

Task 10 – Spike: Tactical Steering (Hide!)

Context:

Tactical analysis by agents can be used to build on basic steering behaviours to create intelligent behaviour. An agent that is being pursued by another can make use of tactical analysis to identify an object in the game environment with which to hide behind and so be out-of-sight from a hunter.

Knowledge/Skill Gap:

The developer needs to know how to give an agent the ability to find and seek good hiding locations using tactical analysis of an environment.

Goals/Deliverables: [CODE] + [SPIKE REPORT]

Create a hunter-prey agent simulation for two or more agents, in which "prey" agents avoid "hunter" agents by concealing themselves behind objects in the environment. The simulation must:

- Include several "objects" that prey can hide behind (simple circles).
- Show a distinction between the "hunter" and "prey" agent appearance and abilities.
- Show an indicator ("x" or similar) to indicate suitable "hide" locations for prey to select from
- Prey agents must select a "good" location, and head to it, based on tactical evaluation.
- Do NOT hide "inside" objects – rather find a location outside (behind).

Note: leave instructions on how to use your code (keys to use) with your code / report.

Start-End Period: Week 7 – Week 8

Notes:

- The environment "objects" can (should) be a simple circle.
- Appearance changes: shape, colour, size, line thickness?
- Ability changes: mass, steering force limits, turn-speed limits.
- The identified "hide" locations are per-hunter, so start with one hunter only. See the extensions for multi-hunters.

DON'T OVER DO IT! ☺ The hunter doesn't need to actually "eat" the prey.
Don't get distracted (on the core work) - stay on target! Do extensions later if you are so inclined.

Planning Notes:

- Use the existing lab code and create a new project. Remove what you don't need from the code. (Always keep your code as small/specific as possible.)
- Add circles to the environment that agents can hide behind
- Add two agents -- one designated as "hunter" and the other the "prey"
- Make the hunter wander randomly. (It can "chase" as an extension.)
- Based on the hunter, draw a line from the hunter to the target object and then extend it through the object (by the object size plus a bit more). This is the hiding spot (as described in lecture notes).
- Draw "x" at each identified hiding spot.
- Change the colour of the "best" hiding spot for the current prey agent. The "best" is per-prey agent, so only start with a single prey agent. (Multiple prey agents mean lots of different "best" locations, but the same set of hiding spots... which the prey should/could share.)

Extensions:

- Give agents the ability to avoid objects. i.e. Simple proximity tests or try fancy proximity boxes and circle-line intersection tests.
- Create multiple hunters and let each prey agent consider multiple choices intelligently. Tip: associate each hiding spot with the "object" so that you can group them together.
- Not all hiding spaces are equal, so let prey agents also consider not only their proximity to the hiding spaces, but also the resulting proximity to the hunter and perhaps the "danger" (how exposed) while they get there. ie. prey should select hiding spaces further away from the hunter if it doesn't expose them to the hunter on the way.
- Make the hunter actually "chase" the prey it can see, and "eat" the prey when it contacts them. You'll probably want to re-spawn new prey afterwards...