

ICP2025 | ICP3025

AI for Games | Applications of AI

Laboratory 3: Steering Behaviours

Lab 3 – Steering Behaviours

|  |
| --- |
| **Please Note:**  Steering behaviours will be covered in greater detail in the lectures if you struggle with any of the concepts of this lab. It is important that you implement at least one of these behaviours but you will probably find that you need to implement at least two to get your NPC to act in an interesting way. The tasks range from the simplest (1) to the most difficult (5). Assistance is available during lab sessions only.  There are many correct ways to implement these steering behaviours and you may decide to adapt the implementation to suit your needs.  It is highly recommended that you read the original paper where all the underlying theory is described in detail - <http://www.red3d.com/cwr/steer/> |

The goal of this lab is to develop the locomotion of your NPC so that it moves around your environment in an improvisational manner. In the previous lab you will have implemented a simple point and move script. In this lab you will expand on these simple scripts to produce more realistic movement and behaviour.

Steering behaviours have been a standard of the creative industries since Craig Reynolds introduced them in 1987. They have underpinned technologies behind CGI crowd scenes (e.g. Lord of the Rings) and the majority of NPC movement behaviour in 3D games.

It is recommended you implement at least one of the steering behaviours described in the 5 tasks. Though for the highest marks you must implement at least two and attempt a bonus task.

|  |
| --- |
| It is recommended you write-up the steering behaviours you have implemented for this lab and post it in your blog report. You should describe your steering behaviour implementation and its potential use within your game. You should aim to include the following information.   * Which steering behaviours you chose to implement and why. * A detailed description on how you implemented it including a code sample. * A short review of the steering behaviour, discussing how successful you believe your implementation to be and how realistic you think the behaviour is. * If you have attempted one of the bonus exercised provide a short description of what you implemented. |

|  |
| --- |
| **1. Wander**  <http://www.red3d.com/cwr/steer/Wander.html>  <http://goo.gl/ylCHtD>  Create an NP, which wanders around your environment by implementing the wander behaviour. There are many ways to achieve this – the links above offer a very robust method. |
| Bonus – Make the wander a tuneable parameterised behaviour using either a slider (Netlogo) or a public variable (Unity). |

|  |
| --- |
| **2. Seek and Flee**  <http://www.red3d.com/cwr/steer/SeekFlee.html>  <http://goo.gl/QYYIAs>  Implement both a Seek and a Flee behaviour so that the NPC is attracted towards one GameObject and away from another. You can implement both behaviours onto a single NPC or have individual seek and flee NPCs. |
| Bonus – Implement some method of switching between the two behaviours in one agent. There are several ways to combine these behaviours, such as averaging the vector or using a finite state machine - it is up to you to select an approach. |

|  |
| --- |
| **3. Pursue and Evade** <http://www.red3d.com/cwr/steer/PursueEvade.html> <http://goo.gl/trn5SM>  The Pursue and Evade behaviours are built upon the simpler Seek and Flee behaviours. However, if your target is moving these behaviours look blunt and lacking in intelligence. With the pursue and evade behaviours, instead of simply pointing at the target’s current position, the NPC makes an estimation about the direction of the target and attempts to shorten the distance. Implement pursue and evade on one of your NPCs using the information available at the links above. |
| Bonus – Think of a situation where seek and flee may produce a more realistic behaviour pursue and evade. |

|  |
| --- |
| **4. Obstacle Avoidance** <http://www.red3d.com/cwr/steer/Obstacle.html>  <http://goo.gl/OvGiR3>  Pick one of the previous steering behaviours and add obstacle avoidance to the behaviour. |
| Bonus – Some things need to be avoided more than others. For example, a wall needs to be avoided, but you may choose to run through fire if there was no other method of escape. How could you modify this behaviour to represent this form of ranked avoidance within your NPC? |

|  |
| --- |
| **5. Flocking** (or other combined behaviours)  <http://www.red3d.com/cwr/steer/>  <http://www.red3d.com/cwr/boids/>  <http://goo.gl/Jb0tU2>  <http://goo.gl/rtuwNY>  Flocking is the combination of three steering behaviours, Separate, Align and Cohere, and as an algorithm underpins many crowd and group behaviour models. However, there are several other combined behaviours available (see first link which contains a list with details). To complete this task, implement a combined steering behaviour of your choice. Flocking may be an easy choice as there are many models and tutorials available on the internet (search for keyword “boids”). |
| Bonus 1 – Nothing natural moves in a perfectly straight line – try combinging the wander behaviour on top of your steering behaviour to produce more natural movement.  Bonus 2 – Think of a specific scenario for your NPC and develop your own combined steering behaviour to suit it. |

# Unity Resources

Translate and Rotate

<http://unity3d.com/learn/tutorials/modules/beginner/scripting/translate-and-rotate>

Random.Range(…)

<http://docs.unity3d.com/ScriptReference/Random.Range.html>

LookAt  
<http://docs.unity3d.com/ScriptReference/Transform.LookAt.html>

RotateToward  
<http://docs.unity3d.com/ScriptReference/Vector3.RotateTowards.html>

Character Controller (Kinematic based motion)  
<http://docs.unity3d.com/Manual/class-CharacterController.html>

Basic A.I. Character (Kinematic based motion)  
<http://unitygems.com/basic-ai-character/>

Roll a ball (Physics based motion)  
<http://unity3d.com/learn/tutorials/projects/roll-a-ball>

# Netlogo Resources

ifElse

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html#ifelse>

Face

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html#face>

Towards

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html#towards>

Subtract Headings <http://ccl.northwestern.edu/netlogo/docs/dictionary.html#subtract-headings>

Any?

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html#any>

In-Cone

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-cone>

In-Radius - <http://ccl.northwestern.edu/netlogo/docs/dictionary.html#in-radius>