

# Orion - Starcraft 2 AI Bot

*Anastasia Borissova Asma Omar Joe Ha*

## **Motivation & Prior Work**

Before starting CMPUT 350 none of us had any experience developing video games and we also never played StarCraft II. The only experience we had playing real-time strategy games was from playing video and board games similar to StarCraft II. The only other experience that we had that is relevant to game development was that one of us took CMPUT 355 at the same time that covered basic gaming algorithms.

The main motivation for this project was to learn more about the game and the development itself. We were very interested in how different strategies are different from each other and how they compare when played against AI. So the final goal for this project is to create a bot that can beat hard AI most of the time.

## **Our Approach**

Since none of us were very familiar with game development or the StarCraft game itself, we decided to take the following approach. Our first step of the development started with extensive research, where we looked at different strategies and tried to play them through ourselves. After deciding on what should be our next step in development, we implement it and run some tests. Finally, after looking at the results we evaluate our strategy and try to improve upon it. After that we repeat the cycle until we are satisfied with our end result.

After playing the game and doing some research we decided that our main focus would be on the rush strategies, because Starcraft game has many components to it and it is hard to plan very far in the future. We chose the following 3 strategies:

- 12 Marines
- 6Rax
- 1v1v1

For our project we wanted to choose some rush strategies that were good for our class Terran and worked well against all other classes. We also wanted strategies that are quite different from each other and cover different aspects of the game. 3 strategies that we chose go from very risky all-in strategies to the ones that are more conservative and allow us to continue playing even if the rush attack doesn't work out at first.

In our development we wanted to implement all of these 3 strategies and later evaluate them to see if we can combine the best parts of them in the final strategy.

## **12 Marines**

**Link:** [@4:30](https://liquipedia.net/starcraft2/12_Marines)

12 Marines was our most risky all-in strategy. The only thing that we actually need to build for it is 1 Supply Depot and 3 Barracks. We also upgrade to orbital command to do Calldown: Extra Supplies and Calldown: MULE.

The main goal of this strategy is to create as many marines as possible and at around 4.5 minutes after the start of the game use all of our units including SCVs to attack the enemy base. In this strategy SCVs act as tanks and help attack the enemy.

Basic Build Order:

- Supply Depot (1)
- Barracks (1 - 3)
- Cut SCV production, produce Marines
- Orbital Command Upgrade
  - Calldown: Extra Supplies & Calldown: MULE

Overall, even though this strategy was easy to implement and it was quick, we found that 12 Marines to be too risky for our team. If the rush attack does not work there is a very low chance of coming back from it and winning later in the game. On top of that it was not 100% successful with all classes.

12 Marines Results:

| 12 Marines | Difficulty |        |       |
|------------|------------|--------|-------|
| Enemy      | Easy       | Medium | Hard  |
| Zerg       | 20/20      | 18/20  | 2/20  |
| Protoss    | 20/20      | 20/20  | 15/20 |
| Terran     | 20/20      | 19/20  | 11/20 |

**6 Rax**

**Link:** [https://liquipedia.net/starcraft2/6Rax\\_Allin](https://liquipedia.net/starcraft2/6Rax_Allin)

This is a slightly modified version of the original 6Rax strategy. The original version specified building a specific number of barracks, supply depots and marines. After executing the strategy, it wasn't the most efficient hence the additional modifications to improve it. We've added base expansion, expanding as much as we can to the other corners to distract the enemy at the beginning of the game and for a longer game. With the base expansions, comes extra barracks, extra supply depots and extra marines.

Basic Build Order:

- Supply Depot(1) @ choke point
- Barracks (1) @ choke point
- Produce as many Marines as we could afford to
- Orbital Command Upgrade

- Calldown: Extra Supplies & Calldown: MULE
- Base expansion ~ build as much as possible of all units/structures

Overall, this strategy did best against Zerg compared to the other races. The early attack, early expansion and the building at the choke point for defense were advantageous. The main disadvantage of this strategy is that it is really poor in late game performance.

#### 6Rax Results:

| 6 Rax   | Difficulty |        |       |
|---------|------------|--------|-------|
| Enemy   | Easy       | Medium | Hard  |
| Zerg    | 20/20      | 18/20  | 12/20 |
| Protoss | 20/20      | 18/20  | 6/20  |
| Terran  | 20/20      | 18/20  | 8/20  |

#### **1v1v1**

**Link:** [https://liquipedia.net/starcraft2/1/1/1\\_Build/Banshee\\_into\\_Siege\\_Tech\\_All-in](https://liquipedia.net/starcraft2/1/1/1_Build/Banshee_into_Siege_Tech_All-in)

The main objective of 1v1v1 is to attack around the 10 minute mark with masses of marines and siege tanks. It is different from the other two strategies we examined, as it adds hellions, banshees, and siege tanks to the army.

#### Basic Build Order:

- Supply Depots
- Refineries
- Barracks w/ Reactor
- Orbital Command Upgrade
  - Calldown: Calldown: MULE
- Starport w/ Techlab
- Factory w/ Techlab

The main advantage of this strategy is that the siege tanks along with banshees can overwhelm the enemy, even if they begin to produce stronger units. However, the biggest disadvantage was that there is not enough gas for both banshees and siege tanks on one base. After running it against the built in AI, we found that it was most effective against Terran and weakest against Zerg.

### 1v1v1 Results:

| 1v1v1   | Difficulty |        |       |
|---------|------------|--------|-------|
| Enemy   | Easy       | Medium | Hard  |
| Zerg    | 20/20      | 13/20  | 3/20  |
| Protoss | 20/20      | 18/20  | 7/20  |
| Terran  | 20/20      | 19/20  | 11/20 |

## **Specific implementations**

In this section we would talk about different aspects of our strategy in more detail.

### **Motivating A Combined Strategy**

After running both strategies, we found that the 1v1v1 and 6Rax were the most dominant with the right amount of risk. We choose not to use 12 Marines, since sending all the SCVs to attack would risk an immediate loss if not successful against a strong enemy. The 1v1v1 strategy was weak against the zerg race due to quick expansion of spawning pools, however we compensated this by using 6Rax which was most effective against zerg. We decided to combine the best elements of each to create a stronger strategy as shown below.

Initially we planned to build a wall at the choke point (barracks and supply depot), however we found that tanks could sometimes get stuck causing a pile up. Therefore, we ignored this part of the 6Rax strategy. Since none of the previous three approaches covered expansion, we decided to add it in our final strategy. Once the build order was completed for the first base, we expanded immediately to the other corners of the map and the surrounding area near the original base. This allowed for quick resource collection to add more army units and distract the enemy from the original base.

### **Build Order**

#### Structure Build:

- Supply depots and refineries as needed.
- Orbital Command Upgrade (Call down MULE and Extra Supplies).
- Barracks > Reactor (2)
- Factory > Techlab (2)
- Starport > Techlab (1)
- Engineering Bay (1)
- Missile Turrets (2)

We followed this basic sequence for our build order. Once this sequence was completed, we began base expansion to the surrounding area and the other possible corners.

### Army Build:

Marines → Hellions (base defence) → Banshees → Siege Tanks → Medivacs → Reapers (base defence)

In following a similar army build to 1v1v1, we found it most successful against early rushes from enemies.

### **Defence**

Since rush strategies are known for having poor defence (due to lack of prioritization for the economy), we decided to:

- Defend the base before attacking with all army units at the second command centre expansion location.
- Medivacs to heal units defending base before attacking.
- Use hellions and reapers as defence units for base.
- Add missile turrets to handle air enemy attacks.

### **Scouting**

Before attacking we needed to figure out the location of the enemy base. We decided to accomplish this with scouting. Our first approach for scouting was a bit different from the final one.

#### Initial approach:

At the beginning of the game we sent 3 marines in 3 different corners where the enemy base could be located. While SCVs are scouting we make sure to capture coordinates of any enemy buildings, structures or units and to which corner it is the closest. When it was time to attack we looked at our results and chose the corner that had the highest number of enemy units or structures noticed close to it.

This approach worked most of the time, however, in the situation where the enemy did a rush attack before us, we would see many enemies near our base. This made our units go sometimes to the wrong corner.

#### Final approach:

In the final approach we still sent out our SCVs to 3 different corners, but now every single time we see enemy buildings, structures or units we add their location to the queue. And when we are ready to attack we look at the front of the queue for the location and send all of our units to it. This approach was simpler and worked better, because we start scouting very early in the game and usually the only enemy we see is located in its base.

### **Attacking & Handling End Game Condition**

The army build order is complete as soon as we have a minimum of six siege tanks (any less and we cannot overwhelm enemy defence). Our bot is sent to the first attack location, as soon as all build order for the army is completed. Once we reach our first location of attack, we attack with all of our units until one of them becomes idle. If the unit is both idle and is at the location of attack, we pop the next location from our queue of locations where

we saw the enemy and send all of our units to that location to attack. We continue this approach until there are no more locations to attack. Usually this should be able to destroy all of the enemy structures. If for some reason the game is not yet complete, we send all of our units to each of the possible expansion locations to check if there is an enemy on that base.

## **Final Results**

| Final   | Difficulty |        |       |
|---------|------------|--------|-------|
| Enemy   | Easy       | Medium | Hard  |
| Zerg    | 20/20      | 20/20  | 13/20 |
| Protoss | 20/20      | 20/20  | 15/20 |
| Terran  | 20/20      | 20/20  | 11/20 |

For the final strategy, we noticed consistent performance against races as opposed to before. Previously, certain races could be detrimental to one of the rush strategies and ultimately lead the bot to losing.

### Pros:

- Able to quickly attack the enemy before stronger enemy units are created.
- Using medivacs to heal units before attacking allowed the strategy to proceed quicker, as we would not need to wait for new units to be created.
- Expansion with multiple orbital commands allowed for MULES to collect minerals faster.
- The additional base expansion in the other corners distracted the units from the main base.
- Having a mixture of ground and flying units was advantageous against the enemy as it was well rounded.
- Sending units to attack at once in a group (stronger together).

### Cons:

- Poor late game performance: if the game progresses our bot can no longer handle strong defence and strong army units from the enemy.
- Not enough vespene gas to produce banshees, siege tanks and medivacs at once.
- Weak defence (due to taking this rush strategy approach).
- Deadlock due to placement of structures: for our current build order structure, we use a series of switch statements to proceed to the next stage (since it is important for certain structures to be built first). In certain scenarios, where a build location cannot be found or when an enemy destroys a structure required to proceed to the

next stage, deadlock occurred. To mitigate this by refining our TryBuildStructure(), to only build if it is not in a radius of another structure.

## **Challenges**

Prior to taking CMPUT 350, none of us had ever played Starcraft 2 before. Due to our unfamiliarity, it was challenging to get started with the project. We spent a lot of time figuring out how to get started. To overcome this, we ended up watching a few tutorials and played the game a few times using the strategies we looked into before starting to implement.

The Starcraft 2 API was challenging to understand. We spent quite some time trying to adopt some of the functionalities at the beginning, we talked to the TA and he walked us through some of the basics on setting up the project.

We encountered a “potential” bug in the API. When calling functions on unit idle for the orbital command, we were unable to call down mules and call down extra supplies in the same code. When we called only extra supplies it worked, and when we called down mules it worked but not both. Extra supplies were to be called only the first time when the orbital command was idle, every other time it was supposed to call down mules if the resources were available, but this never worked. It affected some of our functionality since one of the strategies we looked into specified the functions to be called in that order. We made sure all the resources needed to perform the function calls were available but it never worked.

## **Future work**

We are planning to look into other unit and structure types that we haven’t used to see if they offer any additional benefits.

We need to adopt a better worker management system. For now, we are just using any idle worker or random worker to assign some task. It would be a lot more beneficial if we assign tasks accordingly, for instance, when expanding to the other corners, scvs are travelling back and forth between the opposite corners to mine minerals. It would be better if we have scvs assigned to bases and move and only for attack or if one base is destroyed.

We are planning to also integrate the scanner sweep into our code, we did implement it but we ran out of time, so will make use of it in the future. The scanner sweep will result in a lot more accurate scouting than when sending scvs to scout. We could also make use of the extra scvs instead of sending them away to scout.

We need to keep the code a little bit more organized and cleaner by using priority queue to keep track of build orders instead of switch statements.