



УДК 004.383.4:004.9:519.832-021.131

ВИКОРИСТАННЯ СКІНЧЕННИХ АВТОМАТІВ ДЛЯ МОДЕЛЮВАННЯ ПОВЕДІНКИ ВІРТУАЛЬНИХ ПЕРСОНАЖІВ

USING FINITE STATE MACHINES TO SIMULATE THE VIRTUAL CHARACTERS BEHAVIOR

Шестопапов С.В.¹, Стариш Є.Г.²
Shestopalov S.V.¹, Starish Ye.H.²

¹Одеський національний технологічний університет, м. Одеса, Україна

²Одеський національний університет ім. І.І. Мечникова, м. Одеса, Україна

ORCID: ¹<http://orcid.org/0000-0001-8941-4610>

E-mail: ¹sshestopalov1984@gmail.com, ²evastarish@gmail.com

Copyright © 2025 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v%vi%i.3088

Анотація. Робота присвячена використанню скінченних автоматів для моделювання поведінки віртуальних персонажів в іграх. Дослідження та моделювання поведінки віртуальних персонажів має важливе значення для покращення ігрового досвіду. Це дозволяє створювати більш реалістичних та розумних персонажів, які здатні адаптуватися до дій гравця та навколишнього середовища. У цьому контексті пропонується використовувати скінченні автомати – математичні моделі із заданими станами і переходами між ними.

В роботі проаналізовано предметну область, дано визначення ключовим поняттям таким як «2D-Platformer», токен, матриця взаємодії, скінченний автомат.

Розроблено фрагменти проєктної документації гри жанру «2D-Platformer» з назвою «One Bite», які описують ігрові елементи. Ігрові елементи представлено у вигляді токенів. Розроблено матрицю взаємодії токенів, скінченні автомати головного героя гри та ворогів (віртуальних персонажів), за допомогою яких буде змодельовано їх поведінку.

Подальшим розвитком роботи є програмна реалізація скінченних автоматів головного героя та ворогів для розробки демонстраційної версії гри.

Abstract. The work is devoted to the use of finite state machines for simulation the virtual characters behavior in games. Research and simulation of the virtual characters behavior is great importance for improving the gaming experience. This allows to create more realistic and intelligent characters that can adapt to the actions of the player and the environment. In this context, it is proposed to use finite state machines - mathematical models with given states and transitions between them.

The paper analyzes the subject area, defines key concepts such as «2D-Platformer», token, interaction matrix, finite state machine.

Fragments of the project documentation of the «2D-Platformer» called «One Bite» were developed, describing the game elements. Game elements are presented in the form of tokens. The tokens interaction matrix, finite state machines of the main character of the game and enemies (virtual characters), with the help of which their behavior will be simulated, have been developed.

The further development of the work is the software implementation of finite state machines of the main character and enemies for the development of a demonstration version of the game.

Ключові слова: 2D-Platformer, токен, матриця взаємодії, скінченний автомат, віртуальний персонаж

Key words: 2D-Platformer, token, interaction matrix, finite state machine, virtual characters

Вступ

Комп'ютерні ігри стали невід'ємною частиною сучасної культури, надаючи мільйонам людей розвагу, можливість спілкування та, навіть, навчання. Створюються віртуальні світи, де гравці можуть поринути у різноманітні пригоди та випробувати нові емоції. Одним із ключових аспектів, що визначає захоплюючість ігрового процесу, є поведінка віртуальних персонажів, які складають навколишній світ гри та взаємодіють із



гравцем.

Існує багато ігрових жанрів з віртуальними персонажами. Один із найбільш популярних і найстаріших жанрів комп'ютерних ігор – це «2D-Platformer». У таких іграх гравці керують персонажами, переміщаючись різними рівнями, долаючи перешкоди і борючись з ворогами. Жанр «2D-Platformer» залишається актуальним та приваблює своєю динамічністю, цікавим геймплеєм та часто ностальгічним візуальним стилем, що асоціюється з класичними іграми минулих десятиліть. Відмінною рисою цього жанру є двомірна графіка та управління персонажем у двовимірному просторі.

Відомими представниками жанру «2D-Platformer» є ігри «Super Mario Bros» від Nintendo [1], «Super Meat Boy» від Team Meat [2], «Katana Zero» від Askiisoft та «Dead Cells» від Motion Twin.

Надалі будемо розглядати моделювання поведінки віртуальних персонажів власне для ігор жанру «2D-Platformer».

Дослідження та моделювання поведінки віртуальних персонажів має важливе значення для покращення ігрового досвіду. Це дозволяє створювати більш реалістичних та розумних персонажів, які здатні адаптуватися до дій гравця та навколишнього середовища. У цьому контексті пропонується використовувати скінченні автомати – математичні моделі із заданими станами і переходами між ними. Особливістю скінченних автоматів є їхня простота та ефективність при моделюванні складних поведінкових систем.

Актуальність теми зумовлена складністю розробки та підтримки логіки поведінки віртуальних персонажів у комп'ютерних іграх. Розробники стикаються з необхідністю створення численних умов та ланцюжків логіки, що регулюють поведінку персонажів. Це призводить до ускладнення коду та збільшення ймовірності виникнення помилок.

Відсутність чіткої структури управління станами персонажів робить процес налагодження та модифікації ігрової поведінки трудомістким та важким. Розробники змушені витратити більше часу на пошук та виправлення помилок, а також адаптацію поведінки персонажів під зміни в ігровому процесі.

Таким чином, потреба в ефективних інструментах для управління поведінкою віртуальних персонажів стає дедалі актуальнішою.

1. Аналіз предметної області

Спочатку дамо більш чітке визначення жанру «Platformer» і терміну «скінченний автомат» та розберемося з підходами та методами розробки скінченних автоматів для моделювання логіки поведінки віртуальних персонажів.

Жанр «Platformer» – «жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітним платформам (звідси й назва) та через перешкоди, збирання предметів, зазвичай необхідних для проходження рівня, знищення ворогів» [3].

Коріння жанру «Platformer» йде в аркадні ігри, в яких ігровий процес зосереджений на підйомі сходами. Ці ігри відбувалися на одному екрані і передбачали стрибки. Метою було або піднятися нагору, або перемогти всіх ворогів на екрані. Однією з таких ігор була «Space Panic», випущена 1980 року.

Однак широке визнання як першої платформної гри отримала «Donkey Kong», випущена компанією Nintendo у 1981 році. Мета гри – досягти самої верхньої платформи на екрані, ухиляючись від бочок, що котяться, стрибаючи через них. Популярність «Donkey Kong» підштовхнула до створення безлічі ігор, в яких акцент робився на механіці стрибків. У грі також вперше з'явився один із найзнаковіших ігрових персонажів – Маріо, який тоді називався Джампмен (Jumpman).

На початку свого розвитку платформери були представлені як ігри з одним екраном (Single Screen Platformers) [3]. Цей формат виник з обмежень технічних можливостей ранніх ігрових систем, які не дозволяли створювати великі ігрові світи із прокручуванням екрану. Ці ігри відображаються на одному екрані і зазвичай містять різноманітні перешкоди, які гравець повинен подолати, а також конкретну мету на кожному рівні. Прикладом такої гри є «Donkey Kong», де гравцеві потрібно дістатися до вершини екрану.

З розвитком технологій та можливостей ігрових систем з'явилася можливість створення більш складних та різноманітних рівнів. Це призвело до появи платформерів з прокручуванням екрану (Side and Vertical Scrolling Platformers) [3]. У таких іграх ігровий світ представлений у вигляді великої області, що прокручується у міру просування персонажа. Прикладами таких ігор є «Super Mario Bros» та «Castlevania».

З часом жанр «Platformer» розвивався і розширювався, породжуючи різні піджанри. Через п'ять місяців після виходу «Donkey Kong» було випущено ще одну впливову платформерну гру під назвою «Jump Bug», де крім бігу і стрибків гравець міг стріляти в віртуальних персонажів. Це надихнуло створити один із ранніх піджанрів «gun-and-gun platformers», який став популярним завдяки серіям ігор «Contra». Іншим піджанром став «puzzle platformer», що поєднує елементи головоломок зі стрибками по платформах. The «Lost Vikings» є гарним прикладом такої гри.

У середині 1980-х років відеоігрові консолі зазнавали занепад, і вважалося, що вони повністю пішли зі сцени, поки не було випущено «Super Mario Bros» у 1985 році. Вона стала однією із найбільш продаваних відеоігор в історії, ініціюючи золоте століття платформерів.

Сьогодні ігри жанру «Platformer» діляться на безліч різновидів, включаючи піджанри «2D-Platformer» і «3D-Platformer». Основною відмінністю є кількість вимірів, у яких переміщається головний персонаж: в іграх жанру «2D-Platformer» персонаж переміщається у двох площинах, а в іграх жанру «3D-Platformer» – трьох площинах.



Популярними прикладами ігор жанру «2D-Platformer» є «Cuphead», «Gris», «Celesto», «Spelunky 2» та «Hollow Knight».

Прикладами ігор жанру «3D-Platformer» є «Crash Bandicoot 4: It's About Time», «Demon Turf», «It Takes Two», «Project Feline» та «A Hat in Time».

Важливим аспектом ігор жанру «2D-Platformer» є віртуальні персонажі, які можуть бути або противниками, або помічниками, з якими можна поспілкуватися та дізнатися про лор ігрового світу.

Для моделювання поведінки віртуальних персонажів пропонується використати теорію скінченних автоматів. Скінченний автомат визначає різні стани, в яких може перебувати персонаж, та переходи між цими станами залежно від зовнішніх умов та дій гравця. Цей інструмент дозволяє ефективно керувати діями персонажа залежно від гравця. З іншого боку, скінченний автомат може визначати стратегії поведінки для неігрових персонажів. Використання скінченного автомата сприяє створенню чіткої та структурованої логіки поведінки, що робить процес розробки та підтримки поведінки персонажів більш простим та ефективним.

Розглянемо базові поняття, необхідні для використання скінченних автоматів. Спочатку гру необхідно представити у вигляді набору базових елементів – токенів. В ігровому контексті токени є дискретними елементами, які пропонується реалізувати в грі [4]. Токени охоплюють широкий спектр об'єктів, включаючи персонажів, предмети, ресурси, а також абстрактні концепції. Токени організуються в ієрархічну структуру, де ігровий світ займає верхній рівень ієрархії, а кожен токен є частиною цього світу.

Далі необхідно визначити тип події, яка виникає при взаємодії токенів (наприклад, досить часто це може бути подія зіткнення – collision event) [5, 6].

Взаємодію токенів можна представити за допомогою матриці. Матриця взаємодії токенів (token interaction matrix) – це діаграма, в якій відображені всі види взаємодій (або ключові взаємодії), які можуть виникнути в грі [5,6]. Представляючи собою таблицю, у якій перераховані всі токени як по вертикалі, так і по горизонталі, матриця взаємодій дозволяє аналізувати, які токени можуть взаємодіяти між собою, і які події можуть статися у результаті цієї взаємодії. Важливо відзначити, що зазвичай (не завжди) токени не взаємодіють самі з собою. У дуже великих іграх матриці взаємодії токенів безпосередньо не використовуються. Замість них вводиться додатковий рівень абстракції, що описує взаємодію токенів і властивостей (token-property matrix), а також взаємодію властивостей із властивостями (property-property matrix). Приклад реалізації матриці взаємодії токенів для всім відомої гри «Pac-Man» зображено на рисунку 1 [5, 6].

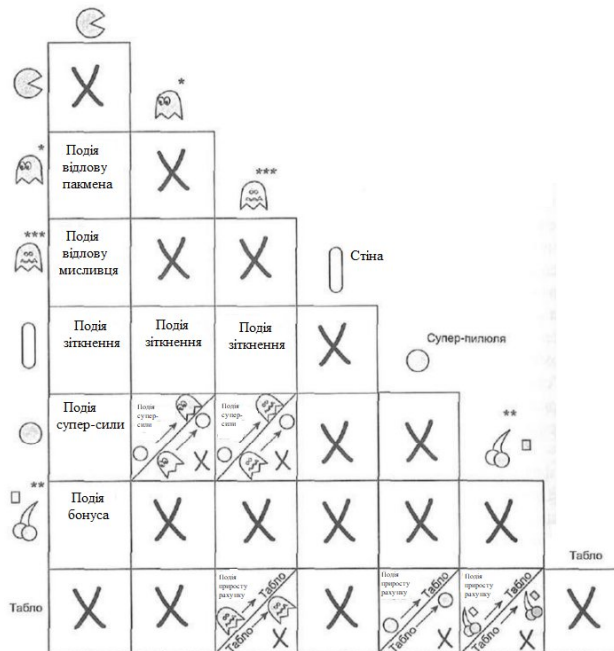


Рис. 1 – Приклад матриці взаємодії токенів для гри «Pac-Man»
Fig. 1 – Example of a token interaction matrix for the game “Pac-Man”

Як видно з рисунку 1, існує два види взаємодії між токенами: симетрична та асиметрична. Симетричними є взаємодії, що спрямовані однаково для обох токенів, та представлені у вигляді звичайних квадратів. Наприклад, в даній матриці симетричними взаємодіями є взаємодії між перешкодою та персонажем гравця, землею та персонажем гравця, і землею та супротивником.

Асиметричними взаємодіями є ті, що змінюються в залежності від обраного напрямку взаємодії. Така взаємодія зображена у вигляді квадрату, поділеного на два трикутника, кожен з яких представляє різну взаємодію. Матриця, представлена на рисунку 1, має декілька асиметричних взаємодій.



Матриця взаємодій допомагає побачити загалом систему зв'язків, що виникають між токенами, і передбачити можливі події у грі. Однак для великих ігор матриця може виявитися недостатньою, тому зазвичай розглядається скінченний автомат окремих токенів.

Якщо гра досить проста (наприклад, всім відома гра «Pong») і події в ній впливають на токени миттєво і не переводять жоден з них у різні стани то маємо чисту модель керовану подіями (E-model – event-based model) [5, 6]. Для такої гри скінченні автомати не потрібні. Більше того їх неможливо розробити. Однак, якщо події, які впливають на токени, можуть змінювати стан останніх (наприклад, гра «Pac-Man»), то необхідно переходити до моделі, керованої станами та подіями (ES-model – event and state-based model) [5, 6]. Власне для ігор такої складності використання скінченних автоматів є дуже інформативним та необхідним.

Скінченний автомат або машина станів (FSM – finite state machine, state machine) – це математична модель, яка може перебувати в одному із скінченної кількості станів у будь-який момент часу. Скінченний автомат може переходити з одного стану в інший у відповідь на деякі вхідні дані [7]. Скінченний автомат не залишає місця для різних тлумачень та неоднозначностей, які можливі при використанні матриці взаємодій. Кожен окремий стан токена, для якого побудований скінченний автомат, представляється прямокутником. Події позначаються колами. Лінії, що йдуть від подій до станів, позначають вплив події на відповідний стан. Невеликі кола, якими закінчуються такі лінії, являють собою маркери подій, які використовуються в тих випадках, коли на один і той же стан може впливати кілька подій. Зазвичай подія призводить до переходу стану, який позначається лінією, що з'єднує початковий і кінцевий стан. У тих випадках, коли у певний стан можуть виконуватися переходи з кількох станів, лінія переходу, що відповідає певній події, яка спричинила цей перехід, закінчується маркером того ж кольору, що й у маркера лінії події, що впливала на вихідний стан. Маркірувати можна не тільки різними кольорами а й цифрами чи літерами. Круговою стрілкою позначається «міні-подія», тобто подія, яка не призводить до переходу стану. Це означає, що, якщо відбувається відповідна подія, стан просто генерується знову, навіть якщо токен вже знаходиться в цьому стані. На рисунку 2 в якості прикладу представлений скінченний автомат для мисливців з гри «Pac-Man» [5, 6].

Як видно із рисунку 2, схему взаємодій у вигляді скінченного автомата набагато простіше зрозуміти.

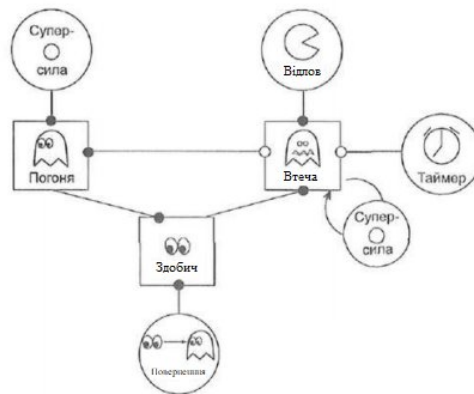


Рис. 2 – Скінченний автомат мисливців гри «Pac-Man»
Fig. 2 – Finite automaton of hunters of the game “Pac-Man”

Слід зауважити, що на рисунку 2 не було показано таких подій, як зіткнення зі стінами лабіринту. Їх не показано, оскільки ці події є більш універсальними, характерними для всіх об'єктів. На рисунку також відображено не всі події, на які реагують мисливці. Наприклад, коли мисливець переходить у стан здобичі, він генерує подію приросту рахунку. Це зроблено через те, що досить часто доцільно обмежити скінченний автомат лише вхідними подіями. Відображення вихідних подій не внесе ясності в ігрову механіку для токенів, що представляють у грі мисливців, але при цьому призведе до захарачення схеми скінченного автомата.

Мисливець може перебувати в одному з трьох станів: «Погоня», «Втеча» і «Здобич».

Спочатку розглянемо стан погоні. Цей стан виникає, коли пакмен з'їдає супер-пілюлю, внаслідок чого настає подія супер сили. Цій події на рисунку 2 відповідають лінії з чорним маркером, тому шукаємо в скінченному автоматі лінію переходу, яка виходить зі стану «погоня» і починається чорним маркером. Така лінія приводить у стан «втеча». Інша лінія переходу не має маркера, з чого випливає, що ця лінія переходу є вхідною.

Стан втечі дещо складніший. У цьому стані на мисливця може впливати два види подій: подія таймера, яка виникає після завершення дії супер-пілюлі (має білий маркер), а також подія вилову мисливця пакменом (чорний маркер). Простеживши перехід, що виникає після події таймера, видно, що у разі її виникнення мисливець повертається в стан погоні. Подія ж вилову переводить мисливця у стан «здобич».

Кругова стрілка показує, що станеться у тому випадку, якщо пакмен з'їсть супер-пілюлю в той момент, коли мисливець вже перебуває в стані втечі. Як видно із схеми скінченного автомата, це призведе до скидання таймера в нуль, після чого час перебування мисливця у стані втечі продовжиться.



Стан здобичі досить простий, оскільки на нього впливають лише події одного виду. Після того, як мисливець стає здобиччю пакмена, він вирушає в мисливський будиночок. Як тільки мисливець дістається до цього будиночка, він отримує подію повернення, що знову переводить його в стан погоні.

Цей скінченний автомат має замкнуту структуру. Інакше кажучи, він немає тупикових гілок. Тобто таких станів, із яких мисливець неспроможний вийти. Якщо порівняти скінченний автомат мисливця з скінченним автоматом пакмена, представленим на рисунку 3, видно, що в останньому випадку тупикові гілки є [5, 6].

Скінченний автомат пакмена дуже нагадує скінченний автомат мисливців. Відмінність його полягає лише в тому, що кількість подій повернення в гру для пакмена обмежена – коли лічильник життів пакмена обнуляється, гра закінчується. Таким чином, у цьому випадку маємо справу з розімкненим скінченним автоматом, оскільки в грі можна досягти стану, з якого не можна вийти. По суті, цей скінченний автомат являє собою всю гру, оскільки події, на які він реагує, обробляються також ігровим світом. Наприклад, коли пакмен втрачає чергове життя, його скінченний автомат генерує подію скидання рівня, а коли лічильник життя обнуляється, – подію завершення гри.

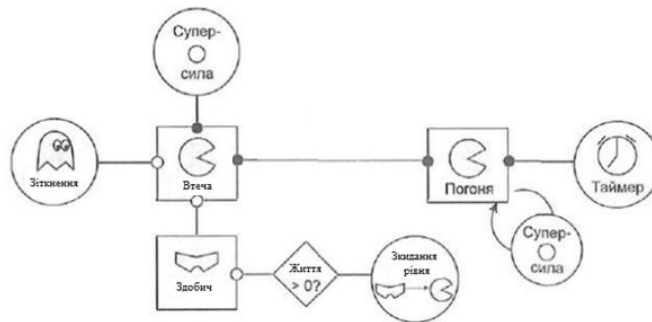


Рис. 3 – Скінченний автомат пакмена

Fig. 3 – Pacman finite automaton

Для ще складніших ігор, де для опису треба крім токенів, подій та станів токенів враховувати ще властивості токенів використовуються моделі, що містять події, стани, властивості та переходи (ESPT-model – events, states, properties and transitions model). Фрагменти скінченних автоматів для гарячого металевого шару та сніжного шару з зазначенням властивостей для гри «Balls!» (ESPT-model) представлено на рисунку 4 [5, 6].

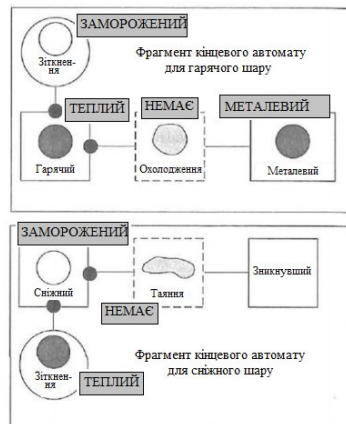


Рис. 4 – Фрагменти скінченних автоматів для гарячого металевого шару та сніжного шару з зазначенням властивостей

Fig. 4 – Fragments of finite state machines for a hot metal layer and a snow layer with an indication of the properties

Перехідні (проміжні) стани показані прямокутниками, що зображені штриховими лініями. Як видно з рисунку 4, одночасно на скінченних автоматах відображені стани токенів, події, які впливають на стани, властивості токенів і переходи. Такі скінченні автомати є досить складними для «читання» та розуміння розробниками гри.

Для моделювання логіки поведінки віртуальних персонажів гри жанру «2D-Platformer» доцільно використовувати ES-model та її скінченні автомати.

Мета і завдання

Метою роботи є розробка скінченних автоматів для моделювання поведінки віртуальних персонажів в іграх жанру «2D-Platformer».

Основними задачами, які необхідно вирішити в ході роботи є:

1. Розробити фрагменти проектної документації, які описують ігрові елементи та представити їх у вигляді токенів.
2. Розробити матрицю взаємодії токенів.



3. Розробити скінченні автомати головного персонажу гри та ворогів (віртуальних персонажів), за допомогою яких буде змодельовано їх поведінку.

2. Розробка фрагментів проєктної документації, матриці взаємодії та скінченних автоматів

2.1. Розробка фрагментів проєктної документації

Пропонується спроектувати та розробити фрагменти проєктної документації для гри жанру «2D-Platformer» з назвою «One Bite». Для цієї гри розробити матрицю взаємодії токенів, скінченні автомати головного персонажу гри та ворогів (віртуальних персонажів).

Опишемо коротко суть гри. Головний герой «One Bite», Морган Ноктфорд, у відчайдушній гонитві від переслідувачів знаходить притулок у таємничому замку. В середині замку головного героя зустрічає великий лорд-вампір Вельхеор, господар цього замку, і кусає його. Прокинувшись в одній із темних тюремних камер замку, Морган виявляє в собі вампірські здібності і вирішує знайти лорда-вампіра, щоб помститися. Основна мета гри полягає у проходженні рівнів шляхом використання вампірських здібностей Моргана, для розвитку яких потрібні ресурси з уражених ворогів. Ігрові рівні дають можливість проходити гру різними шляхами та наділяють гравця бонусами, такими як відновлення здоров'я та розблокування здібностей за дослідження. Дизайн рівнів представлений у вигляді різних кімнат замку, наповнених ворогами, що служать лорду-вампіру. У деяких із них очікуються зіткнення з потужними босами, які потребують особливої тактики бою. Гравцеві також належить враховувати джерела світла, які є додатковим елементом перешкоди. При попаданні в область світла, гравець ризикує бути виявленим ворогами, тому йому вміло необхідно використовувати вампірські здібності, щоб ховатися в тіні та уникати прямого контакту з джерелами світла.

Остаточний зовнішній вигляд головного героя представлено на рисунку 5.



Рис. 5 – Головний герой
Fig. 5 – The main character

Головний герой має основні навички переміщення, типові для платформерів: він може бігати, стрибати по платформах і лазити по драбині. Серед атакуючих умінь головний герой має базову атаку, яка є махом кігтями. За допомогою цього вміння гравець може постійно завдавати шкоди ворогам. Вампірські здібності головного героя представлені у вигляді дерева вмінь, що має три основні гілки, які може розвивати гравець: «Магія крові», «Володіння кігтями» та «Приховані тіні». Ці гілки здібностей надають можливість гравцеві розвинути свого персонажа, як мага, воїна та вбивцю відповідно. Однак гравець може розвиватися не тільки в одній гілці, а ще дозволяється комбінувати здібності, створюючи власний білд. Схема дерева вмінь «Вампірські здібності» зображена на рисунку 6.



Рис. 6 – Дерево вмінь «Вампірські здібності»
Fig. 6 – Skill tree “Vampire abilities”

З самого початку ігрового процесу гравцеві доступна перша вампірська здібність «Голодний укус», з якої розгалужуються основні гілки дерева вмінь. Ця здатність дозволяє заповнювати запаси зарядів крові, за допомогою яких гравцеві можна використовувати здібності з гілки «Магія крові». Також це вміння дозволяє завдавати більше шкоди, ніж базова атака головного героя. Однак після активації деяких вампірських здібностей,







у тому числі здібності «Голодний укус», потрібен деякий час на їх відновлення, перш ніж вони знову стануть доступними для використання. Надалі такий час називається перезарядженням вміння. З точки зору балансу гри, це зроблено для того, щоб гравець не міг використовувати сильні атакуючі вміння на постійній основі. У проєктованій демонстраційній версії гри гравцеві доступно, крім вампірської здібності «Голодний укус», відразу ще перші здібності «Гострі голки» та «Тіньовий крок» з основних гілок дерева. Здібність «Гострі голки» дозволяє головному персонажу кидати у противника гострі голки, завдаючи помірної шкоди. Голки проникають у ціль і залишаються в ній на деякий час, завдаючи додаткових втрат за рахунок кровотечі. Здібність «Тіньовий крок» дозволяє персонажу різко переміститися в тінь, оминаючи джерела світла та удари від ворогів. Ця здібність дозволяє уникати виявлення ворогами та використовувати перевагу несподіванки. Описані базові вміння та навички будуть враховані при формуванні станів скінченного автомату головного героя.

Характеристики головного героя включають дві основні шкали. Перша шкала відображає поточну та максимальну кількість здоров'я головного героя. Друга шкала є кількістю зарядів крові, необхідних для використання вампірських здібностей з гілки «Магія крові».

Система зарядів крові спроектована з урахуванням потреби збалансованої гри. Вона забезпечує обмеження використання дальніх атак, що сприяє підтримці балансу між дальнім та ближнім боєм. Для гравців, які вибрали білд мага, такий підхід створює додаткові умови стратегії. У цьому випадку гравець повинен розумно розподіляти свої ресурси, іноді вдаючись до ближнього бою за допомогою здібності «Голодний укус», щоб відновити заряди крові.

Таблиця 1 – Список ворогів та їхні характеристики

| Тип ворога | Загальна характеристика |
|--|---|
| Кістлявий вовк  | Захищений обладунками із кісток своїх родичів. Атакує головного героя на ближній дистанції. |
| Вампір-лучник  | Нижчий вампір, одягнений у червоний плащ. Використовує зачарований лук для бою на далекій дистанції. |
| Вампір-стражник  | Нижчий вампір, одягнений у червону броню. Використовує меч із щитом для кількох атак та захисту, відповідно, на ближній дистанції. |
| Кажан  | Маленький житель замку, сплячий на стелях та атакуючий головного героя при його наближенні на ближній дистанції. |
| Палаючий дух  | Одна з жертв лорда-вампіра, яка не може знайти спокою. Коли головний герой наближається до духу, то ворог починає переслідувати його та вибухає на ближній дистанції. |

Вороги відіграють ключову роль у створенні динаміки ігрового процесу. Вони не тільки є перешкодами на шляху головного героя, а й служать важливим елементом у поживленні ігрового світу.

У грі присутнє різноманіття ворожих персонажів, кожен з яких має унікальні характеристики та поведінку. У демонстраційній версії гри пропонується п'ять типів ворогів. Список запропонованих типів ворогів та їхні характеристики наведено у таблиці 1.



У пізніх локаціях гравцеві будуть зустрічатися складніші вороги зі зміненими характеристиками: зі збільшеною кількістю здоров'я і шкоди, що наноситься. Завдяки перезарядці ворожого вміння з'являються вільні проміжки часу, в яких ворог не робить жодних дій і гравець має можливість контратакувати.




Усі вороги мають схожу поведінку – вони реагують на зустрічі з головним героєм. Для цього вони мають зону виявлення, яка може відрізнятись як за розміром, так і формою залежно від типу ворога. Наприклад, вороги з ближньою атакою можуть мати більш обмежену зону виявлення, тоді як вороги з дальньою атакою мають ширший радіус видимості.

Вороги, які мають ближню атаку, мають додаткову область ідентифікації гравця, при контакті з якою вони починають завдавати шкоди. У свою чергу, вороги з дальньою атакою починають атакувати гравця після виявлення його у своїй області видимості.

Всі описані характеристики, навички та вміння ворогів будуть враховані при формуванні станів скінченних автоматів кожного типу ворога.

На формування матриці взаємодії та скінченних автоматів впливають ігрові бонуси. Ігрові бонуси є додатковими елементами у грі, які розширюють можливості головного героя, покращують ігровий досвід та стимулюють до дослідження ігрового світу. Вони також є способом балансування ігрового процесу та можуть змінюватися в залежності від рівня складності гри. Види ігрових бонусів наведено в таблиці 2.

Таблиця 2 – Список ігрових бонусів та їхні характеристики

| Вид бонусу | Загальна характеристика |
|---|---|
|  Очки досвіду | Випадають із знищених ворогів. Дозволяють отримати новий рівень головного героя. При підвищенні рівня характеристики також збільшуються |
|  Фрагмент душі | Випадає із знищених ворогів. Надає доступ до нової вампірської здібності в дереві вмінь. |
|  Пляшка крові | Випадає із знищених ворогів. Відновлює здоров'я гравця на певний відсоток. |

2.2. Формування матриці взаємодії токенів

Для формування матриці взаємодії спочатку треба визначити токени. Токенами для проєктованої та розробленої гри «One Bite» є головний герой, ворог, перешкода, ігровий бонус та платформа. Детальну чітку матрицю взаємодії для гри такої складності розробити практично неможливо.

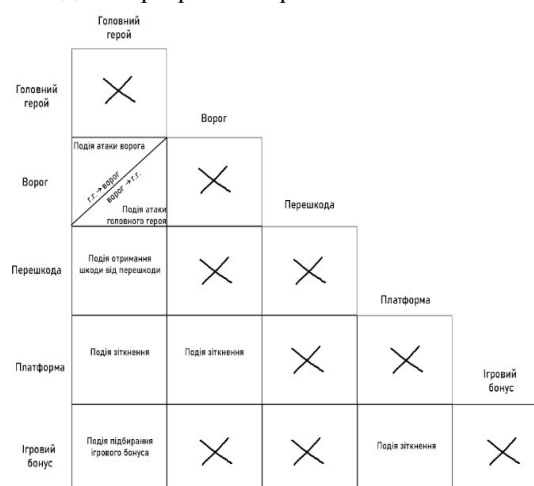


Рис. 7 – Узагальнена матриця взаємодій токенів

Fig. 7 – Generalized token interaction matrix

Узагальнена спрощена матриця взаємодій токенів гри «One Bite» зображена на рисунку 7.



В представленій матриці взаємодій можна побачити, що головний герой може атакувати ворогів, збирати ігрові бонуси та отримати шкоду від перешкод. У той же час, вороги не взаємодіють з ігровими бонусами. Взаємодія між ворогами та головним героєм асиметрична.

2.3. Розробка скінченних автоматів віртуальних персонажів

Токенами, що мають чисельну кількість станів та представляють собою віртуальних персонажів, є головний герой та вороги. Скінченні автомати представляють собою важливий механізм управління поведінкою віртуальних персонажів.

Спочатку розробимо скінченний автомат головного героя. Оскільки повна схема цього скінченного автомату занадто громіздка, вирішено зобразити її в більш спрощеній формі – без умов переходу між станами. Спрощену схему скінченного автомату головного героя наведено на рисунку 8. Даний скінченний автомат описує можливу поведінку головного героя з наступною кінцевою кількістю станів: «Спокій», «Біг», «Стрибок», «Використання драбини», «Базова атака», «Голодний укус», «Гострі голки», «Тіньовий крок», «Відновлення здоров'я» та «Смерть». Початковим станом скінченного автомату є «Спокій», а кінцевим станом – «Смерть».

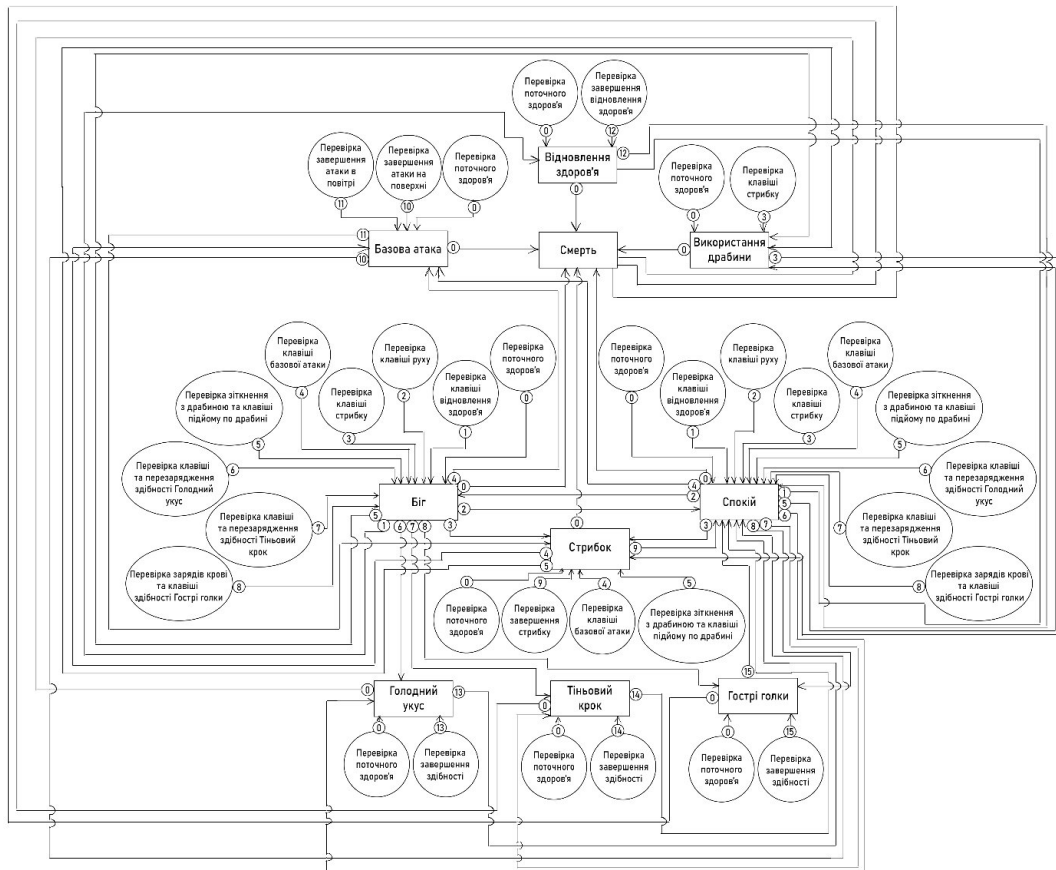


Рис. 8 – Спрощена схема скінченного автомата головного героя
Fig. 8 – Simplified diagram of the main character's finite automaton

Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка клавiші відновлення здоров'я», «Перевірка клавiші стрибку», «Перевірка клавiші базової атаки», «Перевірка зіткнення з драбиною та клавiші підйому по драбині», «Перевірка клавiші і перезарядження здібності Голодний укус», «Перевірка клавiші та перезарядження здібності Тіньовий крок», «Перевірка зарядів крові та клавiші здібності Гострі голки», «Перевірка завершення стрибку», «Перевірка завершення атаки на поверхні», «Перевірка завершення атаки в повітрі», «Перевірка завершення відновлення здоров'я», «Перевірка завершення здібності Голодний укус», «Перевірка завершення стану Тіньовий крок», «Перевірка завершення здібності Гострі голки».

Коли клавiша руху натиснута, головний герой переходить зі стану «Спокій» у стан «Біг». Так само й навпаки: якщо головний герой знаходиться у стані «Біг» і клавiша не є натиснутою, головний герой переходить до стану «Спокій». Якщо головний герой знаходиться у стані «Спокій» або «Біг», і натиснута клавiша, відповідальна за стрибок, – то відбувається перехід до стану «Стрибок» і повернення назад до стану «Спокій», коли стан «Стрибок» закінчується. У той момент, коли головний герой зіткнувся з драбиною і натиснув клавiшу, відповідальну за підйом по драбині, відбувається перехід зі стану «Спокій» або «Біг» до стану «Використання драбини». Для виходу з цього стану необхідно натиснути клавiшу стрибку і в цей момент відбувається перехід до стану «Стрибок». До стану «Базова атака» можна перейти з трьох станів, а саме: «Спокій», «Біг» і «Стрибок», натисненням клавiші, яка відповідає за базову атаку. Після закінчення цього стану головний герой повертається у попередній стан відносно стану «Базова атака». Перехід до стану «Голодний укус» або «Тіньовий крок»



відбувається у той момент, коли перезарядження вміння скінчилося та натиснута відповідна клавіша. Такий перехід можливий зі станів «Спокій» та «Біг» та при завершенні стану «Голодний укус» головний герой повертається до попереднього стану. Якщо в головного героя є в наявності необхідна кількість зарядів крові та натиснута відповідна клавіша, герой переходить до стану «Гострі голки» зі станів «Спокій» або «Біг». Якщо у головного героя кількість поточного здоров'я дорівнює нулю, відбувається перехід до стану «Смерть» з будь-якого з інших можливих станів. В якості маркерів в запропонованому скінченному автоматі використовуються цифри. Скінченний автомат головного героя є дуже складним по конструкції, тому що в ньому міститься найбільша кількість станів порівняно з ворогами. Зі збільшенням числа станів розмір скінченного автомата значно зростає, а кількість переходів між ними збільшується. Однак, саме завдяки такій деталізації можна легко розглянути логіку управління поведінкою головного героя в ігровому світі.

Далі розробляються скінченні автомати ворогів. Першим ворогом є Кістлявий вовк.

Скінченний автомат ворога Кістлявий вовк описує його поведінку з наступною кінцевою кількістю станів: «Патрулювання», «Переслідування», «Спокій», «Ближня атака» та «Смерть». Початковим станом є «Патрулювання», а кінцевим – «Смерть».

Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка виявлення головного героя в області видимості», «Перевірка виявлення головного героя в області атаки», «Перевірка перезарядження атаки» та «Перевірка завершення атаки».

Схему скінченного автомату ворога Кістлявий вовк зображено на рисунку 9.

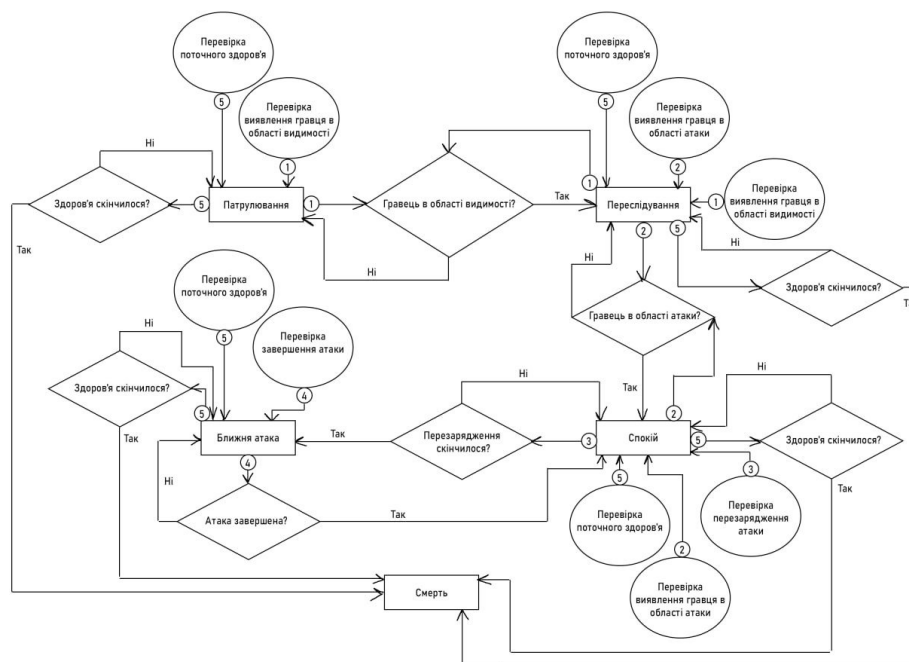


Рис. 9 – Схеми скінченного автомату ворога Кістлявий вовк
Fig. 9 – Schematic of the finite automaton of the enemy Bony Wolf

Коли головний герой потрапляє в область видимості ворога, то Кістлявий вовк змінює свій стан з «Патрулювання» на «Переслідування». А якщо головний герой виходить з цієї зони, то Кістлявий вовк навпаки змінює стан з «Переслідування» на «Патрулювання». При виявленні головного героя в області атаки під час стану «Переслідування» стан змінюється на «Спокій». Якщо головний герой вийшов з області атаки, то відбувається перехід зі стану «Спокій» в стан «Переслідування». Якщо поточний стан «Спокій» та перезарядження ворожої атаки скінчилося, то стан змінюється на «Ближню атаку». При завершенні атаки стан скінченного автомату повертається на стан «Спокій». Якщо кількість поточного здоров'я Кістлявого вовку дорівнює нулю, то з будь-якого стану відбувається перехід до стану «Смерть».

Додавання стану «Спокій» у скінченний автомат ворогів, включаючи ворога Кістлявий вовк, обумовлено необхідністю представлення нейтральної поведінки ворога в очікуванні можливості атаки. Хоча ворог може бути на потрібній відстані для атаки, але його здатність атакувати може бути на перезарядці. У стані «Спокій» ворог очікує, доки його здатність атаки не буде готова до використання, надаючи гравцеві можливість контратакувати. Таким чином, цей стан дає гравцеві час на реагування, а також зрівнює можливості гравця та ворога.

Наступний скінченний автомат, який розробляється, – є автомат ворога Вампір-лучник. Схему скінченного автомату ворога Вампір-лучник зображено на рисунку 10. Даний скінченний автомат описує поведінку ворога Вампір-лучник з наступною кінцевою кількістю станів: «Патрулювання», «Спокій», «Відскок», «Дальня атака», «Смерть». Початковим станом серед них є «Патрулювання», а кінцевим – «Смерть». Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка виявлення головного героя в області видимості»,



«Перевірка можливості виконання відскоку», «Перевірка перезарядження атаки», «Перевірка завершення відскоку» та «Перевірка завершення атаки».

Коли головний герой потрапляє в область видимості Вампіра-лучника, то відбувається перехід зі стану «Патрулювання» у стан «Спокій». Якщо головний герой вийшов з області видимості, стан змінюється зі «Спокій» в «Патрулювання». Зі стану «Спокій» Вампір-лучник може перейти до стану «Дальня атака» або стану «Відскок» за наступними умовами: в «Дальню атаку», якщо перезарядження атаки скінчилося, а в «Відскок» – якщо головний герой увійшов у область відскоку, перезарядження дальньої атаки йде та перезарядження відскоку скінчилося. Назад зі станів вмінь «Дальня атака» та «Відскок» відбувається перехід, коли стан завершено. Якщо кількість поточного здоров'я Вампіра-лучника дорівнює нулю, то з будь-якого стану відбувається перехід до стану «Смерть».

Особливу увагу слід приділити умові переходу до стану «Відскок». Оскільки обидві умови переходу, пов'язані з перезарядженням атаки та відскоку, можуть завершитися одночасно, необхідно встановити пріоритет між ними. Шляхом детального аналізу умов визначено, що при одночасному закінченні перезарядки вмінь «Дальня атака» та «Відскок», першочергово виконується «Дальня атака», а не «Відскок».

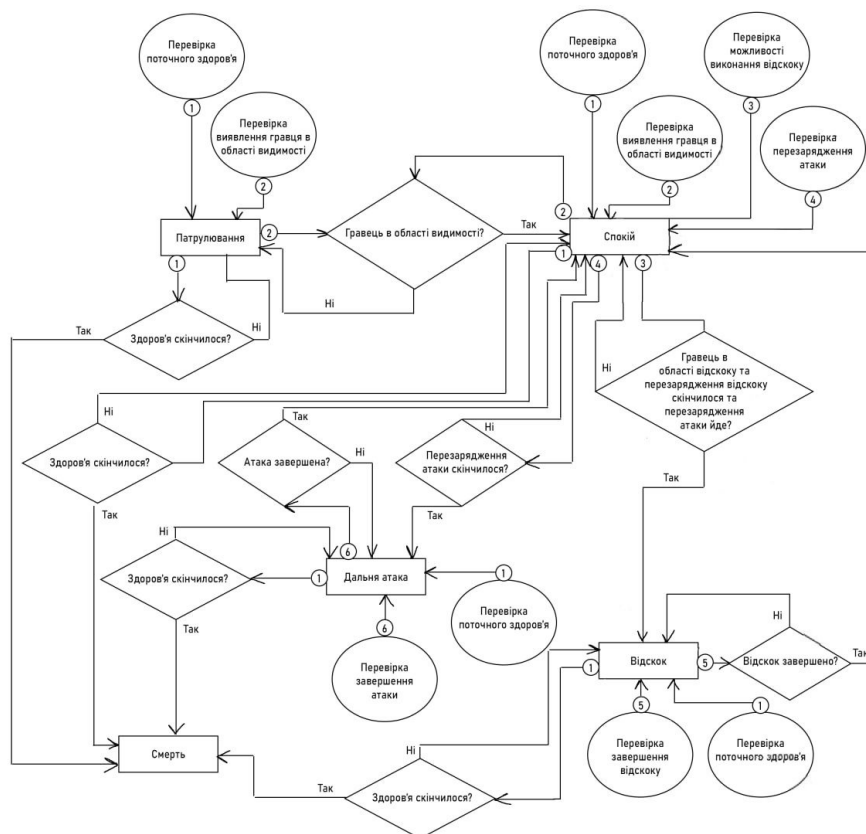


Рис. 10 – Схема скінченного автомату ворога Вампір-лучник
Fig. 10 – Schematic of the final enemy machine gun Vampire Archer

Далі розробляється скінченний автомат ворога Вампір-стражник. Схему скінченного автомату ворога Вампір-стражник зображено на рисунку 11.

Даний скінченний автомат описує поведінку ворога Вампір-стражник з наступною кінцевою кількістю станів: «Патрулювання», «Переслідування», «Спокій», «Ближня атака», «Захист щитом» та «Смерть». Початковим станом серед них є «Патрулювання», а кінцевим – «Смерть».

Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка наявності головного герою у області видимості», «Перевірка наявності головного герою у області атаки», «Перевірка можливості виконання атаки», «Перевірка перезарядження захисту», «Перевірка завершення атаки» та «Перевірка завершення захисту».

Коли головний герой потрапляє в область видимості, Вампір-стражник змінює свій стан з «Патрулювання» на «Переслідування». Якщо головний герой виходить з цієї зони, Вампір-стражник навпаки змінює стан з «Переслідування» на «Патрулювання». При виявленні головного героя в області атаки під час стану «Переслідування», стан змінюється на «Спокій». Але якщо головний герой вийшов з даної області, відбувається перехід зі стану «Спокій» у стан «Переслідування». Зі стану «Спокій» Вампір-стражник може перейти до стану «Ближня атака» або стану «Захист щитом» за такими умовами: в «Ближню атаку», якщо перезарядження захисту йде та перезарядження ближньої атаки скінчилося, а в «Захист щитом», якщо перезарядження захисту



скінчилося. Назад зі станів вмінь «Ближня атака» та «Захист щитом» відбувається перехід до «Спокою», коли стан завершено. Якщо кількість поточного здоров'я Вампіра-стражника дорівнює нулю, то з будь-якого стану відбувається перехід до стану «Смерть».

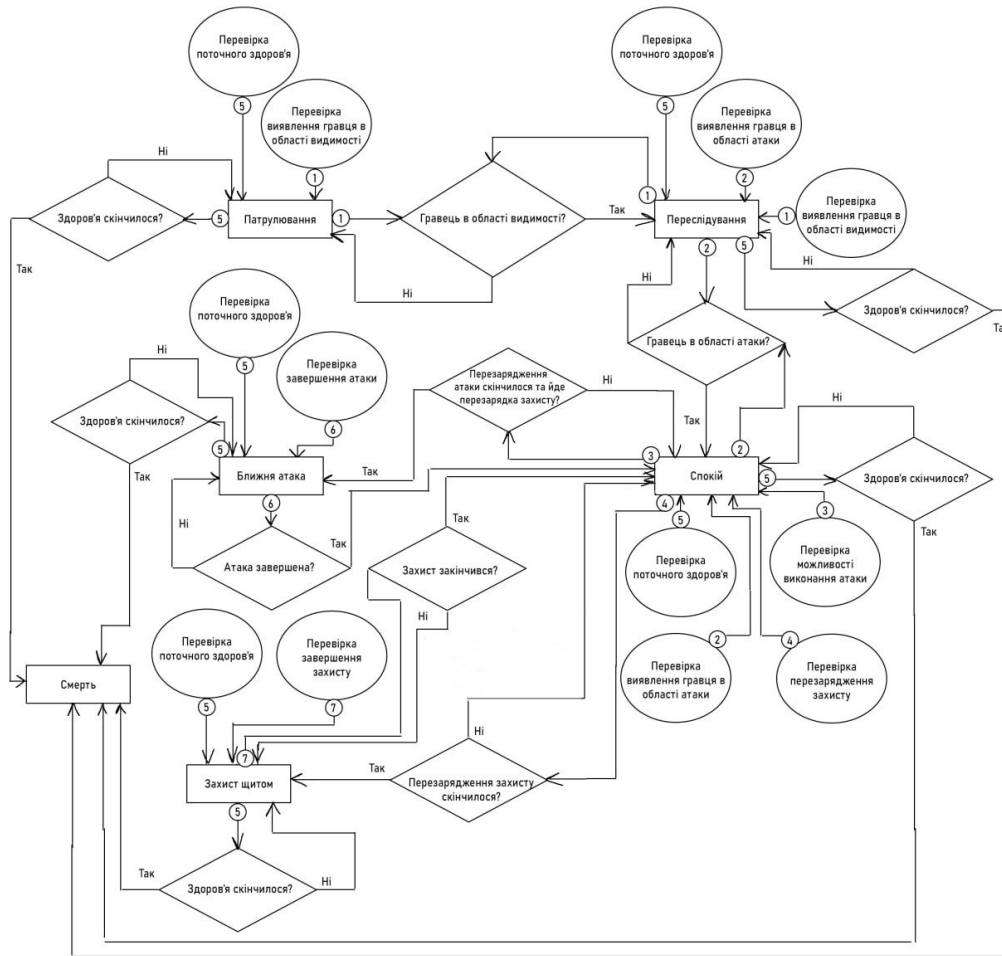


Рис. 11 – Схема скінченного автомату ворога Вампір-стражник
Fig. 11 – Schematic of the finite automaton of the enemy Vampire-guardian

Умови переходу в стан вмінь Вампіра-стражника побудовані аналогічним чином, як і в скінченному автоматі Вампіра-лучника. Особлива увага приділяється ситуаціям, коли перезарядження різних вмінь завершується одночасно. Чітко визначені умови переходу забезпечують пріоритет виконання ворожих вмінь.

Наступний розроблений скінченний автомат – скінченний автомат ворога Кажана. Даний скінченний автомат описує поведінку ворога Кажан з наступною кінцевою кількістю станів: «Очікування», «Спокій», «Переслідування», «Ближня атака» та «Смерть». Початковим станом серед них є «Очікування», а кінцевим – «Смерть».

Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка виявлення головного героя у області видимості», «Перевірка виявлення головного героя у області атаки», «Перевірка перезарядження атаки», «Перевірка завершення атаки».

Схему скінченного автомату ворога Кажана зображено на рисунку 12.

Коли головний герой потрапляє в область видимості ворога, поточний стан з початкового «Очікування» змінюється на «Спокій». Якщо головний герой в цей час залишається в області видимості і не знаходиться у області атаки, поточний стан змінюється на «Переслідування». Якщо головний герой виходить з цієї області, то стан змінюється на «Спокій» та залишається таким до тих пір, поки головний герой не потрапить у область видимості знов. Якщо поточним станом є «Переслідування» та головний герой потрапив у область атаки, то відбувається перехід у стан «Спокій». Перехід зі стану «Спокій» в стан «Ближня атака» відбувається, коли перезарядка атаки завершена і головний герой у області атаки. Назад до стану «Спокій» Кажан повертається коли стан «Ближня атака» скінчено. Якщо кількість поточного здоров'я ворога дорівнює нулю, то з будь-якого стану відбувається перехід до стану «Смерть».

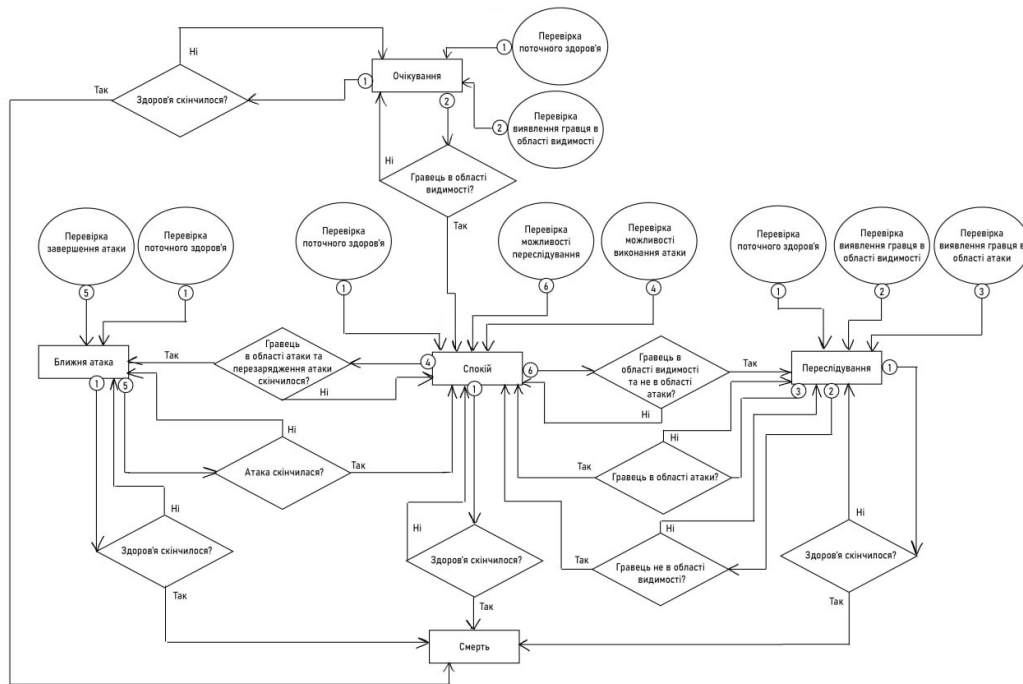


Рис. 12 – Схема скінченного автомату ворога Кажан
Fig. 12 – Scheme of the finite automaton of the enemy Bat

Відмінною особливістю даного скінченного автомата є стан «Очікування», в якому Кажан знаходиться до виявлення головного героя. У цей час ворог прихований від поглядів гравця, висячи на стелі та готуючись до нападу. Після виявлення головного героя ворог переходить у стан «Спокій» і починає активно переслідувати ціль. Проте, якщо головний герой пропадає з поля зору, Кажан вже не повертається до стану «Очікування».

Наступним розробленим скінченим автоматом є автомат ворога Палаючий дух. Схему скінченного автомату ворога Палаючий дух зображено на рисунку 13.

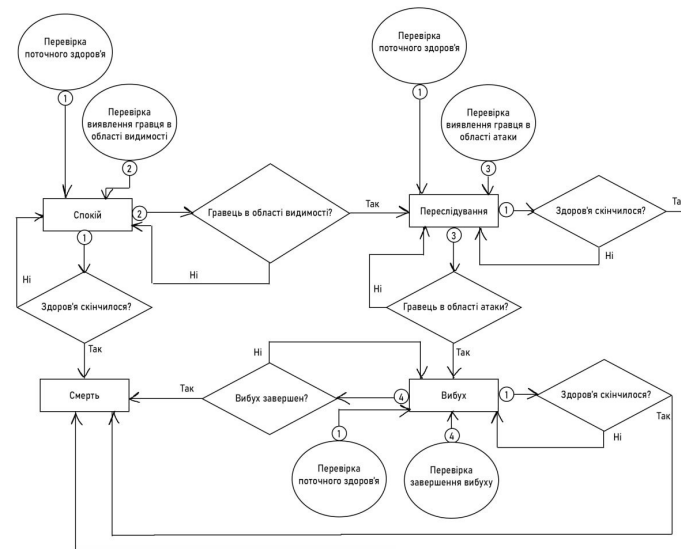


Рис. 13 – Схема скінченного автомату ворога Палаючий дух
Fig. 13 – Schematic of the finite automaton of the enemy Burning Spirit

Даний скінченний автомат описує поведінку ворога Палаючий дух з наступною кінцевою кількістю станів: «Спокій», «Переслідування», «Вибух» та «Смерть». Початковим станом є «Спокій», а «Смерть» – кінцевим станом.

Цей скінченний автомат має такі події: «Перевірка поточного здоров'я», «Перевірка виявлення головного героя в області видимості», «Перевірка виявлення головного героя», «Перевірка закінчення вибуху».

Коли головний герой потрапляє у область видимості ворога, то відбувається перехід зі стану «Спокій» в «Переслідування». Якщо головний герой потрапив у область атаки, відбувається перехід зі стану



«Переслідування» в «Вибух». Коли вибух закінчується, відбувається перехід в остаточний стан «Смерть». Якщо кількість поточного здоров'я дорівнює нулю, відбувається перехід з будь-якого стану у стан «Смерть».

Палаючий дух виділяється своєю унікальною поведінкою в бою. Після виявлення головного героя він неухильно переслідує його, прагнучи фінального удару – вибуху. Після вибуху ворог не повертається до попередніх станів і просто зникає. Це створює особливу динаміку бою, де гравцеві необхідно швидко реагувати та атакувати ворога до того, як той завдасть смертельного удару, оскільки уникнути його переслідування неможливо.

Загальні висновки

Розробка та підтримка логіки поведінки віртуальних персонажів у комп'ютерних іграх відзначається суттєвою складністю. Розробники стикаються з необхідністю створення численних умов та ланцюжків логіки, що регулюють поведінку персонажів. Це призводить до ускладнення коду та збільшення ймовірності виникнення помилок. Для моделювання логіки поведінки віртуальних персонажів доцільно скористатися апаратом, методами та підходами скінченних автоматів. В якості ігрового жанру для моделювання логіки поведінки віртуальних персонажів обрано досить поширений та популярний жанр «2D-Platformer».

В результаті аналізу предметної області розглянуто походження жанру «2D-Platformer» та визначено його основні характеристики та механіки, що впливають безпосередньо на розробку матриць взаємодії та скінченних автоматів.

Розглянуто різні моделі, які доцільно використовувати при проектуванні ігор загалом та моделюванні логіки поведінки віртуальних персонажів зокрема. Показано, що для складних ігор при моделюванні логіки поведінки віртуальних персонажів доцільно використовувати моделі, що враховують події, стани, та переходи. Для таких моделей розглянуто наступні інструменти моделювання поведінки віртуальних персонажів: токени, матриця взаємодії токенів, скінченний автомат.

Розроблено необхідні фрагменти проектної документації, в яких описано основні компоненти та особливості ігрового проекту. Розроблено спрощену матрицю взаємодії токенів та скінченні автомати головного героя та ворогів. Скінченні автомати детально демонструють логіку поведінки віртуальних персонажів

Подальшим розвитком роботи є програмна реалізація скінченних автоматів головного героя та ворогів для розробки демонстраційної версії гри.

Список використаних джерел

- [1]. Ken Horowitz, *Beyond Donkey Kong: A History of Nintendo Arcade Games* // McFarland & Company, Inc.: North Carolina, 2020. – 264p.
- [2]. Super Meat Boy's Hidden Depth [Електронний ресурс] – Режим доступу: <https://www.ign.com/articles/2009/10/14/super-meat-boys-hidden-depth>
- [3]. 10 Types of Platforms in Platform Video Games [Електронний ресурс] – Режим доступу: <https://www.idtech.com/blog/10-types-of-platforms-in-platform-video-games>
- [4]. Unit-1 Game Core Design & Initial Design [Електронний ресурс] – Режим доступу: <https://www.kdkce.edu.in/pdf/Unit%201-NEW%20-%20Game%20Core%20Design.pdf>
- [5]. Шестопалов С. В. Конспект лекцій з освітнього компоненту «Програмне забезпечення ігрових систем» [Електронний ресурс]: для здобувачів СВО «Бакалавр» спец. 123 «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології» ОПП «Розробка ігор та інтерактивних медіа у віртуальній реальності» ден. та заоч. форм навчання / С. В. Шестопалов, І. В. Колумба; Каф. комп'ютерної інженерії. – Одеса: ОНТУ, 2024. – 121с.
- [6]. Andrew Rollings. *Game Architecture and Design* / Andrew Rollings, David Morris. – New Riders, 2003. – 960p.
- [7]. Кінцевий автомат (FSM) [Електронний ресурс] – Режим доступу: https://docs.aiogram.dev/uk-ua/latest/dispatcher/finite_state_machine/index.html

References

- [1]. Ken Horowitz, *Beyond Donkey Kong: A History of Nintendo Arcade Games* // McFarland & Company, Inc.: North Carolina, 2020. – 264p.
- [2]. Super Meat Boy's Hidden Depth [Elektronnyi resurs] – Rezhym dostupu: <https://www.ign.com/articles/2009/10/14/super-meat-boys-hidden-depth>
- [3]. 10 Types of Platforms in Platform Video Games [Elektronnyi resurs] – Rezhym dostupu: <https://www.idtech.com/blog/10-types-of-platforms-in-platform-video-games>
- [4]. Unit-1 Game Core Design & Initial Design [Elektronnyi resurs] – Rezhym dostupu: <https://www.kdkce.edu.in/pdf/Unit%201-NEW%20-%20Game%20Core%20Design.pdf>
- [5]. Shestopalov S.V. Konspekt lektsii z osvithnoho komponentu «Prohramne zabezpechennia ihrovykh system» [Elektronnyi resurs]: dlia zdobuvachiv SVO «Bakalavr» spets. 123 «Kompiuterna inzheneriia» haluzi znan 12 «Informatsiini tekhnolohii» OPP «Rozrobka ihor ta interaktyvnykh media u virtualnii realnosti» den. ta zaoch. form navchannia / S.V. Shestopalov, I.V. Kolumba; Kaf. kompiuternoii inzhenerii. – Odesa: ONTU, 2024. – 121s.
- [6]. Andrew Rollings. *Game Architecture and Design* / Andrew Rollings David Morris. – New Riders, 2003. – 960p.
- [7]. Kintsevyi avtomat (FSM) [Elektronnyi resurs] – Rezhym dostupu: https://docs.aiogram.dev/uk-ua/latest/dispatcher/finite_state_machine/index.html

Отримана в редакції 08.01.2025. Прийнята до друку 03.02.2025. Received 08 January 2025. Approved 03 February 2025. Available in Internet 28 March 2025.