

UML-МОДЕЛІ ЯК ОСНОВА ПРОЕКТУВАННЯ ТА БАЛАНСУВАННЯ СЦЕНАРІЇВ КОМП'ЮТЕРНИХ ІГОР

**Лугова
Тетяна Анатоліївна**

ORCID ID: 0000-0002-3573-9978

доктор філософії, кандидат мистецтвознавства, доцент, доцент кафедри інформаційної діяльності та медіа-комунікацій

Одеський національний політехнічний університет

**Лись
Дар'я Анатоліївна**

здобувач вищої освіти інституту комп'ютерних систем

Одеський національний політехнічний університет

УКРАЇНА

UML MODELS AS THE BASIS OF DESIGNING AND BALANCING COMPUTER GAMES SCENARIOS

АНОТАЦІЯ | ABSTRACT :

Охарактеризовано функціональні можливості використання UML-моделювання при проектуванні та балансуванні комп'ютерних ігор. Зазначено, що UML-моделювання дає можливість не лише візуалізувати та оптимізувати сценарії гри, а й балансувати їх. Співвіднесено види балансів гри з діаграмами UML, наведено приклади. З'ясовано, що на відміну від класифікування, кінцевим продуктом якого є класифікаційна таблиця; та інфографіки, яка загалом представляє логіко-семантичні зв'язки між об'єктами, UML-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

Features of the use of UML modeling in the design and balancing of computer games. It is stated that UML modeling allows not only to visualize and optimize game scenarios, but also to balance them. The types of game balances are correlated with UML diagrams, examples are given. It is found that, unlike the classification, the end product of which is a rating table; and infographics, which are generally semantic links between objects, UML models have more functionality to display real and virtual reality objects through the use of graphical and structured verbal descriptions.

КЛЮЧОВІ СЛОВА | KEYWORDS :

UML modeling; video game balancing techniques, computer game scenarios; gameplay; game mechanics.

UML-моделі; методи балансування відеогри, сценарії комп'ютерних ігор; гейміфікація; механіка ігор.

ПОСТАНОВКА ПРОБЛЕМИ.

Комп'ютерні ігри - це складні креативні системи, які засновані на правилах. Сьогодні комп'ютерні ігри широко застосовуються не лише як засіб розваги, а й як навчальні середовища. З огляду на стрімке зростання популярності відеоігор, обґрунтування їх позитивних впливів у процесі навчання [1], констатується необхідність розробки віртуально-ігрових середовищ необхідного рівня якості. Існують різні інструменти і методи проектування, які дозволяють швидко розробляти якісні комп'ютерні ігри. Серед них затребуваним вважається використання UML-діаграм, які не потребують високого порогу знань для їх використання, а також здатних налагодити порозуміння між геймдизайнером і замовником. Одним з етапів проектування є проектування та балансування сценаріїв комп'ютерних ігор. UML-моделювання використовується загалом у програмуванні різних систем [2], але наразі UML-моделювання показало великі можливості для застосування у таких гуманітарних галузях, як психологія (одна з проєктивних методик), консалтинг (бізнес-моделювання), проєктний менеджмент, геймдизайн (сце-

нографія та механіка гри). Все це зумовлює гостру актуальність вивчення шляхів реалізації UML-моделювання для створення комп'ютерних ігор, які, у першу чергу, орієнтовані на вирішення професійних та серйозних завдань.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ.

Слід зазначити, що аспекти автоматизації процесу геймдизайну активно розробляються в різних галузях науки: педагогіки та педагогічної робототехніці [3], теоретичних [4] і прикладних [5], [6], [7] основ геймдизайну, програмування віртуальних середовищ [8], [9]. Вчені попонують аналіз змодельованих механік, який полягає в перевірці на досяжність і несуперечливість сценаріїв ігрових механік [10], розробляють методологію оцінки для перевірки продуктивності моделі UML, що представляє архітектуру програмного забезпечення [11],

Особливості програмування ігрового сценарію розглядаються [12] в ракурсі розробки складних реактивних систем. При цьому вчені пропонують розуміти ігровий сценарій як набір вимог, послідовність повідомлень, сценарій відтворення. Програмування ігрового сценарію розглядається ними як циклічний процес перевірки

ки системи на відповідність вимогам і синтезу частин системи з вимогами. Для цих систем моделювання і аналіз поведінки є найбільш важливою і проблемою.

МЕТА СТАТТІ - охарактеризувати функціональні можливості використання UML-моделювання на етапі проектуванні та балансуванні сценаріїв комп'ютерних ігор.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ.

Unified Modeling Language (UML) – уніфікована універсальна мова моделювання, яка дозволяє моделювати будь-яку предметну галузь, що є важливою умовою ефективного геймдизайну та успішну реалізацію проекту зі створення комп'ютерної гри. Ця інфографічна методика дозволяє формалізувати задум геймдизайнера та правильно довести основні ідеї до розуміння програміста. 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 [2].

UML-специфікація вимог включає діаграми варіантів використання (*Use Case*) гри (графічний опис); сценарій використан-

ня гри (словесний опис); діаграма концептуальні класів як статика гри (графічний опис); діаграма станів / переходів як динаміка гри (графічний опис).

Діаграма прецедентів описує систему на концептуальному рівні та показує відносини між акторами (персонажами, які беруть участь у ситуації або гри) та прецентами (діями гравця, що призводять до певного відчутного результату). Така діаграма дозволяє побачити повторюваність дій, зайві рухи, що дає підстави для оптимізації або свідомого збагачення дійової системи об'єкта. Така мова опису об'єкта найбільш ефективно реалізується при створенні механіки гри – основних ігрових ходів (дій) гравця.

Так, гра «Змійка», яка може бути описана такими правилами [13]:

- 1) змійка рухається по землі, обмеженою по периметру стіною, в пошуках яблука;
- 2) змійка може рухатися вперед, повертати наліво або направо;
- 3) якщо змійка досягає яблука своєю головою, вона його з'їдає;
- 4) якщо змійка не встигає повернути, вона ударяється об стіну.

Вказані правила можна представити у вигляді діаграми прецедентів (рис. 1).

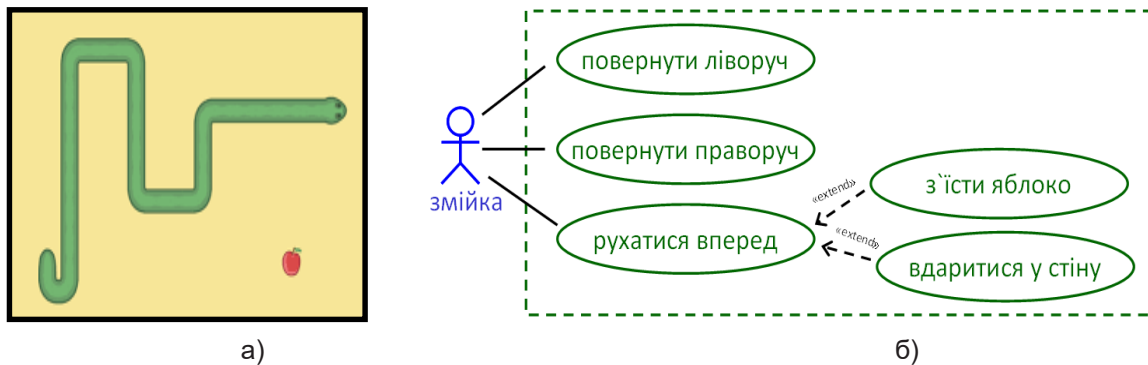


Рис. 1. Гра «змійка». Приклади екранної форми (а) та діаграми прецедентів (б)

Діаграма концептуальних класів (Concept-class) моделює статичку системи з урахуванням деталей структур класів і взаємин між класами. Use Case-діаграма не враховує деталі структур та динаміку. Така мова моделювання найбільш придатна для створення ігрового світу: ієрархії сил та героїв, взаємовідносини між елементами гри (транзитивність та нетранзитивність гри), створення ігрових кланів (родин, груп, ігрових ролей), тощо. Такі діаграми дають можливість скоротити час на пошук подібних та споріднених об'єктів або об'єктів за певними ознаками (атрибутами). На рисунку 2 наведено приклад діаграми концептуальних класів гри «Змійка».

Тож діаграма концептуальних класів

дає відповідь на питання самовизначення ігрових персонажів, самих гравців та окремо кожного ігрового об'єкта в ігровому світі, в той час як діаграма варіантів використання (*Use Case*) моделює динаміку системи на глобальному рівні її взаємин з користувачем. На відміну від класифікування, кінцевим продуктом якої є класифікаційна таблиця, та інфографіка, яка загалом представляє логіко-семантичні зв'язки між об'єктами (діаграми Ейлера та Венна, графіки Ганта тощо), *UML*-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

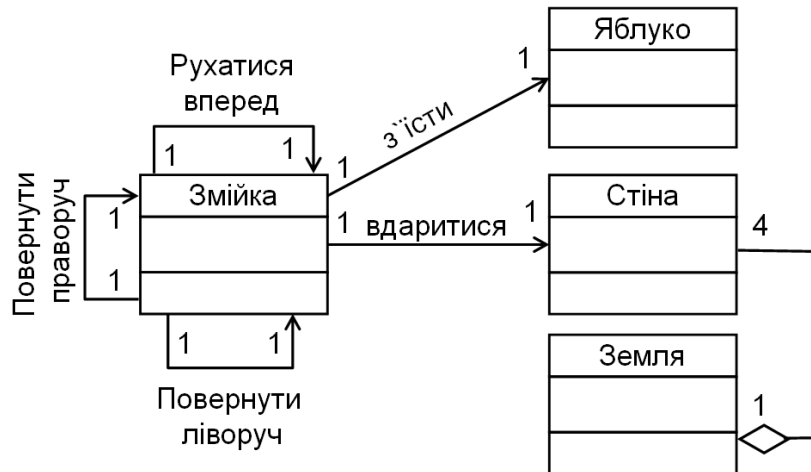


Рис. 2. Приклад діаграми концептуальних класів

Діаграма станів відображає скінчений автомат у вигляді графу, вершинами якого є стани об'єкта, поведінка якого моделюється, а переходами – події, які переводять об'єкт, який розглядається, з одного стану в інший. При цьому вважається, що час перебування об'єкта в певному стані набагато більший за час, необхідний для переходу з одного стану в інший, тобто переходи між станами здійснюються миттєво.

Діаграма станів призначена для опису станів об'єкта й умов переходу між ними. Опис станів дозволяє точно описати модель поведінки об'єкта при одержанні різних повідомлень і взаємодії з іншими об'єктами. Автомат (State machine) – це опис послідовності станів, через які проходить об'єкт протягом свого життєвого циклу, реагуючи на подію, – в тому числі опис реакції на цю подію. Стан (State) – це ситуація в житті об'єкту, протягом якої він

задовольняє певній умові, здійснює певну діяльність або очікує якусь подію. Кожен стан має ім'я та список внутрішніх дій.

UML-моделювання дає не лише візуалізувати та оптимізувати сценарії гри, а й балансувати їх. Термін «баланс» використовується для опису стану систем ігри як «збалансовані» або «незбалансовані». Ігровий баланс – рівновага між ігровими об'єктами: персонажами, командами, тактиками гри тощо. Баланс визначає складність, інтерес і плавність ігрового процесу. Ігровий баланс розуміється як з'ясування чисел, які слід використовувати у грі: на яку відстань і з якою швидкістю може бігти кожен з гравців, як далеко один від одного знаходяться гравці, розміри ігрового майданчика, як довго хто-небудь з учасників водить. Метою цих чисел є опис стану гри [14]. Тож баланс як мова опису числових залежностей гри може бути представлений засобами UML (таблиця 1).

Таблиця 1

Співвіднесення видів балансів з діаграмами UML

Види балансу за С. Гімелрейхом [15]:	Діаграми UML
1. <i>Сила</i> - це відносини об'єктів, які взаємодіють з позиції сили один з одним в грі: швидкість, захист, ударні точки (<i>hitpoints</i>), рівень вмінь (<i>skill rate</i>), атака.	Діаграма прецедентів
2. <i>Час</i> : <ul style="list-style-type: none"> - виникнення ігрових подій (квести, адвенчури, хоррор) - в який час гравцеві подати подію, щоб йому не було нудно або навпаки, як зробити так, щоб гравець не перенаситився тим, що йому подають; - час розвитку ігрових істот (ферми, білдери, ігри, в яких є прокачування персонажів) – визначення тривалості ігрової сесії, частоти виникнення відкриттів для гравця в межах кількості контенту; - тривалість гри та ігрових сесій. Таймер розвитку ігрових істот: короткий (за сесію), середній (коли запускається на цій сесії, а відбувається в наступній), і довгостроковий (щоб повернути гравця, припустимо, через 3 сесії). Не рекомендовано робити занадто довгі таймери, що фруструють гравця, тому що він не бачить свого внеску у процес гри, то мають бути проміжні нагороди. 	Діаграма станів
- <i>Простір</i> : віддзеркалення (симетрія), мапи маршрутів, навігаційні схеми, конфігурація і площа простору взаємодії.	Діаграма станів

Продовження табл. 1

- <i>Економіка</i> : визначити цінність всіх об'єктів, які є в грі (зброї, машини, мапи тощо); побудова діаграми відносин ресурсів, ресурсний цикл; поділ ресурсних циклів з ігрових активностей; побудова шкали асортименту для різних груп користувачів; прив'язка цін до номіналів банкнот і типовим витратам; оцінка попиту на товар, еластичність попиту.	Діаграма концептуальних класів
--	--------------------------------

Для досягнення балансу використовується багато методів: транзитивний, нетранзитивний, некомпаративний, пристосованості до різних умов, різниці в ризику та нагороді, автоматичного балансування.

В межах цієї статті ми розглянемо перших два. Транзитивний «А<Б<В» полягає у зіставленні характеристик об'єктів. Геймдизайнери рекомендують складати таблицю об'єктів з єдиними характеристиками для всіх.

Таблиця 2

	Персонаж 1	Персонаж 2	Персонаж 3
Швидкий	1	0	1
Спритний	1	0	0
Сильний	0	1	1
Слабкий	1	0	0
Повільний	0	1	0
Незграбний	0	1	1

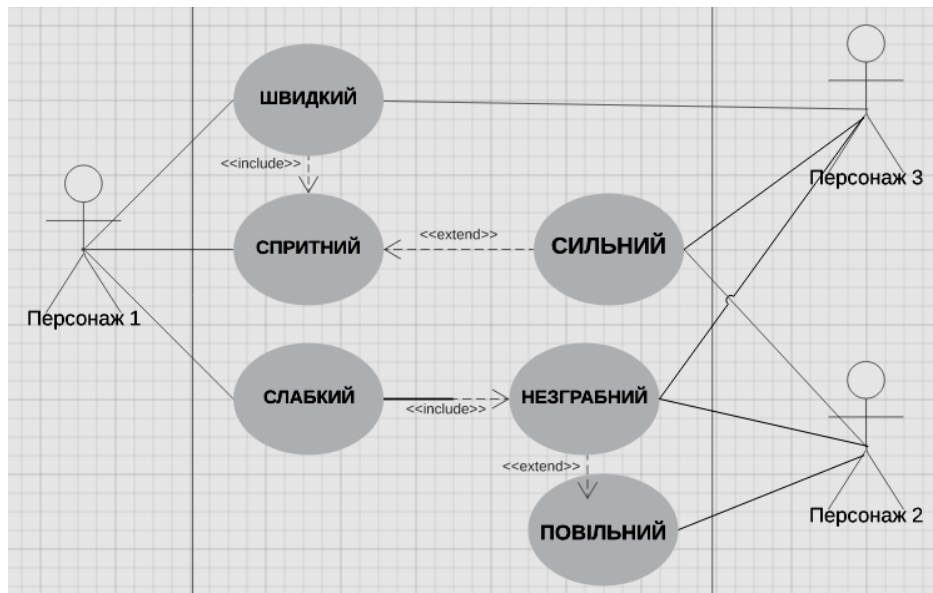


Рис. 3. Балансування сил персонажів у діаграмі прецедентів

Нетранзитивний / інтразитивний метод балансування гри застосовується у разі відсутності непереможних персонажів. Геймдизайнери рекомендують будувати матрицю з характеристиками та очками, коли об'єкти стикаються один з одним. Якщо об'єктів з різними характеристиками багато, спосіб їх порівняти - проводити матчі.

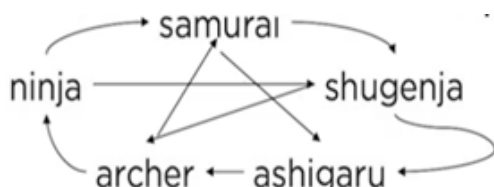


Рис. 4. Балансування нетранзитивних ігор за Casey O'Donnell [16]

На відміну від класифікування, кінцевим

продуктом якого є класифікаційна таблиця, та інфографіки, яка загалом представляє логіко-семантичні зв'язки між об'єктами (діаграми Ейлера та Венна, графіки Ганта тощо), UML-моделі мають більше функціональних можливостей для відображення об'єктів реальної та віртуальної реальності через застосування графічних та структурованих вербальних описів.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ.

Таким чином, можна констатувати, що UML-моделювання має стати основою для проектування та балансування сценаріїв комп'ютерної гри значно раніше за створення програмного продукту гри, тобто ще на етапі геймдизайну. Широкі можливості UML-моделей для відображення реальних подій та об'єктів у віртуальному світі (аналіз складу та кількості дій, впливів та послідовностей, самовизначення та взає-

мозв'язків) можуть бути ефективно реалізовані при проектуванні комп'ютерних ігор, що містять у собі функції проектного менеджменту, бізнес стратегій, управлінського консалтингу, культурної ідентичності тощо.

Результати цієї роботи будуть ви-

користані в проекті ЄС Erasmus+KA2 «GameHub: університетсько-підприємницьке співробітництво в ігровій індустрії в Україні (№ 561 728-EPP-1-2015-1-ES-EPPKA2-SBHE-JP) для розробки комп'ютерних ігор.

Список використаних джерел:

- [1] Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1), 20-20.
- [2] Буч, Г., Рамбо, Д., Якобсон, А. (2015). *Введение в UML от создателей языка*. ДМК Пресс.
- [3] Gumennykova, T., Blazhko, O., Luhova, T., Troianovska, Y., Melnyk, S., & Riashchenko, O. (2019). Gamification features of STREAM-education components with education robotics. *Applied Aspects of Information Technology*, 2(01).
- [4] Blazhko, O., & Luhova, T. (2018). Features of using the canvas-oriented approach to game design. *Applied Aspects of Information Technology*, 1(01), 66-77. DOI: 10.15276/aaait.01.2018.5.
- [5] Dormans, J. (2011). Simulating mechanics to study emergence in games. *Workshops at the Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*. 2011. Retrieved from <https://www.aaai.org/ocs/index.php/AIIDE/AIIDE11WS/paper/download/4093/4448>.
- [6] *The Game Design Tool. Machinations is the browser-based platform to design, balance and simulate game systems*. Retrieved from <https://machinations.io/>.
- [7] Lewis, C., & Whitehead, J. (2011, May). The whats and the whys of games and software engineering. *In Proceedings of the 1st international workshop on games and software engineering*. ACM. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.409.2105&rep=rep1&type=pdf>.
- [8] Меженин, А. В. (2016). Виртуальные 3D среды как средство верификации и тестирования при проектировании. *Приоритетные научные направления: от теории к практике*, (21).
- [9] Суслов, Н. В. (2008). Принципы построения виртуального обучающего пространства. *Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление*, 5 (65). Изъято из <https://cyberleninka.ru/article/n/printspiy-postroyeniya-virtualnogo-obuchayuschego-prostranstva>.
- [10] Antoshchuk, S., Arsirii, O., Blazhko, O., Troianovska, Y., & Luhova T. (2019). Method for Detecting Errors in the Design of Virtual Environments. *The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 18-21 September, Metz, France.
- [11] Distefano, S., Scarpa, M., & Puliafito, A. (2010). From UML to Petri nets: The PCM-based methodology. *IEEE Transactions on Software Engineering*, 37(1). DOI: 10.1109/TSE.2010.10
- [12] Harel, D. (2001). From play-in scenarios to code: An achievable dream. *Computer*, 34(1). DOI: 10.1109/2.895118.
- [13] Лугова, Т. А., & Лись, Д. А. (2018). UML-моделі як основа проектування комп'ютерних ігор. In *Міжнародна наукова конференція студентів та молодих вчених «Modern Information Technology 2018—Сучасні Інформаційні Технології 2018*. Одеса: Екологія.
- [14] Casey O'Donnell Principles of Game Design: Coursera Michigan State University. Retrieved from <https://ru.coursera.org/learn/gamesdesign>.
- [15] Гимельрейх С. *Методология балансировки игр*. Retrieved from <https://gdcuffs.com/balance-methods/>.
- [16] O'Donnell, C. (2014). *Developer's dilemma: The secret world of videogame creators*. MIT press.

Author(-s):

Luhova T., PhD, Candidate of Art History Sciences, Associate Professor of the Information Activity and Media Communications Department
Odessa National Polytechnic University
UKRAINE

Lys D., student of the Institute of Computer Systems
Odessa National Polytechnic University
UKRAINE