# CAP 4053 Artificial Intelligence for Computer Games: Game AI Design

# CAP 4053 Artificial Intelligence for Computer Games - Lecture Notes

## Game AI Design

### Introduction

Game AI design is the process of creating intelligent behaviors for non-player characters (NPCs) in a game. The goal of game AI design is to create NPCs that are believable, challenging, and fun to play against. This lecture will cover the fundamentals of game AI design, including the different types of AI, the components of an AI system, and the techniques used to create intelligent behavior.

### Types of AI

There are two main types of AI used in games: rule-based AI and learning AI.

Rule-based AI is based on a set of predefined rules that determine how the NPC will behave. Rule-based AI is often used for simple behaviors, such as navigating a maze or avoiding obstacles.

Learning AI is based on algorithms that allow the NPC to learn from its environment and adapt its behavior accordingly. Learning AI is often used for more complex behaviors, such as playing a game of chess or driving a car.

### Components of an AI System

An AI system consists of three main components: a knowledge base, a decision-making system, and an action selection system.

The **knowledge base** stores information about the game world, such as the locations of objects and the state of the game.

The **decision-making system** uses the information in the knowledge base to make decisions about what the NPC should do.

The **action selection system** takes the decisions made by the decision-making system and selects the appropriate actions for the NPC to take.

### Techniques for Creating Intelligent Behavior

There are several techniques that can be used to create intelligent behavior in NPCs.

State Machines are a type of rule-based AI that uses a set of states to determine the behavior of the NPC. Each state represents a different behavior, and the NPC transitions between states based on events in the game world.

```
Start of Code
// Example of a state machine
enum State {
        IDLE,
        WANDERING,
        CHASING
};
State currentState = IDLE;
void Update() {
        switch (currentState) {
                case IDLE:
                        // Do idle behavior
                        break;
                case WANDERING:
                        // Do wandering behavior
                        break;
                case CHASING:
                        // Do chasing behavior
                        break;
        }
}
End of Code
```

Behavior Trees are a type of rule-based AI that uses a hierarchical tree structure to represent the behavior of the NPC. Each node in the tree represents a different behavior, and the NPC transitions between nodes based on the result of the previous node.

```
Start of Code
// Example of a behavior tree
Node rootNode;
void Update() {
        rootNode.Execute();
}
End of Code
```

Finite State Machines are a type of learning AI that uses a set of states to represent the behavior of the NPC. The NPC transitions between states based on rewards or punishments it receives from its environment.

```
Start of Code
// Example of a finite state machine
enum State {
        IDLE,
        WANDERING,
        CHASING
};
State currentState = IDLE;
void Update() {
        switch (currentState) {
                case IDLE:
                        // Do idle behavior
                        break;
                case WANDERING:
                        // Do wandering behavior
                        break;
                case CHASING:
                        // Do chasing behavior
                        break;
        }
        // Update state based on rewards or punishments
        if (reward > 0) {
                currentState = WANDERING;
        } else if (punishment > 0) {
                currentState = CHASING;
        }
}
End of Code
```

Neural Networks are a type of learning AI that uses a network of interconnected neurons to represent the behavior of the NPC. The neurons are trained using a set of input and output data, and the NPC's behavior is determined by the weights of the neurons in the network.

```
Start of Code
// Example of a neural network
NeuralNetwork network;
void Update() {
        // Get input data
        float[] inputs = GetInputData();
        // Feed input data into the network
        network.FeedForward(inputs);
        // Get output data from the network
        float[] outputs = network.GetOutputs();
        // Use output data to determine behavior
        DoBehavior(outputs);
}
End of Code
```

### Practice Multiple Choice Questions

1. What are the two main types of AI used in games?

A. Rule-based AI and learning AI

B. Rule-based AI and expert systems

C. Learning AI and expert systems

D. Rule-based AI and neural networks

Answer: A. Rule-based AI and learning AI