УДК 004.89:159.94

РОЗРОБКА АДАПТИВНОГО АЛГОРИТМУ ПОВЕДІНКИ NPC В ІГРАХ НА БАЗІ ІГРОВОГО РУШІЯ UNITY

Забродній О.В., студент
Київський національний університет технологій та дизайну
Корогод Г.О., кандидат технічних наук, доцент
Київський національний університет технологій та дизайну

Ключові слова: дерева поведінки, адаптивні алгоритми, психологічне розвантаження, штучний інтелект, ігровий рушій Unity.

У сучасному світі, де цифровізація та високий рівень стресу є великою проблемою, є актуальним пошук нових методів психоемоційної корекції. [1] Використання інтерактивних візуальних середовищ (відеоігри) дозволяє моделювати безпечні, веселі, а саме головне – без шкоди для оточуючих. Проте більшість сучасних ігрових персонажів (NPC) мають жорстко детерміновану поведінку, що призводить до швидкої адаптації, втрати фокусу та зацікавленості.

Метою даної роботи є розробка адаптивного алгоритму поведінки NPC, який імітує живу істоту з її непередбачуваністю,роблячи розважальний контент більш «людянішим».

Для розробки було обрано об'єктно-орієнтований підхід (ООП) [5] та ігровий рушій Unity. Для організації логіки поведінки NPC були використані архітектура ієрархічних дерев поведінки (BehaviorTree)[2] та логіка персонажа Джесі (Ворона). Вибір ієрархічних дерев обґрунтовано тим, що на відміну від лінійної структури алгоритму,дерево поведінки дозволяє створювати модульні та гнучкі структури, де кожна гілка відповідає за окрему дію, а вузол за умову.

Структурна схема розробленого алгоритму представлена на рис 1. Принцип роботи даної схеми полягає в наступному. Після ініціації обходу (Starttick) керування передається кореневому вузлу (Selector), який на основі пріоритетність дій визначає подальшу поведінку персонажу. Для цього кореневий вузол перевіряє три гілки подій:

1. Гілка підтримки: активація приємних аудіо та аудіовізуальних стимулів.
2. Гілка нейтральної автономності: імітація звичайної ворони (чищення пір'я, пошук їжі, спостереження за гравцем). Це створює ефект присутності та наповненості ігрового світу.
3. Гілка адаптації: виконання непередбачуваних дій, таких як: переміщення ігрових предметів, що виступає як контрольований стресовий фактор.

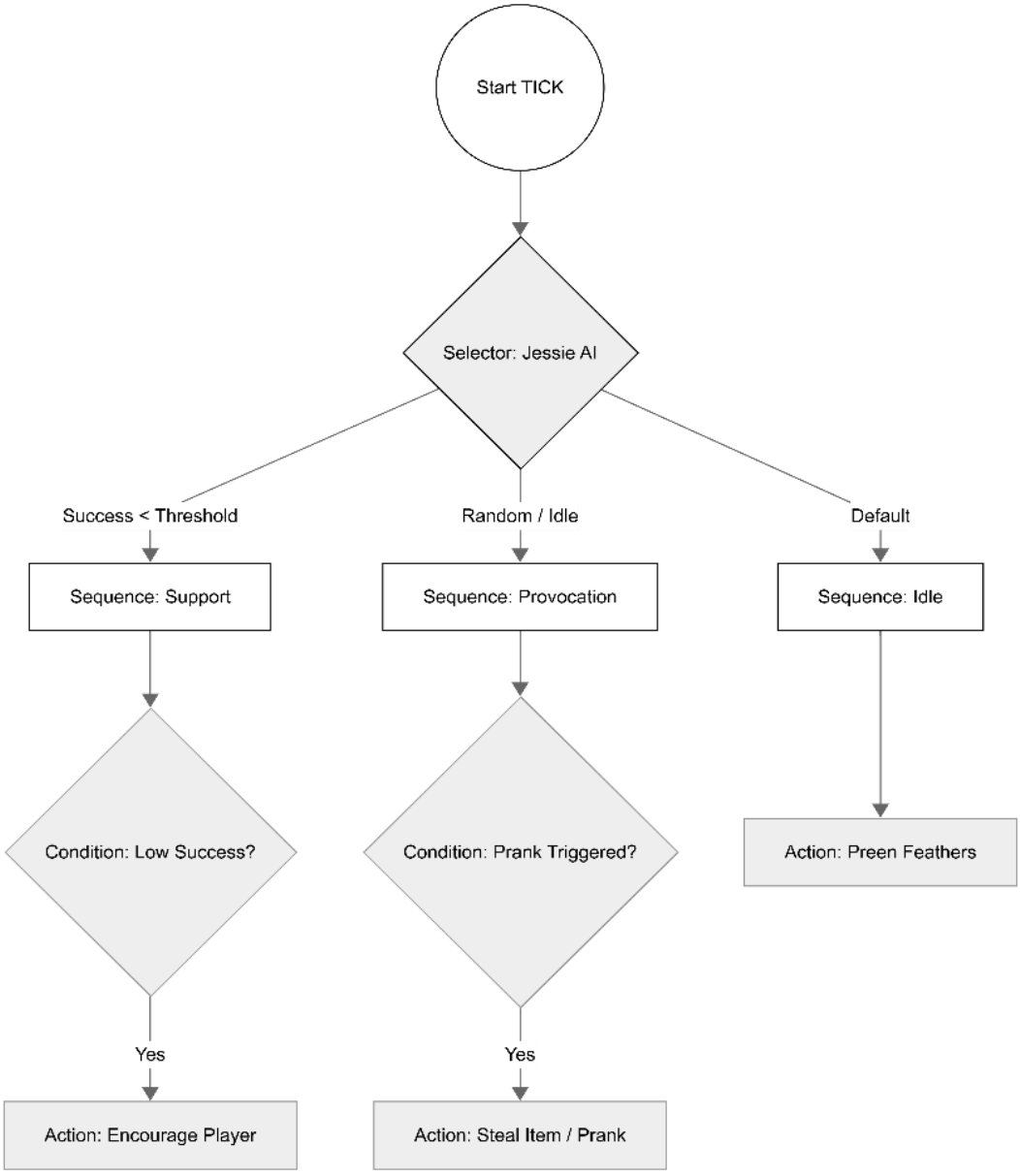

Рисунок 1 – Структурна схема дерева поведінки ігрового персонажа

Вузли типу Sequencета Selectorреалізують логіку обходу дерева, обробляючи три базові стани: Success, Failure, Running. Вузол Selectorвиконує обхід вузлів за принципом диз'юнкції. Натомість вузол Sequencефункціонує як кон'юнктор: він опитує вузли по черзі та припиняє обхід при отриманні статусу Failure. Стан Runningє критично важливим, оскільки дозволяє NPCвиконувати тривалі в часі дії без блокування основного потоку Unity.

Реалізація наслідування від базового класу Nodedозволяєвикористовувати патерн Композит, що забезпечує чистоту кодузгідно з принципами ООП [5].

Таким чином, логіка агента базується на зчитуванні параметрів стану (полів) користувача в класі agent.cs. Алгоритм аналізує швидкість взаємодії гравця з навколишнім світом, його рівень психологічної

підтримки та рівень спокою (визначається через частоту помилкових дій) [2, 3]. Основні параметри стану середовища наведені в табл.1

Таблиця 1

Основні параметри стану середовище

Назва змінної	Тип даних у C#	Опис призначення
playerSuccessRate	float	Коефіцієнт успішності дій гравця для визначення стресу
IdleTimer	float	Лічильник часу для активації нейтральної автономності або провокації гравця
targetItemPosition	Vector3	Просторові координати ігрового предмета.
isPrankReady	Bool	Прапор готовності до ініціації стресового фактора

На основі запропонованого алгоритму було розроблено відповідний програмний застосунок. Технічна реалізація виконана на мові C# у середовищі Unity. В програмі кожна дія агента реалізована окремо як окремий клас, що унаслідується від базового вузла Node. [3] Дані про стан світу зберігаються на дошці Blackboard (спільна пам'ять дерева поведінки), що забезпечує високу продуктивність та низькі витрати оперативної пам'яті замість того, щоб багато зберігати копій змінних або посилань.Проведене імітаційне моделювання показало, що впровадження нелінійного персонажа, який має патерни поведінки підбадьорення та провокації дозволило на 30% довше утримувати увагу гравця, а також змусити захотіти завантажити гру для того, щоб самому прожити ці моменти у грі. Це сприяє емоційній розрядці через залучення механізмів ігрової цікавості.

Розроблений алгоритм дозволяє перетворити нудного ігрового персонажа на більш цікавий і непередбачуваний, що, в цілому, сприяє покращенню настрою і психологічному розвантаженню [4, 5].

Список використаних джерел

1. Millington I. ArtificialIntelligenceforGames / I. Millington, J. Funge. – CRC Press, 2018. – 840 p.

2. Unity Technologies. BehaviorTreesinGame AI [Електронний ресурс]. – Режим доступу: https://docs.unity3d.com/

3.Colledanchise M. BehaviorTreesinRoboticsand AI: AnIntroduction / M. Colledanchise, P. Ögren. – CRC Press, 2018. – 206 p.

4. Dörner R. SeriousGames: Foundations, ConceptsandPractice / R. Dörner, S. Göbel, W. Effelsberg, J. Wiemeyer. – Springer, 2016. – 443 p.

5. Nystrom R. GameProgrammingPatterns / R. Nystrom. – GeneverBenning, 2014. – 354 p.