

Battlecode 2019 Post-Mortem

Ivan Geffner [XSquare]

After five years participating in Battlecode - **Team Potato** [2015, 12th-16th], **Felix and The Buggers** [2016, 9th-11th], **Oak's Disciples** [2017, 3rd; 2018, 4th], **Oak's Last Disciple** [2019, 33th-48th] - I've decided to do a Post-Mortem for Battlecode, especially since this year we got eliminated pretty early on compared to previous editions.

I'll write here my thoughts about the game, its mechanics, what we did and what we could have done better. I hope this discussion helps other teams that are willing to compete and improve in further Battlecode editions. Our code can be found at the following link.

1 The Game: 2019

This year's game was very similar to classic RTS games such as Age of Empires II or StarCraft. Each team starts with between 1 and 3 castles, and the winner is the team that destroys all enemy castles. If after 1000 turns no team has achieved this objective, the winner is decided in order by

- The team who has more castles remaining.
- The team who has the most total HP between all units.
- Random coin toss.

1.1 Economy

This year there were two types of resources: Karbonite, which is used to build units, and Fuel, which is used for almost every single action in the game: moving, attacking, communicating... and also for building units as well. These resources were scattered around the maps in special tiles called *depots*. These depots were infinite, although only one pilgrim could be mining from it at a time. Whenever a pilgrim mines a resource, a small amount of that resource gets added to its personal deposit, but not to the team stash. To add these resources to the shared stash, the pilgrim must deposit them into a Castle or a Church.

Besides depots, each team earned a considerable amount of fuel each turn for free. Also, when a unit kills another unit, the killer gets half of the karbonite cost of the victim into its personal stash.

1.2 Units

- **Castle:** Initial unit of the game and cannot be built by other units. It can build all other units of the game except churches.
- **Pilgrim:** It can't fight but it can mine the resource depots and build Churches.
- **Church:** Puts its resources into the team shared stash. It can also build all other units except Castles.
- **Crusader:** Fast short-ranged unit with medium health, medium vision range, low and cheap attack (fuel-wise).
- **Preacher:** Slow short-ranged unit with high health, short vision range, high attack with medium fuel cost. Its attacks hit a 3x3 area around the target.
- **Prophet:** Slow long-ranged unit with low health and high vision range, low and expensive attack.

1.3 BC19 main quirks

- Initial map is given to all units beforehand (as in 2018).
- Units are controlled independently and they don't have shared vision (as in 2016 and 2017).
- Units cannot move and attack in the same turn.
- There was enough time to do several computations involving the whole map each turn (for instance, BFS).
- Units could send messages to the castles for free, but to communicate with other units they would have to broadcast messages, which cost fuel proportionally to the range of the broadcast.
- Units cannot access the global turn count. They can only know how many turns they've been alive.

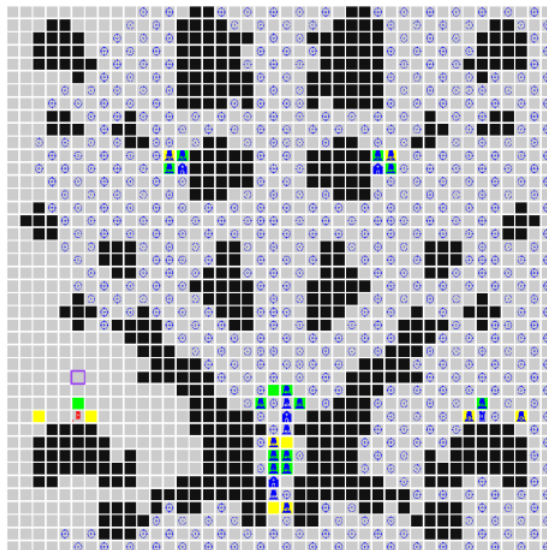
2 Oak's Last Disciple 2019 run

2.1 Pre-Sprint Tournament

Until sprint tournament, the fact that the Castles had low HP and no attack made rush the go-to strategy since investing into pilgrims didn't pay off soon enough to counter a turn-1 rush; in fact, most of the discussions on the discord server were about how to perform the optimal rush. As days passed by, teams tried to counter rush by turtling and either counterattacking or winning by total HP. In the end, the best bots built a single worker that kept mining karbonite next to the castle (it was guaranteed that there was going to be at least a karbonite mine nearby), and used the remaining resources and the free fuel each turn to attack/turtle.

While this rush/turtle race was going on, we invested our time in building an "Eco Bot". Basically, it would optimize where to send each pilgrim - for instance, if we needed a pilgrim at a certain spot, only the closest castle would build it - and pilgrims would request to save some resources to build a church nearby if they detected they were far from other castles or churches.

Also, castles and churches would build prophets if they detected enemy units nearby or if they were rich enough, and these prophets were arranged in a chessboard lattice fashion. Note that the prophets had the same attack and vision range, and they were higher than all other combat units. Therefore, if our bot managed to get a big enough lattice it would be really difficult to penetrate since the opponent's combat units would get killed even before seeing the lattice. Although we knew we would get totally destroyed by rush bots - no time to build a lattice before we get destroyed! - we heavily suspected that **Teh Devs** didn't want people to ignore the economy and would soon nerf rushes.



Prophet Lattice

2.2 Sprint Tournament

In the Sprint Tournament, our bot got kicked out pretty early (as expected) by **CitricSky** and we didn't even see it play on stream. All top bots in this tournament were rush bots, with **Monad** being the winner.

2.3 Post-Sprint, Pre-Seeding

Soon after Sprint, **Teh Devs** changed specs, made Castles and Churches stronger, and gave Castles a long range attack, which was quite a nerf on rushes. After this patch we modified our bot to continue with its strategy but saving a bit of karbonite and fuel for helping the castle resist rush attacks. With the new castle health and attack, saving a bit of resources could stop all turn-1 rushes and gave us the edge later on. Literally, our bot went from F-tier to S-tier in 1 day.

However, as days passed by, we couldn't improve our bot much except by choosing better where to send our pilgrims and building better lattices. Since this strategy was not really hard to implement, soon a lot of teams began catching up and apparently this strategy became the meta for the Seeding Tournament. In short, most of the games consisted of both teams sending pilgrims to different resource clusters, and the team that secured the most clusters ended up winning the game.

2.4 Seeding Tournament

By the time the Seeding Tournament was ran, fortunately for us we still had an edge over most of the other bots since our pilgrim distribution was pretty efficient, and we got 3rd place after losing to **Standard Technology** - a similar bot to ours - and to **DOS**, which surprisingly got to 2nd spot without riding the meta and doing a mix between economy and a continuous preacher rush. Even though by saving some resources you could repeal each of the preacher waves, the preacher attacks did damage in a 3x3 area, hurt your castle little by little and ended up destroying it.

2.5 Post-Seeding

Soon after seeding, teams started to figure out ways to beat eco bots. The most obvious one was still being an eco bot, but *greedier*: instead of saving resources for possible early attacks, you spend all your resources (or at least, more resources than your opponent) into pilgrim/church expansion and cluster securing. The other one was to send some troops early game to clusters you expect your opponent to send pilgrims to (especially clusters or resources located close to the middle of the map). This way, the pilgrims cannot approach the cluster and you ensure to secure it later on. I call these *cheesy* eco bots.

The main problem with greedy bots is that they die to rushes, while cheesy bots have the following:

- They lose against *cheesier* bots. If the opponent sends more troops than you, you lose.
- However, by sending more combat units you might either die to rush or you might actually be outfarmed by greedy bots that didn't invest into early combat units (early combat units slow the economy a lot).
- There are counter-measures eco bots can take against cheesy bots. If a Pilgrim detects cheesing early on - for instance, if it sees a solitary prophet in a resource cluster - it can build a church in the prophet's range, the church can build a prophet that same turn, and because of how the turn queue worked, your prophet gets to attack first. This way you win the cluster while getting a church relatively close to it.
- When facing other cheesy bots, it is literally a coin toss who gets the contested clusters. In Battlecode, experience has told me that consistency is really valuable.

To summarize, the meta post-seeding was the following:

Rush > Greedier/Cheesier bots > Greedy/Cheesy bots > Eco bots > Rush

Since we had our bot practically coded since before Sprint, we had time to test all these possibilities, and we came to the conclusion that our pre-Sprint eco bot was the most consistent one, so we ended submitting it for the final tournament with a couple of improvements:

- Castles and Churches signal where are the closest enemy units: This way our prophets do not go into range of other prophets and our preachers could advance without getting into enemy range, taking into account that enemies could have moved after the signal. This way preacher rushes could be dealt with without them hurting our castle. We tested this against a bot that simulated **DOS** strategy with very good results. We also tried against a bot that would increase the size of every wave with equally good results.
- We incorporated **NP-cgw**'s amazing attack protocol¹ into our bot. After a certain turn, if we're down on castle count or if we have fuel enough, the castles give the signal to attack. However, when the castle gives the signal, each prophet saves the local turn t^{signal} in which it received the signal, and does not advance if the distance d that the prophet would be respect to its target satisfies $d < \left(C - \frac{t^{current} - t^{signal}}{k}\right)^2$, where we set $C = 60, k = 3$. This way, all prophets would always try to be at the same distance from the enemy castle - which shrinks over time - and would all engage at the same time. We checked that although it was not efficient against full-prophet lattices (we still lost slightly more attacking units than the lattice defensive units), it was really efficient against bots that invested into attacking or bots that were greedy with the HP tie-break and mixed prophets with other short-ranged units in their lattice.

¹To be honest I don't know how was theirs, but we could simulate it fairly well.

2.6 Final Tournament Qualifying

The meta for the final tournament was kind of a mix of all strategies described above, along with some other *Miscellaneous* strategies such as **Team Barcode**'s wall or **Knights of Cowmelot**'s church chains (basically abusing the quirks of the turn queue to infiltrate lattices by building consecutive churches). Nevertheless, in my opinion, these strategies were much less consistent than the ones previously described.

In our case, we *greatly* underestimated the amount of greedy and cheesy eco bots around. In fact, we didn't even implement the counter-measures we thought against cheesing since we believed that almost no bots would be doing it. Our assumption that the eco bot was the most consistent was assuming that there would be a relatively equal proportion of the strategies described. However, at the final tournament, there were almost no rush bots left and the more conservative bots were heavily punished by not fully investing into expansion.

Although we did a good start after a narrow win against *Bye21* [chill, it's a joke :)], we lost our first real match against **69 Tons of Data**, which was a way greedier Eco bot. After surviving a few rounds in the Losers Bracket we were definitely kicked out after losing to **Wololo**, which starts as a cheesy eco bot and when it has enough resources it starts sending waves of combat units (to be honest, because of the matchup I knew we were losing this game before playing it D:).

2.7 Final Tournament

This has not happened yet at the time I'm writing this. Hopefully another team can fill this up! :)

3 Final Thoughts

Although we were disappointed for not making it to the finals, we did have a good time coming up with ideas for our bot and chatting with fellow Battlecode participants in Discord. Right after we got eliminated one of **Teh Devs** said on stream that "A lot of people will be disappointed since Oak's was a fan-favorite team". We also love you, Battlecode community ♡.

Also, If I had to say which was our biggest mistake this year it would probably be that we barely scrimmed after seeding (games took too long and we were really lazy about it), and we didn't see the meta shift that was happening right in front of our eyes. If we participate in further editions we'll take this into account and try to be really active in the scrim server.

Good luck in the Finals for those who go², and I hope to see you all next year!

²To be honest, since you compete against each other you cannot all have *good luck*!