep tile inside or over the wall, and then broadcasted a message containing our HQ location followed by a sequence of bits in which the  $i$ th bit is 1 iff the  $i$ th location visible from the HQ (following a fixed order) is interior to the wall. This allowed every other unit to compute exactly the location of the wall, and more precisely it allowed every unit to classify each tile of the map in one of these five categories:

- **Interior:** Tiles that are inside the wall but not adjacent to the wall.
- **Next to Wall:** Tiles that are inside the wall and adjacent to it.
- **Wall:** Wall tiles.
- **Outer Wall:** Tiles outside the wall but such that the  $x$  or  $y$  coordinates respect to the origin are odd.
- **Holes:** Tiles outside the wall that are not "Outer Wall".

Once this was implemented, we could organize a bit where we wanted our buildings to be built: We kept track how many turns were we trying to build the same building and had the soup to do it (we called *desperation index* the number of such turns), and depending on the building, the turn, and the adjacent tile types we would either build it immediately or wait until the desperation index reached a certain threshold (for instance we tried to build our first Design School on an Interior Tile, however if our desperation index reached 8 we would build it also on Next to Wall Tiles, and if it reached 30 we would build it wherever we could.)

This tile classification also allowed us to make our Landscapers go to the closest Wall or Outer Wall tile (prioritizing Wall tiles), to dig from nearby Holes and to deposit it into Wall or Outer Wall tiles until they have height 8. This was sufficient to create a lattice on simple maps, but it failed miserably on edge cases in which water gets inside the wall early on. Our approach to solve this was to make a full list of possible cases sorted by priority:

1. Flooded Interior Tile (this includes the case in which the tile is next to a flooded tile with height similar to the water level)
2. Flooded Next to Wall Tile

3. Flooded Wall Tile
4. Wall or Outer Wall Tile with height  $< 8$
5. Next To Wall Tile
6. Interior Tile
7. Hole Tile
8. Wall with height  $> 30$
9. Wall with  $8 < \text{height} \leq 30$

Whenever a landscaper had a tile with priority 3 or less adjacent to it (or priority 4 or less if we're already on a wall/outer wall tile), it would dig from the tile with the highest priority and deposit it on the tile with lowest priority. Also, if they were not on a Wall tile, they would prioritize going to any visible flooded Interior, Next to Wall or Wall tile.

After this change, Landscapers became much more intelligent and could complete the inner wall even in maps like Hills. Thus, we reached our first milestone and almost in almost every map we would have a giant lattice full of Vaporators. It only remained to capitalize this into a Win Condition, and this was only possible with Delivery Drones (we also needed them to put our stray Miners and Landscapers on top of the Wall!). The idea is to build a huge economy and then spawn a horde of Drones and Landscapers and then crunch the enemy HQ with so many of them at the same time that even by killing one each turn it wouldn't be able to stop them: some of the drones would drown all enemy landscapers near the HQ, while other drones would put our own landscapers in adjacent positions. For this to be possible it was critical that Drones didn't die to random Net Guns (or the HQ itself) while navigating to it<sup>5</sup>, so we made each unit to broadcast each visible Net Gun that was not reported before, and also if they found that a reported Net Gun is no longer in its place they broadcast that it was destroyed. To do this efficiently, each Drone would have a two-dimensional array of the size of the map, every time a Net Gun is reported they would add +1 to all cells at distance 15 or less, and every time a Net Gun is destroyed they would add -1 to all cells at distance 15 or less to its original position. A Drone considers all locations with value  $> 0$  as obstacles<sup>6</sup>. This feature allowed Drones to circle around the enemy HQ (and nearby Net Guns), enclosing all enemy units inside and blocking Miners and Landscapers from getting out. Of course, if by doing this they found any enemy Miner and Landscaper, they would try to capture and drown them if possible :).

---

<sup>5</sup>Note that Drones have blind spots, they can move into a HQ or a Net Gun's shooting range without seeing them beforehand

<sup>6</sup>Visible Net Guns are treated differently since sometimes it happens that a Miner builds a Net Gun next to the Drone

Since this Drone harassing was so effective, we decided that our bot should start building a Fulfillment Center and a Drone, and only afterwards it would start building a Design School, Landscapers and Vaporators. Our bot would build one Landscaper and one Drone for each Vaporator built up to turn 1200, afterwards it would build 1 landscaper for every 2 drones (without taking into account the number of Vaporators). After turn 1400 each Drone would grab any visible Landscaper and attempt to path to the enemy HQ, and after 1550 they would ignore the Net Gun array, effectively crunching the enemy HQ as desired.

We uploaded this bot three days before the seeding tournament without high expectations since we literally didn't play any scrim game against any top bot (our Sprint submission was stuck way below) and we didn't know the meta. However, when we woke up the next day our bot was first in the rankings with more than 200 elo than any other bot. We found out later that apparently the meta by the time consisted almost uniquely of **rush bots** (with some **turtle bots** as well), and our bot did really good against both: Turtle bots die to the crunch and rush bots have a hard time trying to get their rush Miner close to our HQ because we started building Drones really early. By the time, the only team that did well against our bot was **Smite**, that besides building a wall they also built several drones patrolling the HQ. Thus, even if we managed to kill all their landscapers around the HQ, after we put our own landscapers they would just drown them with their drones.

### 3.3 Seeding Tournament

During the Seeding Tournament, rush and turtle strategies were still dominant with very few exceptions such as **NP-ez**, that was a mix of a lattice and a turtle bot. We realized also that our wall building was nowhere as consistent as we believed it was, and that in some maps we would drown the same turn as the examplefuncplayer and in others even before that (yes Eagles, I'm looking at you!). Fortunately for us, we never got two of such maps in the same bo3 and we never faced **Smite** until the finals. This gave us the second seed for the Final Tournament.

### 3.4 Post-Seeding

After Seeding we didn't have much time to work on our bot, so we only did minor fixes. First or all, we heavily prioritized building the inner wall before starting the lattice to avoid drowning early game. Second, we made our Landscapers to prioritize building the Lattice towards the enemy HQ if possible, and third we implemented the construction of Net Guns to both avoid being harassed by enemy drones (which was becoming more and more popular) and to destroy the enemy Drones protecting their HQ from our crunch if we manage to build the lattice all the way there. These changes really improved our winrate against teams that defended using Drones (both

by drowning our Landscapers or by obstructing your units), however other teams started to build defense mechanisms against our strategy, either by using a very similar one (**The High Ground**), or by creating a relatively small fortified lattice around their HQ (**Bowl of Chowder**, **NP-Ez**, **Smite**, **Chicken** and **Steam Locomotive**). Also, some of the rush bots like **Bagger288**, **Kryptonite**, **Battlegoade**, **Confused** and **Prasici** did a lot of improvement as well. Because of this, our last fix before the Qualifying Tournament was to create a RushManager class that would take care of the unit production if we sense we are being rushed.

If we were being rushed we did the following: We would always start building a Fulfillment Center (if we didn't have one already). If the opponent decides to build a Net Gun to counter it, they would have to spend 100 soup more than us, otherwise we can spawn Drones and drown the enemy Landscapers. After the Fulfillment Center, if there is no nearby Net Gun, we would spawn one Drone for each Landscaper that the enemy spawns, otherwise we would spawn one Landscaper for each Landscaper that the enemy has +1. These Landscapers would prioritize digging from hurt buildings and, if they are full of dirt, they would find the closest enemy building and try to bury it. We realized that most of the teams would build a Net Gun right after we spawn our first Drone, so we decided to spawn Drones only if the enemy has spawned a unit or built a building recently (this way we assume they wouldn't have the soup for a Net Gun). This strategy worked really well against most of the rush teams except for **Kryptonite**. This was mostly because our bot assumed that we could produce units at a better pace than the enemy since we are investing in economy while they sacrificed an early worker trying to rush. However, **Kryptonite** rushed relatively late<sup>7</sup>, stacked a lot of soup in between, and then spent immediately all that soup to build Net Guns and Landscapers. Our games against **Kryptonite** depended almost uniquely if our initial drone was able to repel or capture their rush miner or not.

### 3.5 Qualifying Tournament

By the time the submissions to the Qualifying Tournament closed, the teams that beat us more consistently were **The High Ground** - 50% and **Kryptonite** - 40%. However, we didn't have to face any of these teams and entered the Final Tournament after winning our first three matches.

### 3.6 Post-Qualifying

After the Qualifying Tournament we had the feeling that most of the teams kept their strategy secret. Since we didn't have much time to improve our bot, our only modifications were small bug fixes, improving the Net Gun placement, and fortifying

---

<sup>7</sup>I don't know if this was intentional or if their navigation was not very good, in any case it worked in their favor.

our wall around the HQ after round 1400. This would allow us to stack more Drones (since we usually had some Vaporators inside) and perform a second attack at round 2200.

### 3.7 Final Tournament

We didn't have much hopes for the Final Tournament, however we had a lot of fortune with how the teams were seeded: both of the teams that we performed the worst against - **The High Ground** and **Kryptonite** - were on the other side of the bracket, and we would only face **Smite** (which we had a winrate of approximately 65% - 70% against) in Semi-finals if we both got there. Also, a large portion of all rush bots was on the other side of the bracket as well. Even though we had good results against them most of the time, the outcome is much more volatile.

We won our first round against **Anomalous Pandas** and went to face **Prasici** on the second round. We were really scared about this matchup since they revealed that they would try to sabotage your communication using the messages that your bot sent on previous games. Fortunately they just copied said messages, and thus they would have to send them at exactly the same round that we send the original one to be able to confuse our bot. Probably, if instead of copying those messages they had tried to deduce the pattern and then reproduce it on the Final Tournament, they would have succeeded since we didn't actually encrypt our messages that much.

We went on to the Semi-finals and faced **Smite**, who changed their **turtle bot** to a mix of a **turtle** and **lattice** (since the lattice was small and circular some teams call that *cookie*). If they managed to complete their cookie we were doomed because there was no way of penetrating it: a full army of landscapers would keep the Net Guns alive, and we could not drown their Landscapers because of the Net Guns. However, fortunately for us (again!) our Drone harass was effective enough on the Final maps and they couldn't complete the cookie in most of the maps, or in some cases the cookie didn't have the necessary elevation to survive until our second attack at round 2200. This allowed us to get to the Winner bracket finals and face **Battlegaode**, the original **rush bot** that stuck to this strategy until the very end.

In our first two games against **Battlegaode** our early Drones succeeded on repelling their rush and we would win later with our first attack at round 1500. However, in our third game their miner successfully built a Design School next to our base, and even though we ultimately managed to bury it and to drown their Landscapers, we created very deep holes inside our wall that were filled with water. Luckily the elevation of the HQ was 5, and this gave enough time for our Landscapers to fix those holes, complete the wall and ultimately beat **Battlegaode** and advance to the Grand Finals, where we had to face **Smite** again. These last games were really clutch, winning our first game because there was an extra hole on **Smite's** Drone wall, and winning our final game because of unit count.



## 4 Thoughts About the Game

This was probably the most original and unique Battlecode edition from those I have participated, and personally it is my second favorite after 2017. All units were so different from previous editions but nevertheless the micro was balanced from the go. In my opinion, it also had the highest ceiling out of all Battlecode games that I have participated (at least this was the only game in which we still had a lot of big changes to do after the final submission deadline). A big thanks to **Teh Devs** for devising such an awesome game!

If I had to do a specs change I would probably eliminate the ability to remove dirt from buildings but I'd also increase the building resistance two times the current value (three times for HQ). Very often having the ability to "heal" units make the games end in a deadlock (for instance in 2018 before the flying mages). In the current game, if several landscapers were surrounding a net gun it was almost impossible to destroy unless you crunch it with Drones. Eliminating the ability to remove dirt would probably nerf both defensive strategies and rushes. The first ones for obvious reasons and the second ones would still be viable, however they would not be able to create stalemates next to the enemy HQ. Of course, defensive strategies - which I consider that are most likely the optimal strategy with the current specs - and rushing are valid and legitimate strategies, but this is just a personal opinion and I believe it is more exciting to see two teams go all in rather than defending/rushing :).

## 5 Suggestions for New Players

If I had to say what are the most important factors to consider when building your bot, I would say **Simplicity**, **Robustness** and **Structured Code**. In my personal experience, every time I'd try to implement a sophisticated strategy that requires a lot of coordination it would always flop since everything that can go wrong does go wrong. Usually the bots that perform the best are those that perform the basics really well (navigation, communication, micro, macro, etc.). It is especially important to use the first days to implement good and robust primitives for such basics since they'll be needed regardless of the strategy. It is usually more effective to spend your time on the basics and then figure out the optimal strategy depending on what other players do than trying to devise the optimal strategy from the go.

Also, if the code undergoes drastic changes (for instance because you change the bot's main strategy) I would suggest to do a bot from scratch. It is usually faster than expected and it is way better on the long run.

## 6 Final Thoughts

This year's competition has been amazing overall. The game was really good, there were almost no bugs (except maybe the Client leaks, but they were not too annoying), and it was balanced from the go. I'm looking forward to see how Battlecode 2021 is going to be. Also, the Final Tournament weekend at MIT was really enjoyable, especially because the organization was flawless and the Battlecode community is awesome. A big shoutout to **Steam Locomotive**, **The High Ground**, **Blue Dragon** and all other teams that we played board games and Smash on Friday and Saturday nights. For future editions **Teh Devs** should make Terraforming Mars the official Battlecode board game :).