

## ПІДХІД ДО РОЗРОБКИ КОМП'ЮТЕРНОЇ ГРИ НА БАЗІ КРОСПЛАТФОРМЕНОЇ МУЛЬТИМЕДІЙНОЇ БІБЛІОТЕКИ SFML ОБ'ЄКТНО-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ

Багрий Р.О., Тищенко О.О. Дідур В.О.  
(g.cardinal2009@gmail.com, tyschenko.helen@gmail.com, pravetz@ukr.net)  
Хмельницький національний університет

*Запропоновано програмну реалізацію комп'ютерної гри на базі кросплатформеної мультимедійної бібліотеки SFML об'єктно-орієнтованої архітектури, що вносить мінімальні зміни у правила класичної гри. Додатково, програмний продукт надає широкі можливості до модифікування ігрових рівнів, текстур, персонажів, локалізації та ігрового інтерфейсу, дозволяючи користувачу фактично створити власну гру на основі цієї програми.*

Гра-лабіринт являє собою жанр відеоігор, ігровий процес яких відбувається на мапі у вигляді лабіринту. Перші відеоігри-лабіринти з'явилися у 70-х роках ХХ століття і були простими за своєю концепцією і графікою, але з плином часу цей жанр розвивався. Гра Pac-Man, випущена компанією Namco у 1980 році, мала величезний вплив на жанр відеоігор-лабіринтів: вона була першою грою, де ідея лабіринту використовувалась як основа геймплею. «Pac-Man» є однією з найвпізнаваніших та найуспішніших аркадних ігор усіх часів. На основі її концепцій було створено безліч подібних ігор-лабіринтів, де гравець маневрує персонажем у складному лабіринті, збираючи різні предмети або уникаючи ворогів.

При розробці комп'ютерних ігор, ефективне управління інформацією є надзвичайно важливим аспектом [1]. Інформаційна структура системи гри визначає, як дані та функції взаємодіють між собою, забезпечуючи гнучкість, швидкодію та зрозумілість розробленої програми.

Підсистема налаштувань, яка створює (якщо файл не існує) та завантажує файл налаштувань користувача при запуску є стрижнем всієї структури, вона надає іншим підсистемам інформацію, необхідну їм для стабільного функціонування, у вигляді текстових файлів та/або зображень. Інформаційна структура системи наведена на рисунку 1 демонструє, як саме пов'язані всі підсистеми та яку інформацію вони отримують.

Створена програмна реалізація є об'єктно-орієнтованою. Клас «Game» є ядром усієї гри, тут відбувається створення/завантаження налаштувань користувача, у методі «LoadSettings», всі налаштування зберігаються у змінній GAME\_PATHS, звідки інші підсистеми отримують шляхи до текстових файлів та зображень. Методи «LoadGameText», «LoadOverlay», «LoadEntity» й «LoadLifeBar» використовуються, щоб завантажити внутрішньоігровий текст, позиції елементів інтерфейсу, дані про пакмена та привидів й текстури покажчика життів гравця відповідно. Метод «GenerateMap» приймає шлях до зображення мапи у вигляді аргументу і викликає об'єкт m класу Map, який генерує її та повертає початкові позиції всіх персонажів, які були завантажені грою. «SetEntitiesPositions» виставляє привидів та пакмена на початкові позиції, отримані в результаті генерації ігрової карти. «UpdateBatch» спричиняє оновлення об'єкта b класу Batchер, який буде описано згодом, цей метод викликається тільки один раз, при формуванні фінальної картинки гри, яку бачитиме користувач на екрані комп'ютера, якщо ж увімкнено рандомну зміну мап з кожним новим рівнем, цей метод буде викликатися після кожного пройденого рівня. Метод «Game\_KeyPressed» відповідає за зчитування введення користувача з клавіатури, а «Game\_Resized» займається масштабуванням інтерфейсу при зміні розмірів вікна гри, у цьому йому допомагає метод «SetOverlay», який виставляє позиції та кольори ігрового тексту.

Клас «Entity» є своєрідним прототипом для класів «Pacman» та «Ghost», він зберігає основну інформацію, потрібну для відмальовки персонажів: розміри, позицію у ігровому світі, текстуру за замовчуванням, напрямок руху а також має обробку зіткнень з мапою. Класи «Pacman» і «Ghost», які успадковують «Entity», зберігають і реалізують вже специфічні для себе методи та поля.

Інтерфейс «Drawable» та клас «Transformable» є базовими інструментами бібліотеки SFML для реалізації користувацьких графічних сутностей, які будуть відмальовуватись на екрані(Drawable) та змінюватимуть координати, обертання, матимуть систему зіткнень між собою та ігровим світом(Transformable).

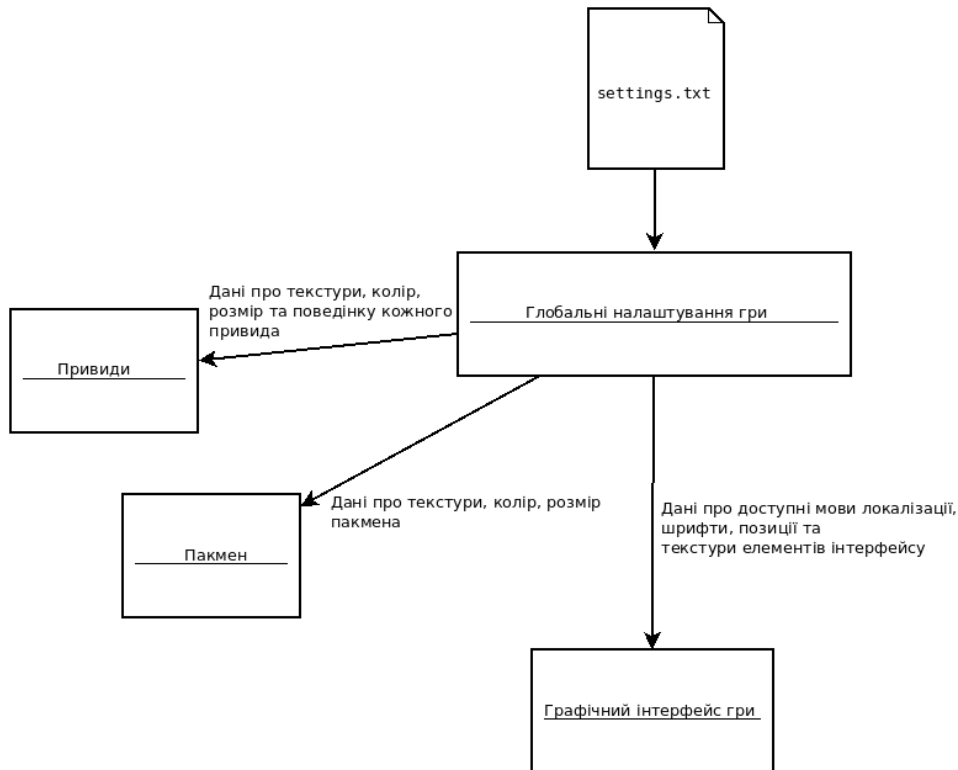


Рисунок 1 – Інформаційна структура програми

Клас «Pacman» містить специфічні для персонажа на ім'я Пакмен поля та властивості: рахунок(поле score, властивість-getter Score), кількість життів (поле lifes, властивість-getter Lives), таймер підсилення гравця(поле energizer\_timer, властивість-getter EnergizerTimer) та інші. Метод «Move» відповідає за безпосередньо рух гравця по рівню, у ньому ж відбуваються перевірки на зіткнення(Викликаючи успадкований від «Entity» метод «Collides») і з'їдання пакменом їжі(Викликаючи метод «Eats»). «EnergizerClock» рахує час до закінчення підсилювачів, які дозволяють головному герою з'їдати привидів. Метод «EatsGhost» викликається, коли гравець зіткнувся з наляканим привидом, цей метод збільшує кількість привидів, яку з'їв Пакмен, нараховує гравцю додаткові очки та звертається до програвача звуків soundPlayer, який відтворює відповідний для цієї дії звук. Метод «Die» запускає анімацію смерті та відповідний звук, коли з Пакменом зіткнувся не наляканий привид. «BonusLife» додає головному герою 1 життя при досягненні ним рахунку в 10 тисяч очок, або кратного йому значення. «Restart» скидає всі досягнення гравця й викликається тільки коли гра починається заново. «ResetEnergizer» скидає таймер підсилення гравця.

Клас «Map» опікується генерацією ігрової карти, респавом їжі та фруктів при зміні рівня. Метод «LoadCfg» дозволяє завантажити інформацію про текстури стін та їжі. Метод «Generate» генерує мапу, яку побачить користувач. Для генерації використовується зображення, де кожен піксель представляє тайл 16x16, колір пікселя визначає тип тайла: стіна, їжа чи просто порожній тайл.

Клас «AnimationPlayer» відповідає за відтворення анімацій персонажів його можна прив'язати до будь-якого класу, який успадковує «Entity». Метод «LoadAnimation» завантажує текстовий файл з анімаціями та створює на основі інформації з нього кадри. «PlayAnimation» відтворює анімацію, знаходячи кадри, які закріплені за її назвою та поточним напрямком руху. «ResetFrames» скидає поточний кадр анімації та час, який був витрачений на відтворення.

За спроектованими складовими було розроблено програмну реалізацію, інтерфейс якої наведено на рисунку 2.



Рисунок 2 – Вигляд роботи програми

Отже, створено створено комп'ютерну гру «Pac-Man» на базі кросплатформеної мультимедійної бібліотеки SFML. Процес розробки гри включав в себе аналіз вимог, проектування архітектури, реалізацію функціональності та тестування. Використовувана бібліотека SFML надала зручний інтерфейс для роботи з графікою, звуком та подіями, що дозволило ефективно реалізувати функціональність гри.

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Божик І.С., Мазурець О.В., Багрій Р.О., Кліменко В.І., Тищенко О.О. Метод організації самонавчальної комп'ютерної гри алгоритмом еволюційного навчання штучних нейронних мереж. Матеріали III Всеукраїнської науково-технічної конференції «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2023». Одеса. 2023. С. 165-167.

УДК 004.42+004.8

### РОЗРОБКА ІГОР В ТЕНДЕНЦІЯХ РОЗВИТКУ ШТУЧНОГО ІНТЕЛЕКТУ: ЗАНЕПОКОЄННЯ ТА РЕАЛІЇ

Гаранін О. М. (olejenuc@gmail.com)

Криворізький державний педагогічний університет (Україна)

В умовах стрімкого розвитку штучного інтелекту багато сфер, включаючи розробку ігор зазнали значних змін в процесах, адже ті чи ті прості задачі тепер можна виконувати за допомогою штучного інтелекту, що із самого початку почало викликати ряд занепокоєнь щодо того, чи не стане це шкідливим для трудового ринку і чи не призведе в наслідку до значного скорочення робочих місць. Використання штучного інтелекту у виробничих процесах викликає дискусії стосовно потенційних соціальних наслідків, що спричинило широке обговорення цього питання серед науковців і професіоналів галузі. Важливим прикладом є занепокоєння щодо використання ШІ в процесах розробки відеоігор [1].

Кількість генеративного контенту, який створюється за допомогою ШІ дійсно великий, і є навіть проекти, які було повністю створено за допомогою даних інструментів. Так, дійсно, вимоги до кандидатів стають все більш широкими і потребують володіння все більшою кількістю інструментів для успішної роботи навіть початку роботи на початкових рівнях кар'єри (trainee-junior).

Але чи дійсно дана проблема викликана саме наявністю технологій ШІ? Насправді, ринок постійно розвивається і збільшення вимог щодо тієї чи тієї посади зумовлена наявністю конкуренції на ринку праці та доступністю технологій. Процес залучення молоді до сучасних технологій зараз відбувається дуже стрімко за рахунок високої доступності комп'ютерної техніки та гаджетів. Отже, в зрізі часу можна побачити основну проблему, яка насправді може впливати на