

**AsterMaster AI Specifications**

***Week 8***

**Allen Jones**

**4-20-15**

**A) Concept**

**1) AI takes Stimuli**

Master Aster consists of a single player ship surrounded by randomly generating asteroids. The enemy ship(s) take in the distances of asteroids, player, and other enemies as stimuli. The stimuli is simply passed to the decision tree for determination.

**2) AI decides optimal action**

AI bots are responsible for navigating the playing field without colliding with hazards, while pursuing the player. The priorities are as follows: Avoid, Flock, Attack, Pursue, circle. The AI uses these priorities and tests against corresponding stimuli, setting the first mindset that passes the test.

**3) Action carried out, results learned**

Once an action is carried out, the result is added to the Learning table that is referenced by the decision tree to test against stimuli. While an AI bot does not initially know to collide with cold/small asteroids, accidental collision update the learning table. (ex, hitting a small rock results in no damage dealt. Update: small rocks are safe to hit.

**B) Stimuli**

The bots keep a record of the nearest asteroid to them independently, and what its exact distance, to use in the decision tree, The Distance to the other enemy is also kept, for flocking determination. Similarly, the players distance is recorded for pursue determination. These various distances allow the bot to know its surroundings.

**C) Decision Making**

**1) Decision Tree**

The Decision Tree uses priority of action to test stimuli against recorded data in the Learning table. Because avoiding asteroids is highest priority, this is tested. The distance to the nearest asteroid is tested, if the asteroid is closer than the scare\_distance of the bot, it checks if the asteroid is a hazard.

**2) Avoid Asteroids?**

The Learning table checks against the nearest asteroids size and temperature, if the result is to avoid, the AI finally passes to the Avoid State. The AI initially avoids every asteroid, but when collisions with harmless asteroids inevitably occur, the Bayenesian Learning table updates, causing future checks to return that a small or cold asteroid is harmless, and it will not be avoided.

**3) Regroup?**

If the bot is safe from asteroids, it checks to see if any teammates are too far away. If so, the AI swaps to Flock state and attempts to regroup. At anytime the AI can swap states, allowing it to appear to flock and avoid at the same time.

**4) Pursue?**

When flocking is complete and the bot is close to a teammate, it begins the next check if it is far enough from the player that it needs to pursue. If so, it changes to the Pursue State. The bot will change to avoid asteroids, or when it gets close to the player.

**5) circle?**

If all else fails and the AI bot is close to the player, it will determine if it should circle clockwise or counter clockwise, and then begin to circle in that direction. Each time the AI has a chance to fire, it will briefly enter the Attack state. This occurs during both the pursue and circle states, allowing the AI to maintain its course while barraging rapid fire semi automatic shots.

**D) State Machine**

**1) Decide**

During the Decide state, the AI determines its next course of action. Rarely ever will the bot remain in the decide state, indicating all tests failed, But if so, simple Wandering will continue so the bot does not stop. Like an auto pilot, the AI will idle until getting successful stimuli.

**2) Avoid**

Most Common state, the AI avoids the current target, in this case is always an asteroid. The Avoid works like a reversed pursue, traveling simply away from the target. The bot cannot remain in the sate long usually, and returns to a intelligent course. The scare\_distance is determined by the size of the asteroid, and the Bayenesian learning table is referenced to see if the asteroid is a hazard. the avoid state will only be entered if the asteroid is a hazard.

**3) Flock**

The AI occasionally uses the flocking state when is has drifted too far from its comrade. The flock state caused the bot to pursue the target, its friend. While currently using far distances as a trigger, more enemies requires smaller flocking checks.

**4) Attack**

AI is constantly and briefly entering the attack state. The AI will fire 1 bullet, and a timer will reset. The Timer is incremented using real time, and after a third of a second the AI will fire again. Between these shots, the AI continues its route.

**5) Pursue**

Quite often the player out runs the bots, and they will begin to pursue. This simply applies thrusts in the direction of the player, until the bot enters close distance to the player. When this occurs, the bot swaps to the circle state to circle the player.

**6) Circle**

When the bot is close to you, it will avoid hitting you by maintain a circular motion around you. During this time, it will still attack, and occasionally swap to pursue to regain lost ground.

At the start of the state, the bot determines if it must circle clockwise or counter clockwise by testing speed against the quadrant it is relative to the player. The AI sets its direction (Clockwise or CC) and begins to circle by thrusting towards the appropriate next quadrant. During this time, its distance from the player is maintained at a minimum, similar to the scare distance to an asteroid.

**E) Acting**

After The decision is made and the state is decided, a switch statement uses the current state and passes the appropriate actions, usually a function corresponding to the state, exhibiting the actions described above.

**F) Learning (Extra Credit Implementation)**

The bot uses a Bayenesian learning system, beginning with a table of defaults that invokes them to avoid all asteroids. Upon collisions with asteroids, it will update the Bayenesian table, and each time the decision tree compares stimuli to the table, the updates values will be tested for, allowing the bots to learn. because they share independent brains, they learn independently. While they do not currently learn from the player, they could fairly easily, adding more difficulty to the game.