AI Control Wheel System
Technical Design Document

Benoit Poyser

Contents

High Concept 3
System Break Down 3
Wheel System 3
AI Control System 3
Data/Variables 4
Communication 4
Tool Assessment 4
Required Systems/Components 4
Validation Techniques 5
Functional Testing 5
Debug System 5
Assets Needed/Used 5
Implementation Tasks 6

High Concept

The AI Control Wheel System will allow the player to control a friendly AI sending instructions through a command wheel. Those instructions are: Follow the player, Stay, Roam and Move to the current position of the player and then wait there until receive another instruction.

Those instructions are not stackable, this means that the AI can receive only one instruction at the time.

System Break Down

Wheel System

Inputs:

- Player Input (E, or shortcuts (1,2,3,4))
    a. If E is being pressed
        - Player Input (Left mouse button over an instruction)
- Command/Instruction Selected

Process:

- Get the AI Reference

Output:

- Send the command selected to the AI reference

AI Control System

Inputs:

- Command/Instruction Selected

Process:

- Follow the Player
    a. Get a random point around the player's position
    b. Move the AI to the random position
- Stay
    a. Disables the AI's movement
- Roam
    1. Get a random point around the current position
    2. Move the AI to the random position
    3. If an orb gets into the AI's vision, the AI move to that position to collect it
    4. Else repeat the Rom step 3 after some seconds (exposed variable)
- Come and Wait

1. Get current player's position
2. Move the AI to the player's position
3. If point was reached disables the AI's movement (same as stay state)

Outputs:

- Execution of the instruction

Data/Variables

Wheel

- Instructions (Enum)

AI Controller

- Variables
    a. IsRoaming? (Bool)
    b. IsFollowingPlayer? (Bool)
    c. IsWaiting? (Bool)
- Data
    a. Roam Area = 500
    b. Chase Proximity = 150

Communication

- Instructions Wheel -> AI Controller
    - To send the instructions
- AI Controller <-> Orb
    - To collect orbs when they collide
- Character -> AI Controller
    - To get the character's position

Tool Assessment

- Unreal 5 AI Perception
- Unreal 5 HUD System
- I'll use my own movement system for the player's character

Required Systems/Components

- Character Controller
- HUD to control the instructions wheel

Validation Techniques

Functional Testing

- Follow Player
    - The AI moves after sending the instruction of following the player
    - Stops inside of the radius created around the player
    - If the player moves again, the AI must keep following it
- Stay
    - The AI must not move at all
- Roam
    - The AI moves around the place looking for orbs
    - If an orb gets into the AI's vision, the AI will move to the orb's position
    - Once there is a collision between the orb and the AI, the orb gets destroyed
    - If there aren't orbs in the AI's vision, the AI moves to a random position in a radius
    - The AI waits x seconds until move again
- Come and Wait
    - The AI moves to the point where the player was standing when sent the instruction
    - Once the AI reaches the point it will stop moving until receives another instruction

Debug System

- Orb Collection
    - Show a debug log of the current orbs collected once the AI hits one of them and the counter was increased.
- Track the AI behavior
    - Draw Line
    - Draw Circle

Assets Needed/Used

- All the assets needed are included in the Stack-O-Bot Project downloaded from the Unreal Marketplace, some of the materials will be modified to customize the project.

Implementation Tasks

1. Create the wheel blueprint
    a. Add instruction (Enum) variable
2. Create AI Controller blueprint
    a. Add Variables
        i. IsRoaming? (Bool)
        ii. canMove? (Bool)
        iii. isWaiting? (Bool)
3. Create the HUD
    a. Add the wheel to the UI
    b. Add orbs counter
4. Add logic to open the wheel
5. Add logic to send instructions to the AI Controller
    a. Instruction 0: Follow Player
    b. Instruction 1: Stay
    c. Instruction 3: Roam
    d. Instruction 4: Come and Stay
6. Add Follow the Player logic
    c. Enables the AI's movement
    d. Disables the Roam state
    e. Get current player's position
    f. Get a random point around the player's position
    g. Move the AI to the random position
7. Add Stay logic
    a. Disables the AI's movement
    b. Disables the Roam state
8. Add Roam logic
    a. Enables the AI's movement
    b. Enables the roam state
    c. Get a random point around the current position
    d. Move the AI to the random position
    e. If an orb gets into the AI's vision
        i. Move the AI to the orb's position
        ii. If they overlap/collide, destroy the orb and increase in 1 the orbs counter
    f. Else repeat the Roam step "c" after some seconds (exposed variable)
9. Add Come and Wait logic
    a. Enables the AI's movement
    b. Disables the Roam state
    c. Get current player's position
    d. Move the AI to the player's position
    e. If point was reached disables the AI's movement (same as stay state)

Research Used

- AI Perception : https://www.youtube.com/watch?v=AyGRqYzrv3A
- Instructions Wheel: https://www.youtube.com/watch?v=isxBg1cUTQY
- Behavior Trees: https://www.youtube.com/watch?v=iY1jnFvHgbE&t=1028s
- Enums: https://www.youtube.com/watch?v=s7MRfdmZ_UU