

В. В. Войтко
А. В. Денисюк
О. В. Гаврилюк
Н. Є. Барчук
Туровець А.В.

РОЗРОБКА ЧАСОВОГО ВІДСЛІДКОВУВАННЯ ДІЙ ДЛЯ СПІДРАНУ У ВИГЛЯДІ МОДУ З ВИКОРИСТАННЯМ FABRIC API ТА MMT

Вінницький національний технічний університет

Анотація

Проведено аналіз технологій часового відслідковування у спідранах та порівняння існуючих аналогів. Розроблено метод побудови таймера спідрану, що включає аналіз вхідних даних, обґрунтування вибору інтерфейсу, розробку структури та алгоритмів системи. Здійснено програмну реалізацію продукту у вигляді моду для відеоігор з використанням Fabric API та MMT. Визначено методiku тестування та проведено перевірку програмного забезпечення.

Ключові слова:

Спідран, таймер, мод, Fabric API, MMT, відеоігри, відслідковування часу.

Abstract

An analysis of time tracking technologies in speedruns and a comparison of existing analogues was conducted. A method for building a speedrun timer was developed, including input data analysis, interface selection justification, system structure, and algorithm development. The software product was implemented as a mod for video games using Fabric API and MMT. A testing methodology was defined, and the software was tested.

Keywords:

Speedrun, timer, mod, Fabric API, MMT, video games, time tracking.

Вступ

У сучасному світі відеоігри та спідран є ключовими елементами цифрової культури. Виникнувши у 90-х, спідран перетворився на конкурентну дисципліну з правилами та спільнотами, де гравці демонструють майстерність, використовуючи «глітчi» для швидшого проходження [1]. Ця практика стрімко набирає популярність як окремий вид кіберспорту.

Події на кшталт Games Done Quick (GDQ) привертають мільйони глядачів, демонструючи навички гравців та збираючи кошти на благодійність. Популярність спідрану не обмежується лише старими іграми; сучасні розробники інтегрують спеціальні режими, залучаючи нову аудиторію [2].

Підтримка ігрових розробників та розширення турнірної інфраструктури, зокрема через онлайн-платформи на кшталт Twitch, є ключовим аспектом розвитку спідрану, роблячи його доступнішим та популяризуючи серед новачків [3]. Ця дисципліна також розвиває критичне мислення та реакцію гравців

завдяки необхідності глибокого знання ігрової механіки. Нові технології, такі як штучний інтелект і машинне навчання, можуть вдосконалити стратегії спідранерів [4]. Важливо, що спідран, як і класичний спорт, вимагає значної фізичної та психічної витривалості, постійних тренувань та концентрації.

З огляду на зростаючу популярність спідрану та потребу у надійних інструментах для часового відслідковування дій гравців, виникає актуальність розробки спеціалізованих рішень. Існуючі методи часто покладаються на зовнішні таймери або ручне відслідковування, що може призвести до неточностей та незручностей. Необхідність інтеграції таймера безпосередньо в ігровий процес, як модифікації (моду), стає очевидною для забезпечення максимальної точності та зручності для спідранерів. Це дослідження спрямоване на розробку такого рішення з використанням сучасних API для моддингу, таких як Fabric API та MMT.

Головним завданням є розробка системи часового відслідковування дій для спідрану, яка дозволить гравцям точно вимірювати час проходження ігрових етапів, враховувати особливості ігрової механіки та автоматизувати процес запису результатів, що забезпечить підвищення ефективності тренувань та чесність змагань.

Порівняння аналогів та розробка системи

Сьогодні використовується низка застосунків та інструментів для часового відслідковування у спідранах. Розглянемо популярні ресурси як аналоги розроблюваної системи для часового відслідковування дій у спідранах: LiveSplit, OBS Game Timer та вбудовані ігрові таймери.

LiveSplit – це одне з найпопулярніших програмних рішень для спідранерів, що надає широкі можливості для точного відслідковування часу, розділення на сегменти (спліти), порівняння з особистими рекордами та світовими таблицями лідерів. Його перевагами є велика гнучкість, наявність багатьох плагінів та активна спільнота. Однак, LiveSplit є зовнішнім застосунком, що вимагає ручної синхронізації або використання хоткеїв, і не взаємодіє безпосередньо з ігровим процесом для автоматичного визначення подій.

OBS Game Timer – це плагін або скрипт для OBS Studio, який дозволяє відображати таймер безпосередньо на стрімі. Він зручний для стримерів, оскільки інтегрується з трансляцією. Проте, його функціонал обмежується відображенням часу, і він не призначений для глибокого аналізу ігрових подій чи автоматичного відслідковування дій гравця. Зазвичай, він працює на основі ручного запуску/зупинки або простих тригерів.

Вбудовані ігрові таймери – деякі відеоігри мають власні вбудовані таймери, які автоматично відслідковують час проходження рівнів або гри в цілому. Це забезпечує максимальну точність та інтеграцію. Однак, такі таймери є ексклюзивними для конкретної гри, і їх функціонал обмежений тим, що передбачено розробниками гри. Вони не можуть бути адаптовані для інших ігор або для специфічних потреб спідранерів, таких як відстеження окремих подій або сегментів.

Після детального розгляду аналогів та їх відмінностей, переваги і недоліки ресурсів зведено в окрему таблицю 1.

Таблиця 1 — Порівняльний аналіз аналогів

Функціонал	LiveSplit	OBS Game Timer	Вбудовані ігрові таймери	Власна розробка
Автоматичне відслідковування ігрових подій	0	0	1	1
Детальне сегментування (спліти)	1	0	0	1

Продовження таблиці 1

Функціонал	LiveSplit	OBS Game Timer	Вбудовані ігрові таймери	Власна розробка
Інтеграція безпосередньо в ігровий процес	0	0	1	1
Зручний та настроюваний інтерфейс	1	1	0	1
Можливість розширення функціоналу (моддинг)	1	0	0	1
Сумарний коефіцієнт	3	1	2	5

Проаналізувавши таблицю 1, можна відзначити, що власна розробка є кращою і має вищий сумарний коефіцієнт у порівнянні з аналогами за наведеними критеріями. За проведеною оцінкою власна розробка є кращою за «LiveSplit» на 66%, за «OBS Game Timer» на 400% та за «Вбудовані ігрові таймери» на 150%.

Велика різниця у сумарному коефіцієнті свідчить про те, що кожна система, розглянута як аналог, спеціалізується на вузькому спектрі функціоналу, що, в свою чергу, впливає на її придатність для комплексних потреб спідранерів. Тому актуальною є власна розробка системи, орієнтованої на розширення функціоналу, повну інтеграцію в ігрове середовище та забезпечення максимальної точності та зручності.

Враховуючи переваги й недоліки систем-аналогів, було визначено функціонал власної системи часового відслідковування дій для спідрану. Блок-схему загального алгоритму роботи системи наведено на рис. 1.

Розроблена система акумулює такий функціонал:

- автоматичне відслідковування початку та завершення проходження етапів/гри;
- детальне сегментування часу (спліти) для ключових подій;
- відображення таймера безпосередньо в ігровому інтерфейсі;
- можливість збереження та порівняння результатів;
- налаштування та персоналізація відображення таймера;
- інтеграція з ігровою логікою через Fabric API та MMT для високої точності.

Діаграми станів є інструментом моделювання, який використовується для опису поведінки системи або її частин шляхом визначення різних станів об'єкта та переходів між цими станами.

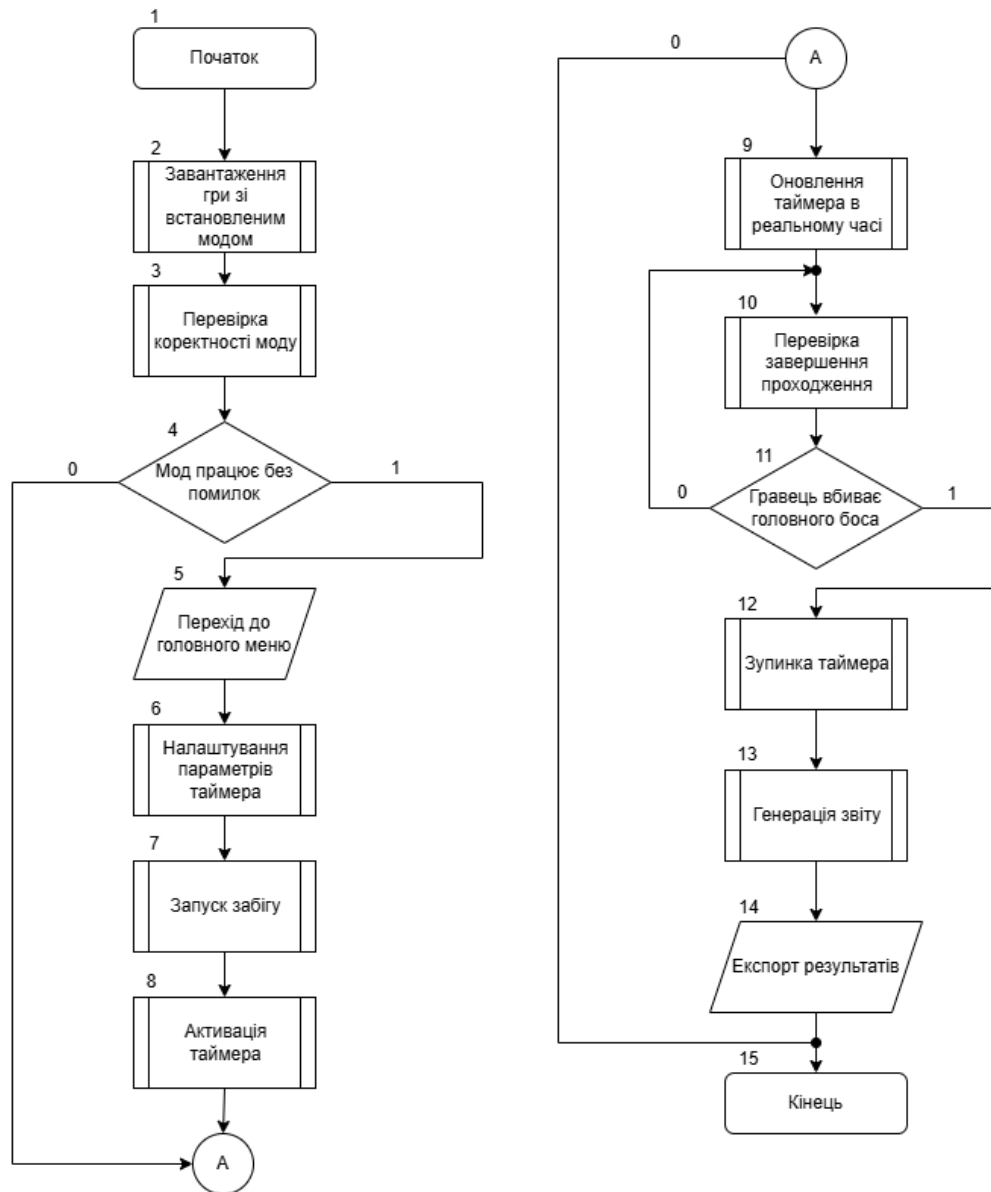


Рис. 1. Блок-схема роботи алгоритму часового відслідковування дій для швидкісного проходження

На рисунку 2 наведено діаграму станів модуля відслідковування ігрових подій під час швидкісного проходження гри.

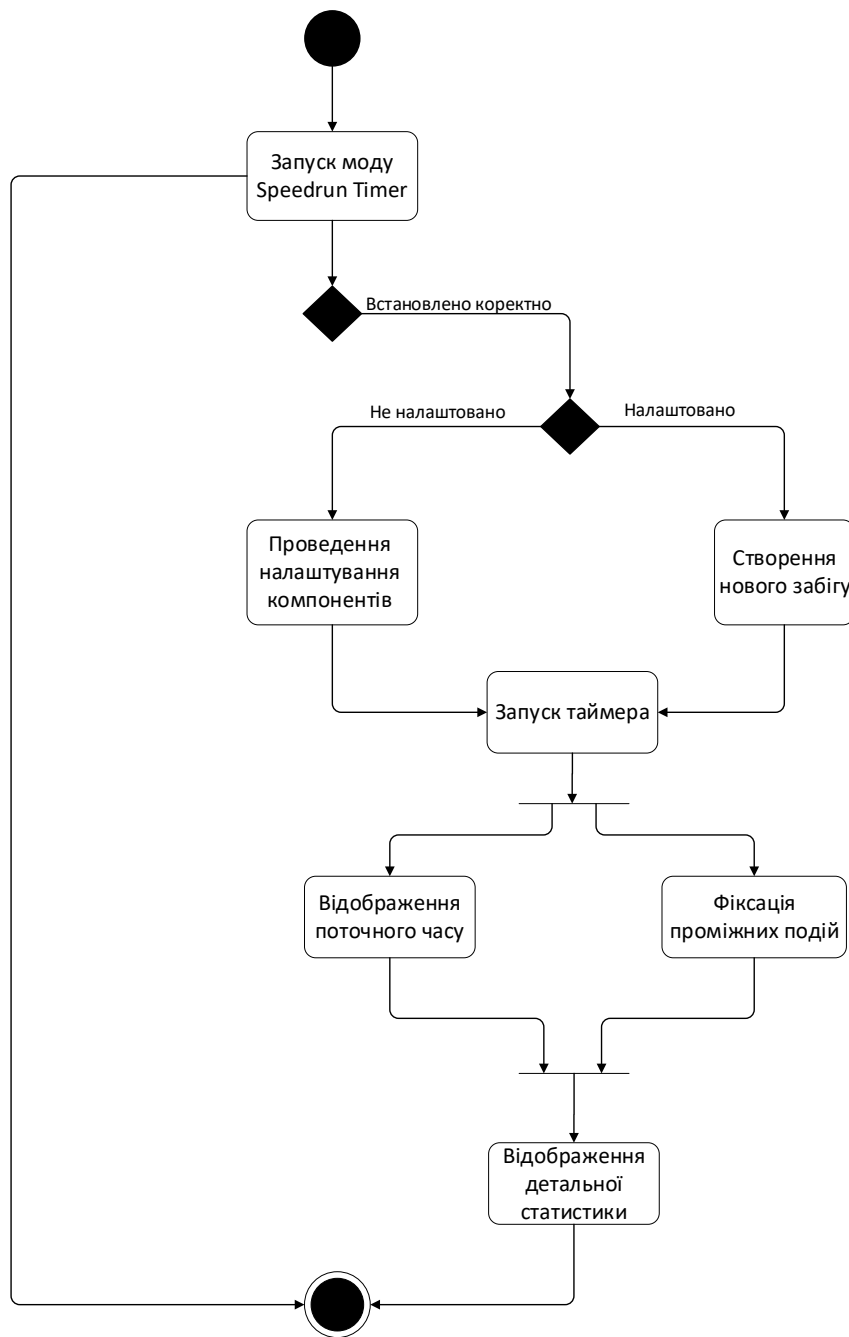


Рис. 2. Діаграма станів модуля відслідковування ігрових подій

Висновок

Розроблена система дозволяє користувачам інтерактивно взаємодіяти з таймером, отримуючи точну інформацію про проходження сегментів та загальний час. Система забезпечує автоматизований збір даних про ігрові події, що усуває необхідність ручного відслідковування та підвищує об'єктивність результатів. Також забезпечено можливість інтеграції з ігровим процесом для зручності спідранерів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Witkowski E. Speedrunning: Constructing Competitive Game Play in The Digital Age. Journal of Gaming & Virtual Worlds. 2018;10(2):131-147.
2. Scully-Blaker R. A Practiced Practice: Speedrunning through Space with Deleuze. Game Studies. 2014;14(1).
3. Therrien C. Speedrunning as Perfectionist Play. DiGRA 2015 Conference Proceedings. 2015.
4. Newman J. Playing with Videogames. New York: Routledge; 2008. 224 p.

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: dekanfki@i.ua.

Денисюк Алла Василівна – асистент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: alladen@ua.fm.

Гаврилюк Олена Віталіївна – асистент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: kafedra_pz_2105@ukr.net.

Барчук Наталія Євгенівна – асистент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: kafedra_pz_2105@ukr.net.

Туровець Артем Володимирович – студент групи ЗПІ-21б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: tyomichchanell@gmail.com

Viktoriia Voitko – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dekanfki@i.ua.

Alla Denisyuk – Assistant of Software Engineering department, Vinnytsia National Technical University, Vinnytsia, e-mail: alladen@ua.fm.

Olena Gavruhuik – Assistant of Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: kafedra_pz_2105@ukr.net.

Natalia Barchuk – Assistant of Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: kafedra_pz_2105@ukr.net.

Artem Turovets – student of group ЗПІ-21b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: tyomichchanell@gmail.com