

Program: GDP1

Course: INFO6017 – Artificial Intelligence – Winter 2024

Professor: James Lucas

Project # 1: Steering Behaviour Basics!

Weight: 25% of “project” mark

(or an even weight, given less or more than 4 total projects)

Due Date: Sunday, February 11, 10:00pm

Note: This project must be done independently. No group submissions are allowed.

Description and Purpose

For this project you will create a scene that demonstrates the following steering behaviours:

- Seek
- Flee
- Pursue
- Evade
- Approach

The scene consists of a 3rd person controllable player, at least 5 AI agents/enemies (1 per behaviour), and some kind of landscape to help discern movement.

Your Submission

- **Submit a single .zip containing everything required to build and run your project.**
- A ReadMe file is **required** for this project (see Implementation Checklist for details).

“Show-Stopper” Marks

While these aren’t “worth” any marks, they could deliver large penalties, or in some cases a mark of zero.

- Does not compile (mark=0)
- Build fails, doesn’t run, or runs with crashes/errors (I may investigate for a simple fix to something like a last-minute typo, but if it’s not a super simple fix for me then... mark=0)
- No/Awful/Nonsensical documentation (-30%)
- No/Terrible/Painful to look at conventions/style (-50%)

Implementation Checklist

1. The Player:

- 3rd person camera
- Model is always visible
- Has controls to rotate left and right, move forward, backward, left, and right.
- Is faster and more maneuverable than the agents (moves faster, rotates faster).

2. The Agents:

- There is at least one agent per behaviour in the scene.
- Visibly distinct, depending on the behaviour, perhaps by unique color or unique model.
- Current behaviours of the agents are easily changed. (eg. I can change a single line in a config file, or change a single line in code somewhere to turn a “pursue” into a “seek”).

3. The Landscape:

- There is a simple static landscape to help discern movement and distances. Don't go nuts here, think 'checker board' not 'grand canyon'. Nothing too busy or obstructive.

4. The ReadMe file describes:

- How to build and run the program. (Configurations, Platforms, etc.)
- Player controls: what they are, and where/how to alter player speeds.
- How to tell which agent has which behaviour (eg. “Pursue”: red ducks; “Flee”: blue boats, etc.)

Marking Scheme

This is how the marks are divvied up.

Item	Marks
Notes for marking: <ul style="list-style-type: none">For all behaviours below, agent motion should be smooth and gradual (ie. Not 'snapping' to the goal direction)The “behaviour” should not be class-specific – there should be a single class (eg. “cEnemyObject”, “cAgentEntity”, etc) that can exhibit any behaviour.	
Seek	10
Flee	10
Pursue	10
Evade	10
Approach	10
TOTAL	50

Additional Requirements

- While you may freely “borrow” mine (or anyone other) code ***but*** your code should be “sufficiently” different from mine. See the “plagiarism” test, later in this document, for more details.
- Further, you ***cannot*** simply use an existing game engine (or part of a game engine), even if it’s a “from source” engine (i.e. you have the entire source) to complete this assignment; it should be either completely new or significantly modified. This includes, but is ***not*** limited to: Unity, Unreal, Cry, Anarchy, XNA, Cocos, Ogre, the framework from the OpenGL text, etc. In other words, you are expected to have made the vast majority (essentially all) of the engine ***in this term by yourselves, from “scratch”*** - i.e. starting from something as rudimentary as the “OpenGL Book” code or the GLFW starter code (we started with that in class).
- You also may not use any “3rd party” physics code/libraries like bullet, havok, etc. The exception to this is code taken from the text book, of course, but you also can’t just use Ian Millington’s complete “cyclone” engine code – you can use code inspired from that, but it has to be almost completely “yours” and created this term.
- The most “engine” type code you can use is limited to GLFW, glad, and OpenGL Math (glm); anything more is almost certainly “too much engine” code.

75/10-year old “squinty eye” plagiarism test:

(Credit: Feeney)

I have very little tolerance for plagiarism, but many students are unclear about what it is.

Basically, it’s submitting somebody else’s work as your own.

There is sometimes some confusion over this because you could argue nothing is actually “unique” (see: <http://everythingisaremix.info/> for a fascinating overview of this).

The whole point of assignments/tests/projects in this course (or any course, really) is to try to see if you are actually able to ***do*** the coding that’s asked of you. In other words: How competent are you? Handing me someone else’s code and/or making a trivial change isn’t good enough.

Also, it’s illegal:

- <http://www.plagiarism.org/ask-the-experts/faq/>
- <http://definitions.uslegal.com/p/plagiarism/>
- <http://en.wikipedia.org/wiki/Plagiarism>

- <https://www.legalzoom.com/articles/plagiarism-what-is-it-exactly>

In other words, I'm not going to be drawn into a giant debate over how "different" your code is from mine or anyone else's, if any sensible person (including me) would conclude that the code/application is pretty much the same thing, then it is. It is up to my discretion to decide this.

- While you may freely "borrow" mine (or anyone other) code **but** your code should be "**sufficiently**" different from mine (you might want to replace the word "sufficiently" with "significantly").
- In other words, you **cannot** simply use an existing game engine (or part of a game engine) to complete this assignment; it should be either completely new or **significantly** modified.
- How will I determine this?
 - If I showed your application and/or your source code to either a pragmatic 75-year-old mother, or a typical 10-year-old, or even some random person walking down the hallway (i.e. a non-expert), and they looked at it, tilted their heads, squinted their eyes, and said "you know, they look the same," then they **are the same**.
 - Another test would: How much time it would take for a **competent programmer** (for example, **me**) to make the changes you are submitting? The point here is that I don't "care" if you tell me "But it took me *weeks* to make the changes!" Fine, but if I can make those same changes in 10 minutes, then not a lot of work has been done (certainly **not** sufficient work – these projects should show take **days** of work having been done).