## **Artificial Intelligence Research 3**

## Author: Emrys Quin

# Terminology

In terms of within programming, give a **definition** for the following **terminology** using your **own words.** *(Minimum* ***two sentences*** *each)*

**Inheritance:** The passing on of everything contained in one object/class to a child class/object, which may alter variables/functions/etc without affecting the parent class. All logic implemented into the parent will be automatically adopted by the children.

**Polymorphism:** meaning ‘many forms’, this is used to alter the methods/properties of a parent class so that multiple versions exist. Specifically concerning methods, this is done with ‘method overloading’ and ‘method overriding’. Overloading is creating multiple methods in a class with the same name but different parameters, and overriding is adjusting the base method to alter the result.

**Parent/Base/Superclass Class:** A superclass is defined as being above, or not having more, and simply refers to a class to that is inherited from by another class. Parent/base classes are more specific and refer to a specific class being inherited from by children/subclasses.

**Child/Derived/Sub Class:** These classes are the ones deriving their base logic and properties from a parent/base class. They may override/overload methods, or add new functionality/variables but they must inherit everything defined within their parent class.

**Generalisations:** Extracting common properties from two or more classes, then combining them into a generalised superclass. The inherited properties can be variables, attributes, or methods. These are ‘general’ attributes/properties that are inherited by multiple classes.

**Specialisations:** This is the reverse process of generalisation, as it requires a subclass from a base class. It is not the amalgamation of two or more classes that generalisation is, rather it ‘specialises’ its information/logic into a single child class on top of the generalised information it already possesses.

# Behaviours

Choose two examples of games that use predator and prey AI. Describe how the behaviours are used by the game, and how does that provide a good player experience.

Choose two different Genres for your two examples.

*(Minimum* ***two******sentences*** *each)*

(Genre e.g. include: Action, First/Third person shooter, Real time strategy, Survival horror, Racing, Role-playing)

**FPS(Horror) Predator and Prey AI in Alien : Isolation**

In Creative Assembly’s first foray into the First-Person Shooter genre (coming from a background of RTS titles) the player takes control of Ripley (the prey), a character with no effective recourse for combating the eponymous Alien (the predator AI) which stalks her throughout the course of the game. By using a guiding design philosophy which focuses on unscripted AI behaviour (referred to as ‘psychopathic serendipity’[[1]](#footnote-1) by the devs) rather than scripted jump scares, the predator AI in Alien Isolation prioritises versatility and adaptation over predictable/deterministic patterns.

The behaviours of the predator AI rely on various modes of searching as well as reacting to input. Because the prey is not another NPC, it would be inherently unfair for the predator to always know exactly where the player was. Therefore, the Alien has to actively search for the player utilising behaviour trees, which determine action in reaction to the Alien’s sensory input (sight, sound and even touch all implemented). These behaviour trees expand and become more complex as the player achieves certain conditions over the course the game, which simulates the predator ‘learning’ and adapting to the player’s choices.

Whilst the predator AI searches for the player, it generates a hidden ‘menace’ value, designed to scare and keep players on edge whilst the Alien is nearby/in view. The Alien will react slowly to the player if it catches a brief glimpse of them, but more immediately if the player fires a gun or stands still for a long period of time.

**Racing Predator and Prey in Mario Kart**

In the racing game Mario Kart players acquire power-ups over the course of a race. These power-ups are randomly assigned items which vary in power, and are acquired after a player picks up one of several coloured cubes which float over the points of race-course. A good example of predator and prey AI is demonstrated by one specific power-up: The Blue Shell.

The Blue Shell is a power-up which was implemented as a form of negative feedback loop, i.e. designed to offset a player’s advantages and keep a level playing field. It, when fired, actively seeks whoever is in first place and stuns them for a brief period of time, giving other players a chance to catch up. The shell (predator) singles out a particular target – whoever’s in first place (prey) - and utilises pathfinding AI to catch up to and stun them.

# Pseudo Code and UML Diagram

**Write very simple pseudo code** and include a **State** **UML diagram** displaying and describing the behaviour of predator and prey AI.

LIFE

Float speed

Float health

*Move()*

PREDATOR : LIFE

Float damage

*Attack()*

*Chase()*

PREDATOR STATES

Wander territory

Seek prey

Attack

PREY : LIFE

*Graze()*

PREY STATES

Wander

Run

Hide

# Class UML Diagram

Create a **class UML diagram** that shows how the **inheritance and polymorphism** will work in the predator and prey AI system.

# Identify generalisations and specialisations

Using your parent “Life” class and the “Predator” and “Prey”. Identify the methods and variables that are the same and unique to each class.

**Life:**

Generalisations:

**Predator:**

Specialisations:

**Prey:**

Specialisations:

# Testing

Create a test plan and run the tests to make sure all the behaviours you have created are functioning correctly.

## Predator:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **Test Case Description** | | **Comments** | | **Pass/Fail** |
| 1 | [What is being tested] | |  | |  |
| **#ID** | **Step** | **Expected Results** | | **Obtained Results** | |
| **1** | [Test being run] |  | |  | |
| **2** |  |  | |  | |
| **3** |  |  | |  | |

## Prey:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#ID** | **Test Case Description** | | **Comments** | | **Pass/Fail** |
| 1 | [What is being tested] | |  | |  |
| **#ID** | **Step** | **Expected Results** | | **Obtained Results** | |
| **1** | [Test being run] |  | |  | |
| **2** |  |  | |  | |
| **3** |  |  | |  | |

# Journal

Questions for Andrew:

1. How do we implement flocking behaviours into the state machine so that the agents are *always* true to cohesion, alignment and avoidance?
2. Would making each state a composite behaviour be applicable?

1. Tommy Thompson, ‘The AI of Alien: Isolation | AI and Games’, YouTube, 2016. <https://www.youtube.com/watch?v=Nt1XmiDwxhY&ab_channel=AIandGames> [↑](#footnote-ref-1)