**StarCraft 2**

**Report**

StarCraft 2 is a game of strategy where players compete against each other or against the computer. Each player starts with a ‘base’, which allows each player to construct basic units that collect resources. Moreover, players can build more buildings that unlock new units such as combat units, and then they can upgrade these units to get better damage, health and defense points. The main objective is to gather an army and destroy all of their base. In StarCraft 2 there are 3 races (Terran, Protoss, and Zerg). The Protoss are a robotic-based race, which we used to play. This race is considered as our agent, each race has different abilities and building units making our strategy depend on the choice of agent.



As shown in the tree-diagram, you cannot build something unless you’ve built the level above it. In order to produce a "High Templar," first each player must have a Nexus then build a Gateway, then build a Cybernetics Core, then a Twilight Council, then a Templar Archives. Buildings, workers, and fighters all cost you resources and time.

The ZIP file we downloaded contained all we needed for us to manipulate the game as we want, it contained different maps, all classes of objects in the game as well as the required methods and packages. Our goal is to use these methods based on our strategy that will help us win all our games.

To start off, we import the packages we downloaded from the ZIP file, then we import the maps we’ll use, the race we’ll play with and the difficulty as well as the race we’ll play against.

In the game, each player needs the largest army to defeat his opponent. In order to do that, first resources (minerals and gas) need to be collected. For minerals, we just need to mine them with workers and as the game starts each player has 12 workers for the Protoss race workers are called Probes.

We used the method distribute\_workers( ) to evenly distribute the workers so that we are not wasting them and using them optimally, so 3 workers per patch of minerals is the most optimal way.

As the game moves on, we build more builders (probes) however each Nexus (where probes are built) can only produce a fixed amount of probes, so we need to build a new Nexus. This allows us to get more resources so that we can build more expensive units. After some time, we will reach our maximum capacity, then we need to increase our supply of Pylons (our maximum capacity of units).

The next step is to expand, so that we can have more units and resources. To do that we build a new nexus, preferably not so close to our first nexus, so our method builds a nexus as time goes by, within a range of 15. The method expand( ) is constantly working with time, which means we are constantly expanding, increasing our capacitance. With each Nexus, several Pylons are built to supply us with power as well as more workers to gather the resources.

To harvest gas we need to search for nearby Geysers then build assimilators on them, the gas is used in building of some units. An important note is that for all the previous methods, we must first check if we have sufficient resources to build the unit, then check if we actually need it, then finally check if another one is already being built.

By now we have everything ready, our resources are increasing, our base is expanding and our workers are growing in numbers. After many trials, we found out that our most preferable strategy for attacking, was actually through defending. When we start the game, we build a forge, and for each nexus we build 10 Photocannons, which are very powerful stationary defending units. This buys us time to gather as much resources as possible, then after each Nexus has 10 Photocannons, we start building our attacking force. For each Nexus we should have, 3 Photocannons, 10 Voidrays and 5 Immortals. After reaching a total number of FULLY UPGRADED units of 30, we start attacking. This strategy insures that during the beginning it will be very hard for opponents to attack our base, as Photocannons are very powerful, and during the short period at the start it will be hard to put together an army strong enough to take them down. During that time we build a fully upgraded army that is supposed to be strong enough to take down the enemy’s base.

We came to this conclusion after experimenting with different units, Zealots and Voidrays were not strong enough to defend our base and it was constantly getting destroyed. Also almost immediately attacking the enemy’s base was not successful as our attacking units were very few and weak. We tried wating some time to create an army of Immortals, Zealots and Voidrays then attack, however the enemy’s defense was still too strong for them. After experimenting with different units, we found out how powerful the Photocannons were, so we decided that we’d build many of them at the start to ensure our base was safe, this bought us a lot of time to gather enough resources to build a fierce army that was hard to stop. We are aware that would also give the enemy time to build a strong defense, but fortunately we would have the advantage of being a step ahead as most armies try attacking and always lose to our strong defense.

**Limitations**

One of the limitations that we faced during our implementation was construction of a bot using machine learning and neural networks due to the operating system and the high requirements required to implement these methods. This could lead the bot to predict the opponents moves and action due to scenarios implemented.

But this bot has a 60% chance to win over the Hard bot and winning againist the medium and easy bots

**Citations**

https://pythonprogramming.net/starcraft-ii-ai-python-sc2-tutorial/