

Магістерська дисертація Антоненко Івана Вікторовича

Автоматизація тестування продуктивності програмного забезпечення ігор

науковий керівник – к.т.н., доц. Шаповалова С.І.

Глобальний ринок ігрового програмного забезпечення



* Entertainment software sales only.

- **Предмет дослідження** - програмні засоби тестування програмного забезпечення.
- **Об'єктом дослідження** є процес тестування програмного забезпечення.
- **Метою дослідження** є створення алгоритмічної бази та відповідного програмного забезпечення автоматизованого тестування продуктивності ігрових застосунків.

Наукова новизна:

- Удосконалено спосіб визначення порушення норми характеристики продуктивності на базі збереження даних про їх значення та екстремуми з попередніх кадрів для уникнення реєстрування схожих проблем на сусідніх кадрах;
- удосконалено кореляцію переміщення камери на базі кубічної інтерполяції для отримання більш плавних переходів та уникнення стрибків значень характеристик.

Завдання дослідження:

- провести аналіз базових методів тестування програмного забезпечення для визначення специфіки автоматизованого тестування ігрового програмного забезпечення та отримання основних характеристик продуктивності;
- провести дослідження програмних засобів тестування програмного забезпечення для визначення недоліків та/або відсутність необхідного функціоналу автоматизованого тестування програмного забезпечення ігор для подальшої реалізації;
- розробити метод визначення порушення норми характеристики продуктивності для уникнення реєстрування схожих проблем на сусідніх кадрах;
- удосконалити кореляцію переміщення камери для отримання більш плавних переходів та уникнення стрибків значень характеристик;
- на основі запропонованих рішень розробити програмну реалізацію автоматизованого тестування ігрового програмного забезпечення;
- провести обчислювальні експерименти для визначення коректності тестування.

Типи тестування



Системи тестування

Назва	Виробник	Форма представлення
NvidiaOpenAutomate	NVidia	Фреймворк з базовими функціями для тестування
AutoTestManager	Epic Games	Набір базових функцій у ігровому двигуні Unreal Engine
Selenium	ThoughtWorks	Програмне середовище для тестування Web-додатків

Стандартний набір інструментів для тестування

Назва інструменту	Функції статистики	Джерело інформації для функцій статистики
Консольні команди двигуна	<ul style="list-style-type: none">- Frames Per Seconds- Кількість текстур- Об'єм текстур- Об'єм оперативної пам'яті- Об'єм відео пам'яті	<ul style="list-style-type: none">- Двигун- Операційна система
Диспетчер завдань	<ul style="list-style-type: none">- Навантаження на центральний процесор- Оперативна пам'ять- Диск	<ul style="list-style-type: none">- Операційна система
Perf + WPA	<ul style="list-style-type: none">- Різниця по пам'яті між двома точками	<ul style="list-style-type: none">- Операційна система
Very Sleepy	<ul style="list-style-type: none">- Час виконання функцій	<ul style="list-style-type: none">- Операційна система
Razor CPU\GPU	<ul style="list-style-type: none">- Профілювання потоків та кадрів	<ul style="list-style-type: none">- Операційна система

Основні характеристики ігрового програмного забезпечення

Характеристика	Позначення	Вплив
Кількість кадрів в секунду	FPS	Візуальне сприйняття гри
Об'єм оперативної пам'яті	M_o	Кількість одночасно завантажених ігрових об'єктів
Об'єм відеопам'яті	M_v	Кількість одночасно завантажених текстур, шейдерів
Об'єм текстурного пулу	T	Якість одночасно завантажених текстур
Час завантаження ігрових рівнів	t	Швидкість переходу користувача до ігрового процесу
Кількість примітивів для відображення	O	Складність створюваних сцен

Проблеми тестування продуктивності програмного забезпечення ігор

	Назва	Причина
1	Визначення порушення норми характеристик: FPS, M_o , M_v , T, t, O	Неможливість відмінності одного і того самого порушення на різних кадрах
2	Кореляція переміщення камери	Відтворення одного треку на різних конфігураціях платформи

Задача визначення порушення норми характеристики

$$v \in \{FPS, M_o, M_v, T, t, O\}$$

- Тимчасові змінні: t_c, v_l ,

де t_c – поточне значення часу перевірки актуальності порушення, v_l – найгірше значення характеристики.

- Вхідні дані: $v_c, t_d, lim_v, lim_d, t$,

де v_c – поточне значення характеристики, t_d – різниця часу між кадрами, lim_v – крайове значення характеристики, lim_d – допустима різниця від крайового значення характеристики, t – границя часу перевірки актуальності порушення.

- Вихідні дані: v_c, v_{min}, v_{max} ,

де v_c – поточне значення характеристики, v_{min} та v_{max} – екстремуми значення характеристики.

Алгоритм методу визначення порушення норми характеристики



Задача кореляції переміщення камери

- Тимчасові змінні: $i, v_i, t_i, v_{i-1}, t_{i-1}$,

де i – ітератор проходу по всім елементам масиву значень позицій камери, v_i – позиція камери на ітерації i , t_i – час відтворення треку на ітерації i , v_{i-1} – позиція камери на ітерації $i - 1$, t_{i-1} – час відтворення треку на ітерації $i - 1$.

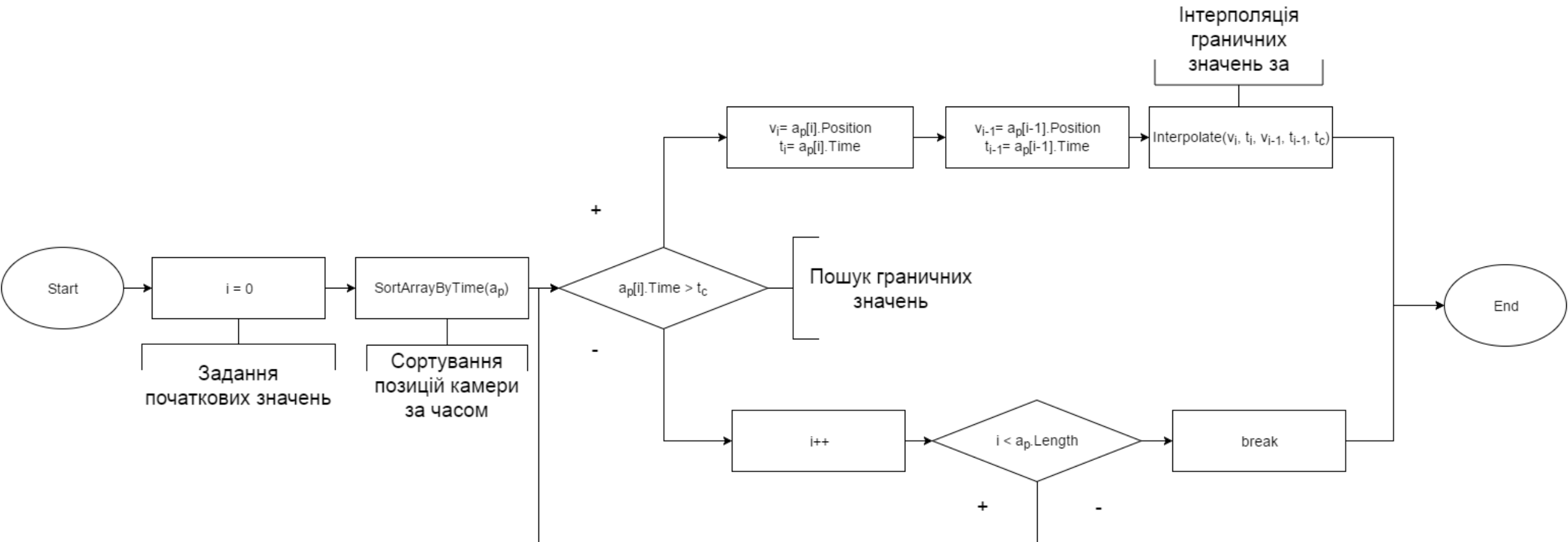
- Вхідні дані: a_p, t_c ,

де a_p – масив значень позицій камери $a_p[i]$. *Position* у деякий момент часу $a_p[i]$. *Time* розміром a_p . *Length*, t_c – поточне значення часу відтворення запису.

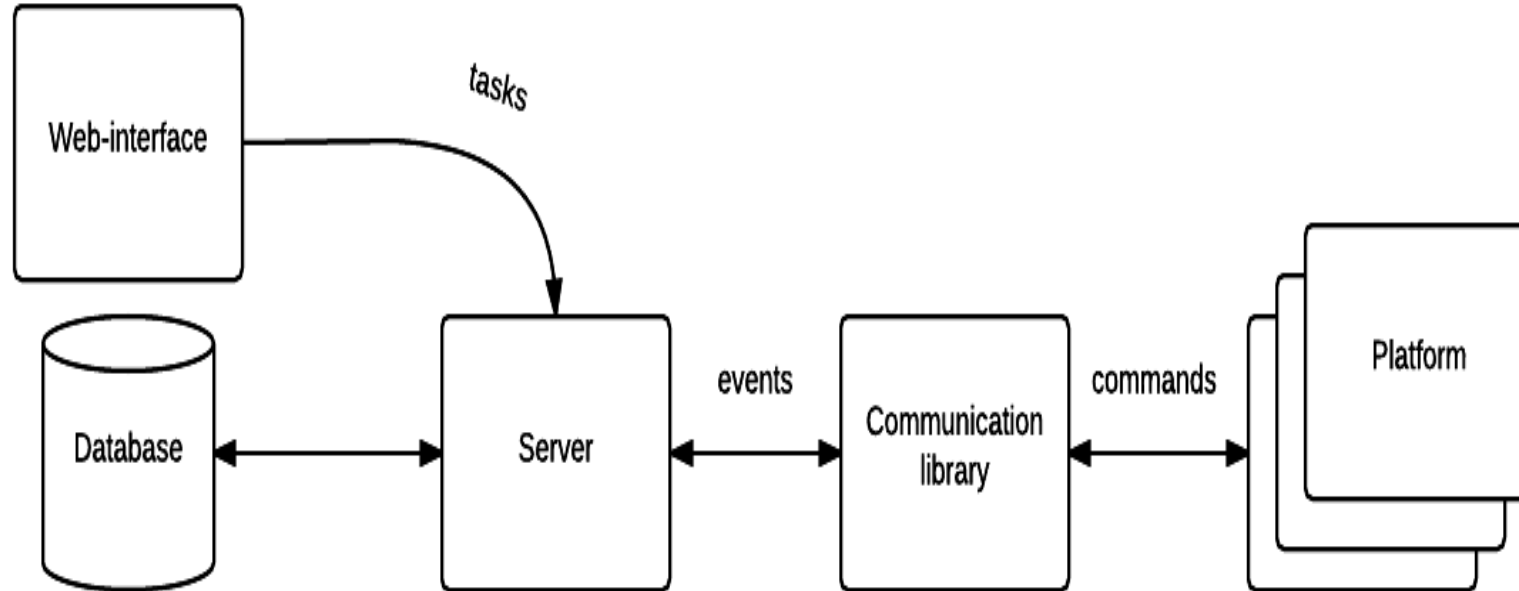
- Вихідні дані: v_c ,

де v_c – значення позиції камери у поточний час відтворення запису.

Алгоритм методу кореляції переміщення камери



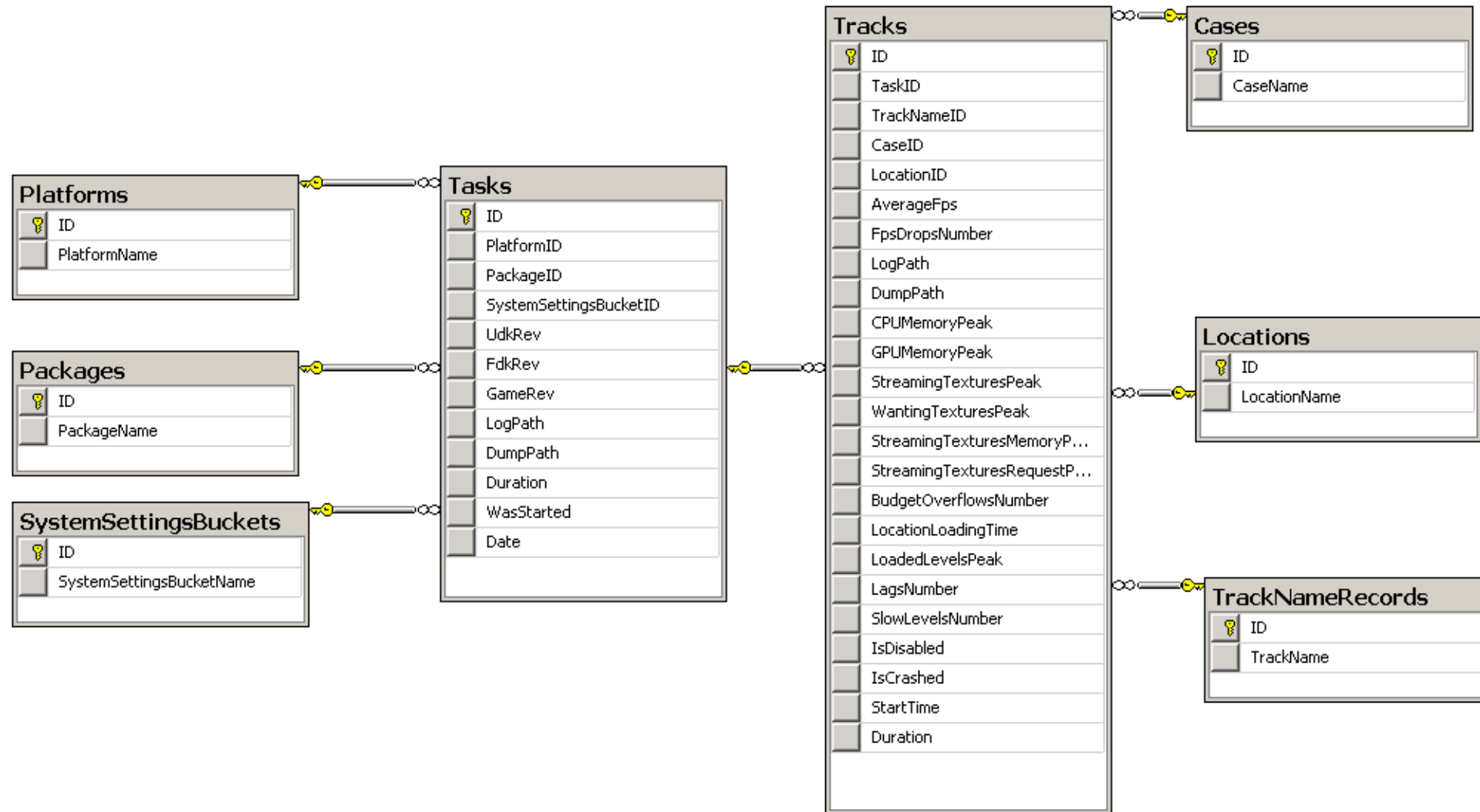
Архітектура системи



Список технологій

Назва компоненту	Назва технологій	Визначальна перевага
Бібліотека формування та відправки повідомлень	C++	Можливість зручного додавання бібліотеки у більшість ігрових двигунів
Бібліотека координації ігрового додатку	C++	Можливість зручного додавання бібліотеки у більшість ігрових двигунів
Веб-інтерфейс користувача	ASP.NET, C#, JavaScript	Провідні технології для створювання веб-сторінок на платформі .NET
База даних	MSSQL	Потужна база даних, що має зручну інтеграцію з платформою .NET
Система обробки повідомлень	C#	Мова високого рівня платформи .NET, що дозволяє проводити зручну комунікацію з усіма іншими компонентами
Бібліотека комунікації з платформою	C++, C++/CLI, Python	Мова програмування задається кожною платформою

Фрагмент моделі бази даних – “Системні зв'язки”



Фрагмент моделі бази даних – “Порушення”

StreamingTexturesRequestPeaks	
ID	
TrackID	
WantingTexturesValue	
ScreenshotPath	
TrackTime	

MemoryPeaks	
ID	
TrackID	
MemoryPeakTypeID	
MemoryPeakValue	
ScreenshotPath	
TrackTime	

Levels	
ID	
LevelName	

LevelLoadingLags	
ID	
LevelLoadingTimeID	
FpsValue	
ScreenshotPath	
TrackTime	

FpsDrops	
ID	
TrackID	
FrameUnitStat	
GameUnitStat	
DrawUnitStat	
GpuUnitStat	
ScreenshotPath	
PrfScreenshotPath	
GameprofilePath	
DropTypeID	
TrackTime	

StreamingTexturesBudgetOverflows	
ID	
TrackID	
OverBudgetValue	
ScreenshotPath	
TrackTime	

LevelLoadingTimes	
ID	
TrackID	
LevelNameID	
LevelLoadingTime	
AverageFps	

DropTypes	
ID	
DropTypeName	

MemoryPeakTypes	
ID	
MemoryPeakTypeName	

Обчислювальний експеримент визначення порушення норм характеристики

Назва характеристики	Позначка	Номер ітерації	Час виконання, хв	Кількість порушень
Кількість кадрів в секунду	FPS	1	33.4	17
		2	25.3	11
		3	23.7	5
		4	21.1	2
Об'єм оперативної пам'яті	M _o	1	33.4	7
		2	25.3	5
		3	23.7	3
		4	21.1	2

Обчислювальний експеримент кореляції переміщення камери - 1

Конфігурація: AMD Jaguar 8 Cores, AMD 7870, 5 GB GDDR5

Номер етапу	Виконувач	Час виконання, хв	Кількість порушень	Не знайдені	Зайві
1	Тестувальник	15.7	7	5	0
2	Система без методу	28.2	12	0	0
3	Система з методом	28.2	12	0	0

Обчислювальний експеримент кореляції переміщення камери - 2

Конфігурація: Intel® Core™ i3-3220 CPU @ 3.3 GHz, NVIDIA GeForce GTX 560, 8 GB DDR3

Номер етапу	Виконувач	Час виконання, хв	Кількість порушень	Не знайдені	Зайві
1	Тестувальник	19.3	14	5	0
2	Система без методу	41.8	25	0	6
3	Система з методом	30.1	19	0	0

Висновки

- Проведено аналіз базових методів тестування. Обґрунтовано використання grey box testing для автоматизації тестування продуктивності програмного забезпечення ігор.
- Досліджено специфіку тестування ігрового програмного забезпечення. Визначено базові та специфічні характеристики, які необхідно враховувати при тестуванні.
- Проведено аналіз існуючих систем автоматизованого тестування продуктивності ігор. Визначено, що відсутнє програмне забезпечення тестування системи реального часу з одночасним контролем заданої характеристики, що обумовлює необхідність розробки та автоматизації процесу тестування програмного забезпечення.
- Для коректної роботи автоматизованої системи тестування продуктивності програмного забезпечення ігор є важливою і актуальною можливістю відрізнити порушення на сусідніх кадрах у системах реального часу. Для цього було удосконалено спосіб визначення порушення норми характеристики на базі збереження даних про її значення та екстремуми з попередніх кадрів, що склало наукову новизну роботи.
- Для коректного тестування на стороні ігрового додатку є важливою і актуальною задачею плавного відтворення позицій камери. Для цього було удосконалено кореляцію переміщення камери на базі лінійної інтерполяції для отримання більш плавних переходів та уникнення стрибків значень характеристик, що склало наукову новизну роботи.
- Запропоновані методи ідентифікації норми порушення характеристики та кореляції переміщення камери, що відсутні в інших системах тестування продуктивності, реалізовано у відповідних бібліотеках.

Висновки

- На основі реалізації запропонованих методів розроблено автоматизовану систему тестування продуктивності програмного забезпечення ігор, що дозволяє користувачеві проводити ітераційний прогін тестів з аналізом статистики для поступового вдосконалення поточної версії гри.
- Розроблений програмний продукт може бути використаний у компаніях сфери розробки ігрового програмного забезпечення, які потребують щоденного ітераційного тестування продуктивності гри.
- Розроблено постановку завдань тестування продуктивності програмного забезпечення ігор для перевірки коректності запропонованих методів моніторингу характеристик.
- Доведена доцільність визначення продуктивності за запропонованим методом визначення порушення норми характеристики для вдосконалення програмного забезпечення гри.
- Доведено коректність методу кореляції переміщення камери для отримання більш плавних переходів та уникнення стрибків значень характеристик.
- Дослідження, які були проведені під час виконання магістерської дисертації, показали, що на певних етапах розробки моделі виникали деякі ідеї, на які у майбутньому слід було б зупинитись. Спосіб визначення порушення норми характеристики може використовувати більш широкий набір мета-даних, наприклад, інформацію про об'єкти сцени. Зважаючи на вищезазначене, напрямок робіт в цій області вбачається перспективним.
- Автоматизована система тестування продуктивності програмного забезпечення ігор була апробована в ТОВ «Інформаційні комунікації» та отримано акт впровадження.

Дякую за увагу!