Project 4
CS4013/5013: Artificial Intelligence

Initial/demo due date: May 1, 2018 11:59pm

Final code and write up due date: May 10, 2018 6:30pm

NOTE: You may NOT use slack days on the May 10 deadline, per university policy!

Introduction

This project focuses on multi-agent systems in a new environment: Capture the Flag. We are specifically paralleling the spacesettlers project but using drones. The learning objectives remain the same: learning about multi-agent coordination and tying together the different AI components you have learned about this semester.

The rules for drone capture the flag are:

- Each drone will be equipped with an AR marker attached to card stock (Dr McGovern will provide this if you need it). These will be specified for each team and each drone ahead of time so that all teams know the identities of all other drones. AR markers should be facing forward. Feel free to color the rest of the card stock so that each drone is easily identified as belonging to a team by the observers.
- Each team will have an AR marker to represent their flag and a second marker to represent their base. These identities will be known to all teams in advance.
- There will be a set of AR markers mounted evenly on each wall to help with localization. These will be known to every team. The room is approximately 12 meters by 10 meters. Given the window spacing, we will mount AR Markers every 2 meters along all 4 walls. Marker 1 will start in the north west corner of the classroom (basically by the door behind the podium) and will proceed along the north (window) wall. Marker 7 will start on the east wall. Marker 12 on the south wall and Marker 18 will start on the west wall.
- A team can generate additional markers for its own internal use so long as it sticks within its number-space. 0-99 belong to the wall markers (and extra may not be generated). “Flying Twisters” own 100-199 and the color purple. “Game of Drones”

owns 200-299 and the color red. “Don’t Mind Me :)” owns 300-399 and the color green. “Blue Eyes Ultimate Dragon” owns 400-499 and the color blue. 101,201,301,and 401 will be the base for that team. 102, 202, 302, and 402 will be the Flag. 103-104, 203-204, 303-304, 403-404 are drones. Any numbers not mentioned are free to be used by the team.

- Each team can choose where to place their flag and their base within the competition room. The drones should start within 6 feet or 2 meters of the team’s base. The flag and the base must be mounted horizontally on a wall or the side of the podium. They may not be mounted on a ceiling, floor, other drone, chair, person, etc. Tags may not be obscured by anything other than a drone.

- To “capture” a flag, you must first scan the opposing team’s flag and take it back to your base by then scanning your base. Touching a flag is against the rules. You can only “capture” the flag once before you deposit it at base, no matter how many times you scan it within an image frame (e.g. the 30 fps scan rate doesn’t mean you get 30 flags!). You cannot pickup another flag until the flag is deposited at your base.

- To “tag” a drone or a base, you can scan its AR code. A drone can only be “tagged” once within a 2 second window. This prevents you from scanning a drone over and over again using the 30 fps video. A base can only be tagged once every 30 seconds.

- Any attempt to intentionally maim any flag, base, drone, or marker will be considered to be an immediate disqualification event for that team and game and the opponent will win, even if they have not yet scored.

- Drones must communicate within their own team (see provided extra hardware below) but may not communicate with other teams (honestly this is more of a hardware limitation than anything else. We simply can’t support it across teams as we would need a large number of USB ports on each laptop).

- Drones may play a sound effect up to 5 seconds long any time that it scans a tag. Tunes that differentiate the different kinds of tags and help a team keep track of score are highly encouraged. Any sound or song is acceptable provided it is: not offensive, not explicit, does not consist entirely of body sounds (e.g. no farting, you know which team I’m talking to), and does not contain any political sounds.

- Games will be 4 minutes long. The team with the maximum score at the end of the game wins that game. There are no time-outs. If a drone crashes during a game, the team who owns the drone can do their best to get it up and running again. Make sure you don't lose any points if you restart your program since the official scores only happen at the end of the game (see below).

- In order to score a game, each team will need to report the following at the end of a game.
  - Number of opponent drones tagged (via scan)
  - Number of own drones tagged (yes, you can tag your own drones, just as in spacesettlers)
  - Number of opponent bases tagged
  - Number of flags deposited at your base by each drone

The final score for a team will be calculated from the printouts at the end of the game. The sound effects during a game will be for verification but are not considered official.

$$\begin{aligned} Score(team) &= 10 * \text{Number of flags collected by team} \\ &+ 2 * \text{Number of bases tagged by team} \\ &+ \text{Number of opponent drones tagged by team} \\ &- \text{Number of drones tagged by opponent team} \end{aligned}$$

The learning objectives of this project are to:

1. Create a team of agents that can cooperate to achieve a common goal. The agents must cooperate to achieve the goal.
2. Implement and use planning to enable your agents to achieve their goals efficiently.
3. Tie together the AI methods you have been learning about this semester to make an effective multi-agent team.

Your task for this project is outlined below. The main foci of this project are planning and multi agent coordination.

- Create an team of two drones that coordinate their behavior with one another to achieve the goal of acquiring flags and tagging enemy drones. All drones should move around the environment intelligently.

- Identify a set of high-level behaviors appropriate for each drones in your team. Implement the set of high-level behaviors designed by your group. You are not constrained by a specific method for implementing your agent but you should use what you have learned so far this semester.

- Implement PLANNING within your team or within a single agent. I suggest that you implement it at the highest level (e.g. deciding which high-level behavior to choose at any time) but you could also use planning at a lower level such as improving A* or other navigation functions. For each behavior used by your planner, specify a set of pre-conditions and post-conditions, appropriate for use in planning.

- Implement an approach for effective multi-agent coordination. Given the framework of the drones (e.g. communication is only possible via USB serial cables), this is most easily done using a common language between the two teams. Ensure that you re-plan as frequently as events warrant.

- Integrate methods from other parts of the semester into your agent. For example, if you think learning will help, integrate a learning method. Likewise, use A* if you can. Create a good knowledge representation. Use search effectively.

Additional software

- You will need to install the ar-markers library. This library can be found at `https://pypi.python.org/pypi/ar-markers/0.4.1`. The installation instructions are simple (use pip). This library will be used to scan your markers. The markers will be on canvas but you can generate more using the library.

- A suggestion on a cross-platform sound playing platform for python: `https://pypi.python.org/pypi/playsound`. On the mac, you will need this to make it work: `pip install pyobjc`

- In terms of crashes and restarting the score, this may be a useful thread for you: `https://stackoverflow.com/questions/21120947/catching-keyboardinterrupt-in-python-during-`

Extra-credit opportunities

To keep grades within a fair range, all extra credit will be capped at 10 points. This means that no project can receive a grade higher than 110.

In-class demos and competition

All teams will compete with each other in a double elimination tournament on the final day of class (May 2, 3-4:15). Since this is very similar to the ladder for the other project but you only get one chance at it, there will be extra credit awarded for your final placement in the tournament. If the winning team is a pair of student teams from the class, both project groups will automatically earn 10 extra credit points. Likewise, if the second place team is a pair of student teams from class, they will earn 8 extra credit points. Third place receives 5 points and fourth place receives 3 points. Note that there are two teams competing in the tournament (using the same rules and same Mambo drone) as specified above but who are not enrolled in the class. These teams are eligible to win first and second place, so it is not automatic that students will receive 10 or 8 points! Note, any team that demos successfully will get a minimum of 3 extra credit points, per these rules.

Wanted dead or alive: bugs or new features

The pyparrot library is brand new and likely has some bugs to work out. We want to know about them and fix them! If you find a bug, you can receive extra credit according to the following scale:

- **1 or 2 points:** General bugs are likely given how new this code is. Finding a bug and reporting it can get you 1 point. Fixing the bug and giving us the fix (you can't check it in directly but you can give it to us in the bug report) can get you two points. Both bug and fix must be verified for any extra credit to be awarded. Bug reports should be verbose and state exactly what happened. Simply stating "it broke" will not count. Feel free to report them on GitHub using issues.
- **3-10 points:** If you a particularly clever way to implement a feature that is not currently implemented, please submit a pull request with your code. If I accept it, you will gain extra credit worth 3-10 points, depending on the feature and the code you submit (one line quick fixes are worth less than more complicated features).

Wanted (alive please): creative individuals

Creativity is highly encouraged! To make this real, there are up to 10 points of extra-credit available for creative solutions.

- Document it in your writeup! I can't give extra credit unless I know you did something extra.

- You must still be doing learning (speak to me if you want to verify that your approach is still learning).
- Remember that by being creative I am referring to the algorithm and *not* to the ability to creatively download code. **All project code must be written exclusively by your group except for Visual SLAM, which I am explicitly giving you permission to download. You are also welcome to use any functions in the opencv libraries.**

Those pesky details

1. Update your code from project 3. You can update your code at the command line with "git pull". I have released fixes for the codebase since project 3 and have made significant improvements in vision processing since then so you want the new code!
2. Create a cooperative multi-agent program to play CTF. You will cooperate with your chosen other project group. The agents should communicate and solve common goals, as described above.
3. ONE member of your team should submit your final code to canvas, along with your writeup.
4. Bring your drone to class on the day listed on the schedule for project 4 demos. Be prepared to demo for EC!

Grading rubric

This project has two deadlines and thus two parts to the rubric!

- Demos on May 2 (Note these points are separate from the EC points for winning)
  - 10 points for participating in demos on Wednesday May 2 with code that is turned in by Tuesday May 1 at 11:59pm and is using planning and multi-agent coordination (basically a preliminary version of your project, ready enough to demo).
  - 5 points if you participate in the demos but your code is NOT using planning or effective multi-agent coordination

  - 0 points for choosing not to participate
  - Plus 5 points EC if your project is actually completed and ready to grade by the demo date (you must indicate this by turning in your writeup and commenting that it is ready to grade).

- Planning
  - 20 points for correctly implementing planning within your group's set of agents. A correct planner will choose among the actions intelligently using PDDL-style planning. Planning code should be well documented to receive full credit.
  - 18 points if there is only one minor mistake. An example of a minor mistake would be not re-planning at the correct time or incorrectly specifying the pre or post-conditions for an action.
  - 15 points if there are several minor mistakes or if documentation is missing.
  - 10 points if you have one major mistake. An example of a major mistake would be not using an pre or post conditions or incorrectly applying them to your state representation.
  - 5 points if you accidentally implement a search algorithm other than planning that at least moves the teams around the environment in an intelligent manner.

- Multi-agent coordination
  - 30 points for correctly implementing multi-agent coordination within your group's set of agents to achieve a common goal. A correct multi-agent systems will coordinate team behavior intelligently and rarely have issues running into each other and is well documented.
  - 25 points if there is only one minor mistake. An example of a minor mistake would be not always coordinating well in a decentralized team or sometimes (rarely!) assigning incorrect roles in a centralized system. Note that errors in planning should count above rather than here. This focuses on the multi-agent coordination itself.
  - 20 points if there are several minor mistakes or if it is not well documented.
  - 15 points if you have one major mistake. An example of a major mistake would be not frequently failing to coordinate actions.
  - 10 points for several major mistakes.

  - 5 points if you somehow manage to coordinate occasionally but it is entirely accidental within the code

- High level behaviors
  - 10 points for making good use of your knowledge from AI so far to create good high-level behaviors
  - 8 points for one minor mistake.
  - 5 points for several minor mistakes or one major mistake.
  - 3 points for implementing a behavior that is at least somewhat intelligent, even if it doesn't fulfill the requirements of the high-level behavior specified by your team.

- Knowledge integration
  - 10 points for making good use of your knowledge from AI so far. For example, integrating your learning results into your planning or mulit-agent coordination system or making excellent use of your A* agent would both count here. This is a good spot for creativity!
  - 5 points for trying to integrate knowledge from AI so far but only doing a partial job of it

- Coding practices: We will randomly choose from one of the following good coding practices to grade for these points. Note that this will be included on every project. Are your files well commented? Are your variable names descriptive (or are they all i, j, and k)? Do you make good use of classes and methods or is the entire project in one big flat file? This will be graded as follows:
  - 10 points for well commented code, descriptive variables names or making good use of classes and methods
  - 5 points if you have partially commented code, semi-descriptive variable names, or partial use of classes and methods
  - 0 points if you have no comments in your code, variables are obscurely named, or all your code is in a single flat method.

- 10 points for your writeup. The writeup must contain:

- **3 points:** A full description of the high-level behaviors chosen by your team
- **3 points:** A full description of the planning approach chosen by your team. This includes a full description of the goal condition, the state representation, and the pre-conditions and effects for each action.
- **2 points:** A description of how you implemented multi-agent coordination
- **2 points:** A description of how you integrated your knowledge from other parts of the course into your existing agent.
- Any writeup longer than 2 pages will be given an automatic 0