

What I Did This Week

This week I implemented the Strategy section properly. I wrote the Goal class and rewrote Strategy, OffensiveStrategy and DefensiveStrategy. My AI agent now adopts a strategy of variable length actions, and pursues that strategy until it reaches the goal or the cut off time is reached. At that point, a new strategy is selected and pursued. Weapons are not implemented yet, but the OffensiveStrategy would only require a WeaponsGoal and a WeaponsAction to be defined to complete a damage-giving offensive strategy.

I also implemented Situation, DefaultSituation and the SituationalStrategyAgent classes. These classes allow for different situations to change strategy selections based on the Situations defined by the game designer.

The code is in <https://github.com/hareeshganesan/Vooga/>.

How to Run It

Run the MainGame class. The AI agent will now select either an offensive or defensive strategy (printing it in the console), and pursue that strategy for 10 seconds or when the goal is reached, whichever comes first. You can also define a new Situation class if you'd like with different offensive and defensive strategy weightings, and add it to the AI agent by looking at the declaration in the LevelObjectFactory.

Note: The Collisions Engine was not updated with the Sprite animation changes. As a result, the sprites exhibit strange behavior that has no algorithmic effect on the AI strategies, but may casually magnetize players and mess up and attach two characters to each other. The structure is there though, and the general behavior should still be obvious.