УДК 004.584
DOI https://doi.org/10.32782/2663-5941/2023.3.1/25

Кравченко С.М.
Державний університет «Житомирська політехніка»

Сугоняк І.І.
Державний університет «Житомирська політехніка»

Марчук Г.В.
Державний університет «Житомирська політехніка»

Гришкун Є.О.
Державний університет «Житомирська політехніка»

Венгловська Ю.М.
Державний університет «Житомирська політехніка»

UML-МОДЕЛЮВАННЯ ПРОЦЕСУ ПРОЕКТУВАННЯ ГРИ В ЖАНРІ ГОЛОВОЛОМКИ

Об'єктом дослідження є принцип застосування UML-моделювання моделі для розробки комп'ютерних ігор. В даній статті представлено UML-моделювання побудови моделі логічної гри в жанрі головоломки з використанням 2D графіки.

Проведено аналіз вивчення даної області подібних процесів і проведення власних досліджень, які в свою чергу потребують більш поглиблено занурюватись у саму сутність наукової області та розширювати горизонти для подальших досліджень.

Визначено алгоритм роботи та стани програмної системи, фізична модель. Продемонстровано застосування об'єктно-орієнтованої моделі системи, здійснено проектування ігрового процесу і здійснення етапів розробки комп'ютерної гри. Використано UML-модель для візуалізації та оптимізації представлення гри.

У роботі визначено аналіз вимог, концептуальне моделювання побудові моделі розробки логічної гри., Проаналізовано основні етапи створення комп'ютерної гри. Охарактеризовано функціональні можливості використання UML-моделювання при розробці гри. UML-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

UML-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

Ключові слова: *моделювання програмного забезпечення, логічна гра, фізична модель, UNITY , UML, графіка, середовище розробки.*

Постановка проблеми. Для того, щоб розробити складний проект з великою кількістю механік та можливостей, обов'язково треба спочатку його спроектувати та чітко визначити, що саме зможе робити користувач.

Існують високо рівневі вимоги, якими повинна володіти гра, щоб мати можливість добре виконувати покладені на неї функції, тим самим відповідно задовольнивши всі зазначені стандарти, специфікації та інші формальні документи. Тому було використано UML-моделювання.

Актуальність теми дослідження. Популярність комп'ютерних ігор зростає з кожним днем, діти та дорослі проводять за цим заняттям все більше часу. Люди звертаються до ігор, щоб впоратися з невпевненістю, урізноманітнити рутину та поспілкуватися з друзями. Не всі вони грали раніше, багато хто робить це вперше. Проте всіх їх об'єднує спільний інтерес – бажання випробувати у віртуальних світах щось нове, незвідане, спробувати удачу і отримати насолоду як від ігрового процесу, так і від досягнутих у грі результатів.

У зв'язку зі зростаючою комп'ютеризацією, розвиток морфології дизайн-об'єктів виходить з наступності еволюційного досвіду в межах

певної художньо-естетичної традиції та базується на інноваційних наукових досягненнях [8].

Логічне мислення – це безцінний і, мабуть, найнеобхідніший навичка для програміста: і досвідченого, і початківця. ІТ-фахівцеві вдається виявляти закономірності, відстежувати їх і самостійно задавати. Логічне мислення також допомагає програмісту створювати нестандартні рішення поставлених завдань, справлятися зі складними завданнями та сприяє підтримці вогню творчості. Для розвитку логіки відмінно підходять комп'ютерні ігри – це захоплюючі міні-тренування для мозку, які дозволяють відволіктися від поточних завдань і допомагають добре перезавантажитися [9].

Будь-яка гра – це не просто гарна графіка і завдання, яке потрібно виконати, це і сюжетна лінія, і різні рішення і дії, які призведуть до різних результатів. Тому, для легшого орієнтування, створюються блок-схеми з усіма згаданими можливостями. Особливо це корисно у великих проектах, над якими працює велика кількість людей.

Аналіз останніх досліджень і публікацій.

Дослідженням щодо використання UML-моделювання займались такі зарубіжні та вітчизняні вчені як У. Боггс, М. Боггс, Р. Буч, Дж. Рамбо, А. Джекобсон, А.М. Вендров, Т. Кватрані, А.Г. Українець, М.Ф. Бондаренко. Базова система позначень UML популярно і доступно викладена в книзі Мартіна Фовлера [2; 5].

Вчені пропонують аналіз змодельованих механік, який полягає в перевірці на досяжність і несуперечливість сценаріїв ігрових механік [6], розробляють методологію оцінки для перевірки продуктивності моделі UML, що представляє архітектуру програмного забезпечення [6].

Запропоновано розглядати процес створення ігрового сценарію як, насамперед, набір вимог, послідовність відтворення сценарію [8]. Для цих систем моделювання і аналіз поведінки є найбільш важливою і проблемою [4].

Метою статті є визначення доцільності використання UML-моделювання при стадії проектування комп'ютерних ігр.

Виклад основного матеріалу. UML (Unified Modeling Language) представляє собою стандартизовану мову моделювання, що складається з інтегрованого набору діаграм, яка розроблена, щоб допомогти розробникам систем та програмного забезпечення визначати, візуалізувати, створювати і документувати артефакти програмних систем, а також для бізнес-процесів моделювання. UML – це набір найкращих інженерних практик, які дуже ефективні при моделюванні великих та складних систем.

В основу UML-моделей покладений принцип спрощення моделі об'єктів через їхню класифікацію, тобто поділ на різновиди відповідно до важливих ознак (структури або атрибутики, поведінки або зв'язку з іншими класами, спадкоємності ознак) та поєднання у класи.

UML є дуже важливою частиною процесу розробки об'єктно-орієнтованого програмного забезпечення. Цей вид моделювання забезпечує в основному графічні позначення для опису дизайну програмних проектів. Використання UML допомагає проектним групам спілкуватись, вивчати потенційні проекти та перевіряти архітектурний дизайн програмного забезпечення.

Так як UML-моделювання використовується загалом у програмуванні різних систем, визначається доцільність використання і для відображенні об'єктів гри через застосування графічних та структурованих описів.

Представлено різні етапи проектування сценарію гри у вигляді діаграм UML.

Так як проект доволі великий, і розробляється у команді з трьох учасників, надалі будуть зображуватися лише діаграми, що стосуються частини проекту, пов'язану з головоломками.

Діаграма прецедентів

Діаграма варіантів використання описує систему на концептуальному рівні та показує відносини між акторами (персонажами, які беруть участь у ситуації або грі) та прецедентами (діями гравця, що призводять до певного відчутного результату). Така діаграма дозволяє побачити повторюваність дій, зайві рухи, що дає підстави для оптимізації бо свідомого збагачення дійової системи-об'єкта.

На рисунку 1 представлено діаграму прецедентів гравця.

Алгоритм роботи та стани програмної системи

Діаграми діяльності описують, як діяльність координується для надання послуги, яка може бути на різних рівнях абстракції. Як правило, подія має бути досягнута деякими операціями, особливо якщо операція призначена для досягнення кількох різних речей, які вимагають координації, або як події в одному варіанті використання співвідносяться одна з одною, зокрема випадки використання, де дії можуть збігатися і вимагати узгодження.

Створення діаграми діяльності представлена на рис. 2–4.

Рис. 1. Діаграма варіантів використання гри

Рис. 2. Діаграма активності для перетягування предмета

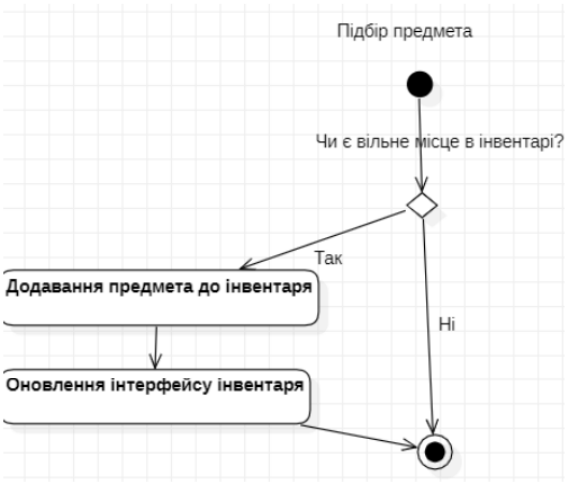

Рис. 3. Діаграма активності для підбору предмета

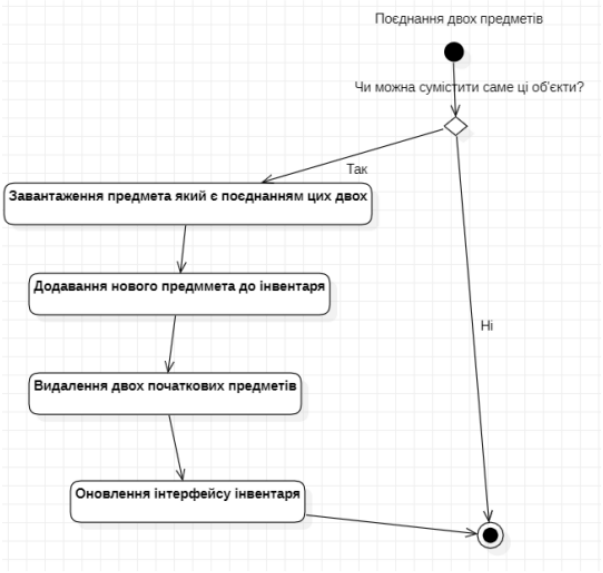

Рис. 4. Діаграма активності для використання предмета з інвентаря

Об’єктно-орієнтована модель системи

Діаграма класів являє собою набір статичних, декларативних елементів моделі. Вона дає найбільш повне і розгорнуте уявлення про зв'язки в програмному коді, функціональність та інформацію про окремі класи. Додатки генеруються часто

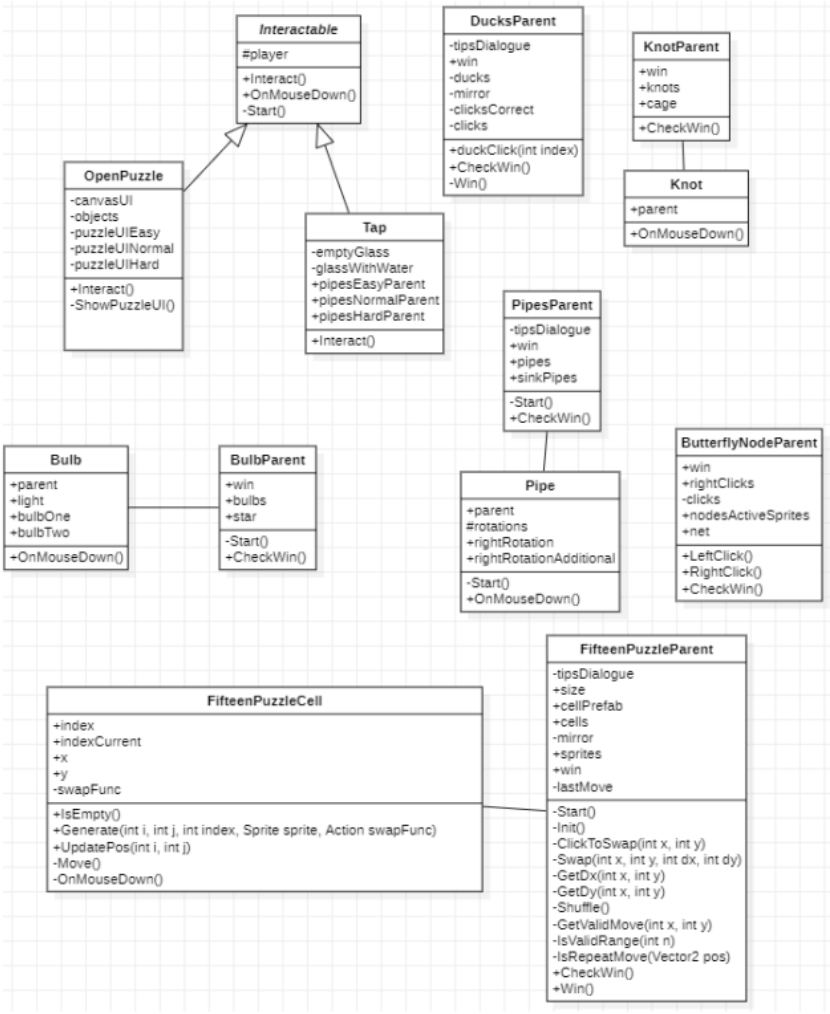

Рис. 5. Діаграма класів логічної гри

саме з діаграми класів. При моделюванні об'єктно-орієнтованих систем цей тип діаграм використовують найчастіше. Виходячи з даних відомостей, було побудовано діаграму класів, зображену на рис. 5.

Взаємодія об'єктів системи

Діаграми послідовності UML – це діаграми взаємодії, які детально описують, як виконуються операції. Вони фіксують взаємодію між об'єктами в контексті співпраці. Діаграми послідовності зосереджені на часі, і вони візуально показують порядок взаємодії за допомогою вертикальної осі діаграми, щоб відобразити час, які повідомлення надсилаються та коли. Для спроектованої гри було реалізовано діаграму послідовності – рис. 6. Так як в цій частині проектування мова йде лише про окремі головоломки, то взаємодії відбуваються лише в межах кожної задачі.

Як видно з діаграми, зазвичай один клас звертається до іншого на початку роботи (метод Start()), при перевірці на виграш (метод CheckWin()), при кліку мишею та деякі інші. В цих методах найчастіше клас, що відповідає за міні-гру в цілому, викликає об'єкти пов'язаного класу.

Фізична модель та прототип програмного комплексу

Взаємодія компонентів системи

Для демонстрації взаємодії компонентів системи використовуються діаграми реалізації, які включають дві UML діаграми: діаграму компонентів (component diagram) і діаграму розгортання (deployment diagram).

Діаграма компонентів, також відома як діаграма компонентів UML, описує організацію та підключення фізичних компонентів у системі. Діаграми компонентів часто малюються, щоб допомогти змоделювати деталі впровадження та ще раз перевірити, чи всі аспекти необхідних функцій системи охоплені плановою розробкою. Діаграма основних компонентів проектованої системи зображена на рис. 7.

Архітектура програмного комплексу та його розгортання

Діаграма розгортання – це тип діаграми UML, який показує архітектуру виконання системи,

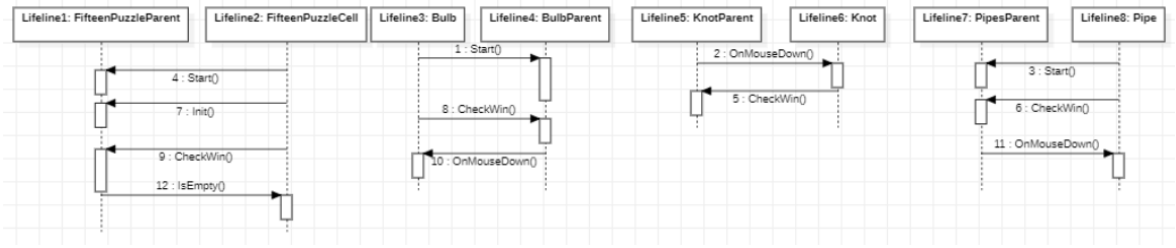

Рис. 6. Діаграма послідовності

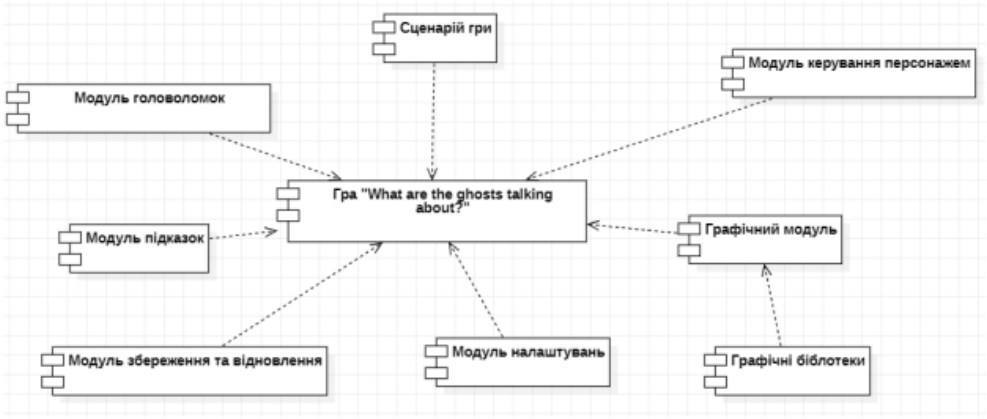

Рис. 7. Діаграма компонентів

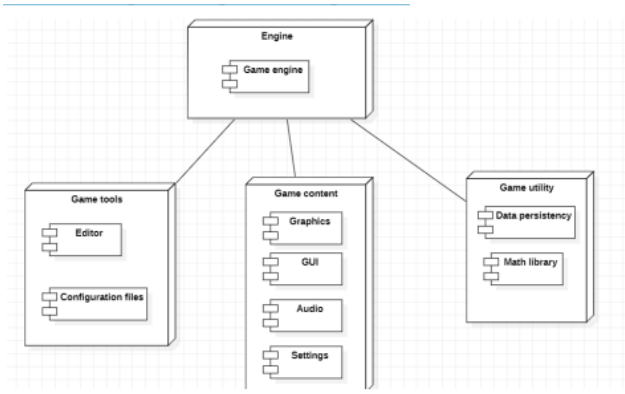

Рис. 8. Діаграма розгортання

включаючи такі вузли, як середовища виконання апаратного чи програмного забезпечення, і проміжне програмне забезпечення, що їх з'єднує. Діаграми розгортання зазвичай використовуються для візуалізації фізичного обладнання та програмного забезпечення системи. Використовуючи його, ви можете зрозуміти, як система буде фізично розгорнута на апаратному забезпеченні. Діаграми розгортання допомагають змоделювати апаратну топологію системи порівняно з іншими типами діаграм UML, які здебільшого описують логічні компоненти системи.

Виконана діаграма зображена на рис. 8.

Висновки. У роботі визначений підхід до застосування UML-моделювання при проектуванні сценарію комп'ютерної гри. Представлено, що UML-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

Визначено, що запропонований підхід може застосовуватись в даній сфері розробки, а саме дає широкі можливості застосування UML-моделей для відображення подій (концептуальне моделювання, логічний та фізичний рівень) при розробці гри.

Список літератури:

1. Хокінг Д. М. Unity в дії. Мультиплатформенна розробка на практиці. 2-е міжн. видання. 2016. 336 с.

2. G. Booch, J.Rumbaugh., I.Jacobson. The Unified Modeling Language User Guide . MA.: Addison-Wesley Publishing Co. 1999. 512 p.

3. Blazhko, O., & Luhova, T. Features of using the canvas-oriented approach to game design. Applied Aspects of Information Technology. 2018. №1(01). 66-77p.

4. Лугова, Т.А., Лись Д.А. UML-моделі як основа проектування та балансування сценаріїв. ΛΌGOΣ мистецтво наукової думки: Технічні науки. № 7. С. 33-37.

5. S. Distefano, M. Scarpa, A.Puliato. From UML to Petri nets: The PCM-based methodology. IEEE Transactions on Software Engineering. 2010. № 37 (1). 65-79p.

6. Harel, D. From play-in scenarios to code: An achievable dream. Computer. 2000. № 34(1). 56-78p.

7. Antoshchuk, S., Arsirii, O., Blazhko, O., Troianovska, Y., & Luhova T. Method for Detecting Errors inthe Design of Virtual Environments. The 10th IEEE International Conference on Intelligent Data Acquisitionand Advanced Computing Systems: Technology and Applications (IDAACS), (Metz, France, 18-21 September 2019). France, 2019. 20-24p.

8. Сухорукова Л. А. Види та класифікація технологічних засобів створення мультимедійного продукту [Електронний ресурс]. Традиції та новації у вищій архітектурно-художній освіті: Харків. Держ. акад. дизайну і мистецтв. – Харків, 2012. URL: https://ksada.org/articles/suhorukova-article-07.pdf. (дата звернення: 21.11.22).

9. Найцікавіші логічні ігри для програмістів. [Електронний ресурс]. URL: https://itfuture.online/uk/najczikavishi-logichni-igry-dlya-majbutnih-programistiv/(дата звернення: 11.10.22).

Kravcenko S.M., Suhoniak I.I., Marchuk H.V., Gryshkun Ye.O., Venhlovska Yu.M.

UML MODELING OF THE DESIGN PROCESS OF A PUZZLE GAME

The object of the research is the principle of applying the UML modeling model for the development of computer games. This article presents UML modeling of the construction of a logic game model in the puzzle genre using 2D graphics.

An analysis of the study of this area of similar processes and the conduct of own research, which in turn require a more in-depth immersion in the very essence of the scientific area and expanding the horizons for further research, has been carried out.

The work algorithm and states of the software system, physical model are defined. The application of the object-oriented model of the system was demonstrated, the game process was designed and the stages of computer game development were carried out. The UML model was used to visualize and optimize the presentation of the game.

The work defines the analysis of requirements, conceptual modeling of the construction of a logical game development model. The main stages of creating a computer game are analyzed. The functional possibilities of using UML modeling in game development are characterized.

Key words: *software modeling, logical game, physical model, UNITY, UML, graphics, development environment.*