

Firat Erkol – 219204648

Finn Doose-Brunns - 220200432

Fiona Stürzekarn – 220200224

Iana Mazur - 219204806

Tobias Edrich - 218201941

1) Die Datei **search.py** implementiert die Methoden für die Angriffs- und Verteidigungs-/Eroberungsschiffe.

Die Datei **assign.py** enthält die Methoden für die Schiffszuweisung und das Anhängen von Planeten.

Die Datei **monkey_business.py** enthält die Suchfunktionen zum Auffinden der benötigten Planeten und Schiffe, wie z.B. leere Planeten in der Nähe (`nearby_empty_planets`), die Listen der Suchmethoden (`defcap_empty_planets_list`, `attack_ships_target_planets_list`), feindliche Schiffe in der Nähe (`nearby_enemy_ships`), mein Planet, der nicht vollständig angedockt war (`my_planet_nearby_not_full`), befreundete Schiffe in der Nähe (`nearby_friendly_ships`), und die Bedingungen zum Navigieren der Schiffe.

Die Datei **monkey_business.py** muss über das Terminal mit dem Befehl `halite.exe -d "240 160" "python monkey_business.py"` "python monkey_business.py" ausgeführt werden, um das Spiel mit sich selbst oder dem Namen eines anderen Bots zu starten. Es ist auch möglich, die Datei "**run_game.bat**" auszuführen, um das Spiel zu starten. Der Inhalt kann über den Code-Editor geändert werden.

2 – a) Es wurden zielbasierte Agenten eingesetzt. Jedes Schiff (Agent) muss in der Lage sein, die Auswirkungen seiner Aktionen abzuschätzen, bevor es sie in der Realität ausführt.

Der Zielzustand wird durch die Suche nach den gewünschten Zielplaneten oder -schiffen entsprechend dem Schiffstyp (Angriff oder Verteidigung/Eroberung) und dem Startzustand definiert. Die Suchmethode für die Verteidigungsschiffe besteht darin, die Planeten nach ihrer Entfernung zum Schiff und der Andockkapazität des Planeten zu sortieren, um den Abbau zu maximieren und die Kosten zu minimieren, und wurde als ein Basisverfahren für "Minimax" betrachtet. Für die angedockten Planeten verwenden wir "Listenverständnis".

Für die Angriffsschiffe wurde die Anzahl der angedockten Schiffe anstelle der Andockkapazität verwendet.

Die Schiffe werden durch ein Verhältnis zwischen den Typen zugewiesen.

2 – b) Strategien zum Spielverlauf:

Rush: (sehr riskant) Die eigenen Schiffe fliegen so schnell wie möglich zum gegnerischen Team. Das Ziel ist es dieses zu überraschen und die Wirtschaft zu zerstören. Gerade auf kleineren Karten könnte diese Strategie gut funktionieren, da die Wege kürzer sind. Alle drei Schiffe bei dieser Strategie loszuschicken wäre sehr riskant. Bei nur zwei Schiffen wäre jedoch das Risiko, dass diese auf jeden Fall in der Unterzahl wären und wenn sie nicht die gegnerische Wirtschaft massiv schwächen wird man einfach überwältigt werden vom gegnerischen Team. Eine Abwandlung von dieser Strategie wäre so schnell wie möglich auf die gegnerische Seite der Karte zu kommen, um die Planeten dort zu besetzen. Dann kann man sich von dort aus in das eigene Territorium vorarbeiten und hat (sollten keine Kämpfe stattfinden) mehr Planeten besetzt, als das gegnerische Team. Eine weitere Abwandlung von dieser Strategie wäre, dass man nur ein Schiff, so schnell wie möglich, auf die gegnerische Seite der Karte schickt. Dieses Schiff wird dann daran arbeiten Planeten auf der gegnerischen Seite zu zerstören, um den Gegnern weniger Möglichkeiten der Ressourcengewinnung zu geben. Bei den Explosionen der Planeten muss man jedoch aufpassen, dass das eigene Schiff nicht davon zerstört wird (es soll so viele potentiell gegnerische Planeten wie möglich zerstören, bevor es besiegt wird).

Eco-Booming: Man sucht sich größten Planeten, der gleichzeitig am dichtesten dran ist und besetzt ihn mit allen drei Schiffen. Wenn ein weiteres Schiff produziert wurde, wird dieses auch an den Planeten andocken, wenn noch Platz ist. Ansonsten wird wieder der nächste große Planet gesucht. Diese Strategie hat als Ziel, dass die eigene Flotte an Schiffen so schnell wie möglich sehr groß wird. Wenn dies geschafft ist kann man die gegnerische Flotte einfach überwältigen und so gewinnen.

Strategien während des Spielverlaufes:

Hit n Run: Hierbei feuert ein eigenes Schiff auf ein gegnerisches (möglichst angedockt), welches alleine ist (bzw. dem nicht sofort Schiffe helfen können). Nachdem einmal (evtl. zweimal) angegriffen wurde fliegt das eigene Schiff so schnell wie möglich weg in Sicherheit (z.B. andere eigene Schiffe). Diese Strategie wird vor allem später im Spiel interessant, wenn man mehrere Schiffe hat und man verteidigende Schiffe erwarten muss. Wenn man ein wehrloses Schiff angreift wird das gegnerische Team nicht darauf warten, dass das eigene Schiff zerstört wird, sondern wird versuchen es zu retten oder wenigstens auch das angreifende Schiff zu zerstören. Dies kann einmal durch schlichte Verfolgung versucht werden oder es wird versucht den Fluchtweg abzuschneiden, was wahrscheinlicher ist. Dadurch, dass man sofort wieder wegfliegt kann man Schiffe, die zur Verfolgung oder zum Wegabschneiden losgeschickt wurden, umgehen. Führt man diese Strategie mehrfach durch kann man dadurch auch Schiffe zerstören, oder sie dazu zwingen einen Planeten zu verlassen. Diese Strategie kann auch mit (gegnerischen) Planeten durchführen.

Baiting: Ein eigenes Schiff wird nahe eines (für das gegnerische Team sehr attraktiven) Planeten positioniert. Jetzt wartet man darauf, dass ein gegnerisches Schiff kommt, um an den Planeten anzudocken. Sobald das geschieht wird das eigene Schiff das gegnerische Schiff angreifen und zerstören. Dies ist wahrscheinlich am besten möglich, während sich beide Teams noch ausbreiten und immer mehr Planeten

besiedeln. Der Nachteil ist, dass man ein Schiff zu dieser Zeit nicht Ressourcen sammeln lässt und man somit wirtschaftlich zurückhängt (wenn man annimmt, dass das gegnerische Team alles genau so macht, wie das eigene, nur kein Schiff ohne Aufgabe ist). Allerdings könnte das Schiff ohne Aufgabe auch als Wächter gegen Angriffe von anderen Schiffen fungieren. Eine Abwandlung dieser Strategie ist, dass man keinen Planeten als Köder verwendet, sondern ein eigenes Schiff. Wenn das gegnerische Team zum Beispiel Wächter einsetzt, um die Wirtschaft zu schützen, kann ein einzelnes Schiff erstmal auf einer Seite des gegnerischen Bereiches Schiffe angreifen (möglichst angedockte). Dadurch muss das gegnerische Team darauf reagieren und wenn Schiffe dem angegriffenen Schiff zur Hilfe kommen, werden mehrere eigene Schiffe auf der anderen Seite des gegnerischen Bereiches einen Angriff starten. Hierbei wird darauf gesetzt, dass die Flotte mehr zerstört (beim zerstören zählen Schiffe, Planeten und die Zeit, wenn man z.B. Schiffe zum abdocken bringt), als sie Ressourcen gekostet hat (ein Schiff kostet 72 units + die units, die das Schiff hätte abbauen können, wäre es nicht im Kampf). Eine weitere Abwandlung der Strategie ist, dass man mit einem Schiff einen Angriff antäuscht. Ziel ist es ein Schiff zum abdocken zu bringen, oder die Zeit der Wächterschiffe zu verschwenden, ohne in Gefahr zu kommen. Dies kann man auch mit mehreren Schiffen machen, allerdings besteht das Risiko, dass man weniger Zeit der gegnerischen Schiffe und mehr die eigene verschwendet hat. Außerdem muss man abschätzen, ob die „Hit n Run“ Taktik vielleicht besser für die Situation gemacht ist.

Overpowern: Diese Strategie ist quasi grundlegend für jeden Kampf. Man möchte alle Kämpfe möglichst in der Überzahl bestreiten. So nehmen die eigenen Schiffe den geringsten Schaden und der Kampf ist außerdem schneller gewonnen, was der eigenen Wirtschaft hilft.

Swapping: Bei dieser Strategie tauscht man angeschlagene, nicht angedockte Schiffe gegen nicht angeschlagene, angedockte Schiffe aus. Dies dauert im Schlimmsten Fall 10 turns (wenn man den Weg nicht mit einberechnet), führt aber dazu, dass die eigenen Schiffe immer volle Leben haben, wenn sie in einen Kampf geraten. Diese Strategie ist also besonders hilfreich, wenn es viele Kämpfe gibt.

3) Halite 2 ist ein Spiel, in dem die Spieler Planeten finden und erobern, um den gleichnamigen Rohstoff „Halite“ abzubauen und mehr Schiffe zu produzieren. Die Spieler schreiben ihre eigenen Bots, um gegeneinander anzutreten. Die Voraussetzung, um zu gewinnen, ist es, als einziger zu überleben, alle Planeten zu besetzen oder bei Zeitüberschreitung mehr Punkte zu haben. Bei Gleichstand wird ein zufälliger Gewinner ermittelt.

Halite 2 verwendet eine 2D-Ebene mit X- und Y-Koordinaten als Karte und alle Entferungen sind euklidisch. Es handelt sich um eine **Multi-Agenten-Umgebung**, da jeder Spieler mehrere Schiffe besitzt. Die Schiffe des eigenen Teams **kooperieren**, so dass eine **positive Korrelation** besteht. Die Schiffe des gegnerischen Teams **konkurrieren** gegeneinander, so dass die Schiffe des gegnerischen Teams eine

negative Korrelation aufweisen. Die Umgebung ist **vollständig observierbar**, d. h. der Agent kennt den vollständigen Zustand der Umgebung zu einem bestimmten Zeitpunkt. Es gibt keinen Teil der Umgebung, der vor dem Agenten verborgen ist. Der **Beobachtungsraum ist diskret**, weil eine endliche Anzahl von Zuständen und Agenten eine endliche Anzahl von Aktionen haben. **Die Umgebung ist statisch**, d. h. sie bleibt unverändert, während ein Agent die Umgebung beobachtet und in der Runde eine Entscheidung trifft. Es gibt eine **nicht-deterministische** Reaktion auf die Aktionen. Der nächste Schritt ist ungewiss, also arbeiten wir mit den **stochastisch kalkulierten Erwartungswerten** und **Wahrscheinlichkeiten**. Außerdem wird der Nichtdeterminismus nur durch die Aktionen anderer Agenten verursacht, so dass die Umgebung **strategisch** ist. **Die Umgebung ist sequentiell**. Die aktuelle Aktion des Agenten kann alle zukünftigen Zustände der Umwelt verändern, und die aktuelle Aktion hängt von den vorherigen Beobachtungen ab.