

Розроблення ефективних компонентів комп'ютерних ігор шляхом використання патернів проектування

Кирил Сидоров
студент 4 року навчання
керівник: Бублик В.В

Опис дослідження



Предмет дослідження

- Комплекс патернів проектування
- Стратегії оптимізації оновлень
 - квантування часу
 - агрегування оновлень
 - системи бюджетування
 - системи контролю вагомості



Мета дослідження

1. Дослідження застосування патернів проектування в комп'ютерних іграх
2. Аналіз підходів до керування оновленнями
3. Реалізація компонентів та оцінка їх впливу на швидкодію ігор

Актуальність теми



Індустрія ігор
демонструє економічне
зростання



Оптимізація покращує
досвід користування

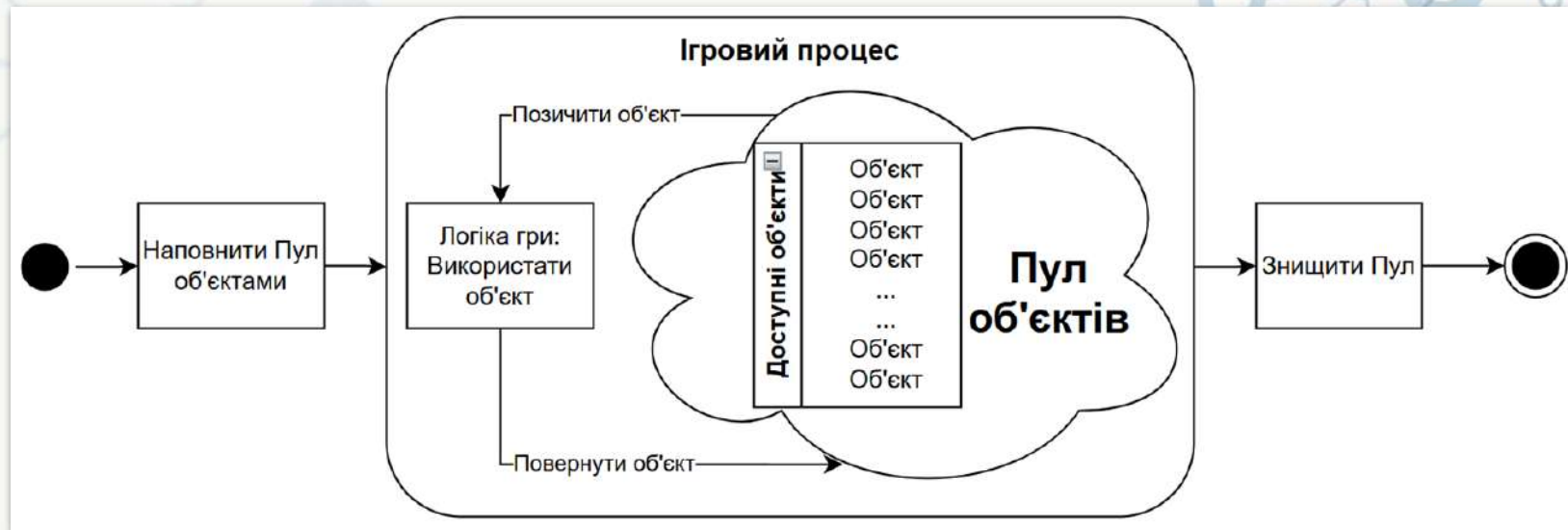


Недостатня швидкість
шкодить бізнесу



Низька доступність
спеціалізованої
інформації

Пул об'єктів



Підтримує різні типи об'єктів:

Дрібні
об'єкти

Ефекти

UI

Персонажі

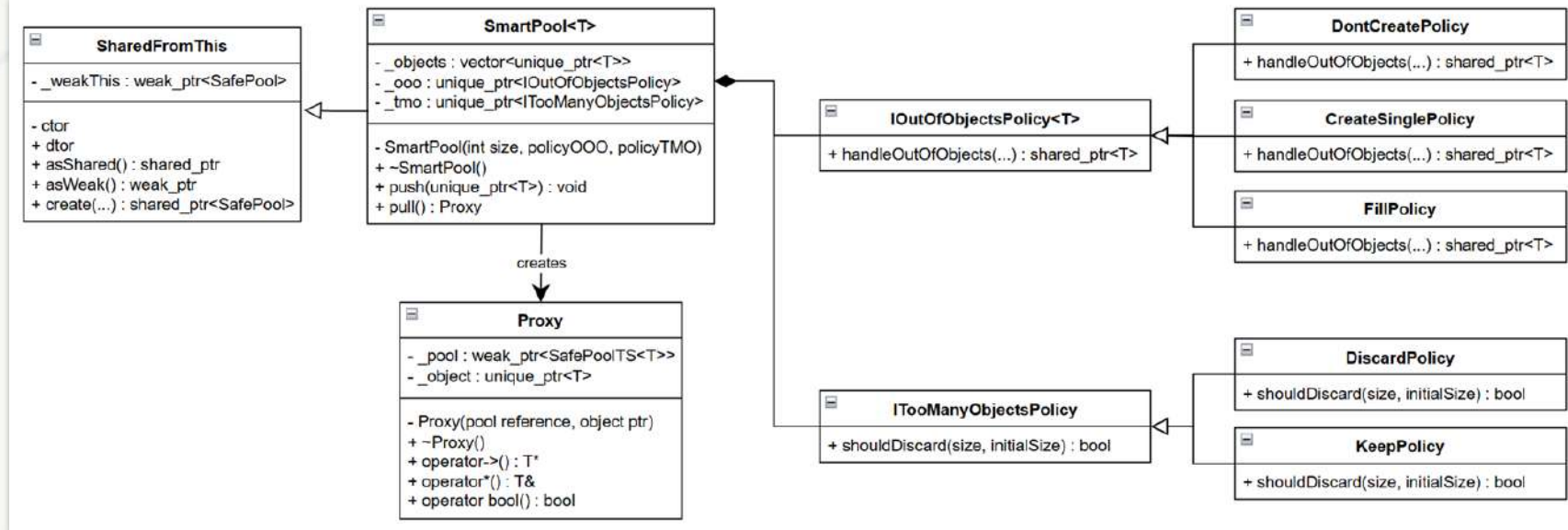
Системні об'єкти

Ащр “якщр...”?

- Що якщо об’єктів недостатньо?
- Що якщо об’єктів забагато?



Пул об'єктів + Стратегія



Ігровий цикл

```
extern bool g_GameExitRequested = false;

int main()
{
    initializeGame();

    while (!g_GameExitRequested)
    {
        processInput();
        updateGameObjects();
        render();
    }

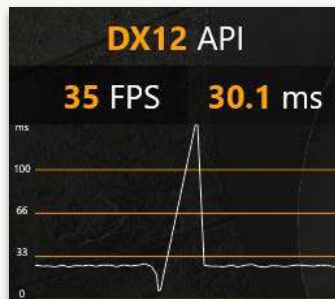
    deinitializeGame();
}
```

Завжди був...

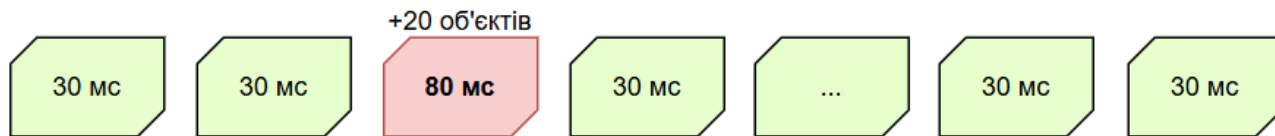
Це що, просто
нескінченний цикл?



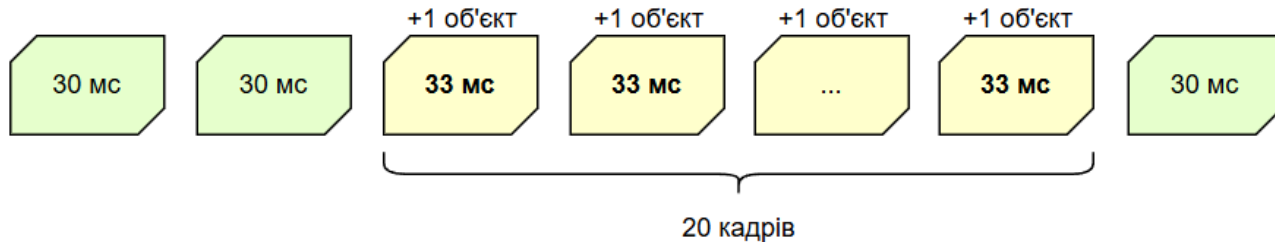
Квантування часу



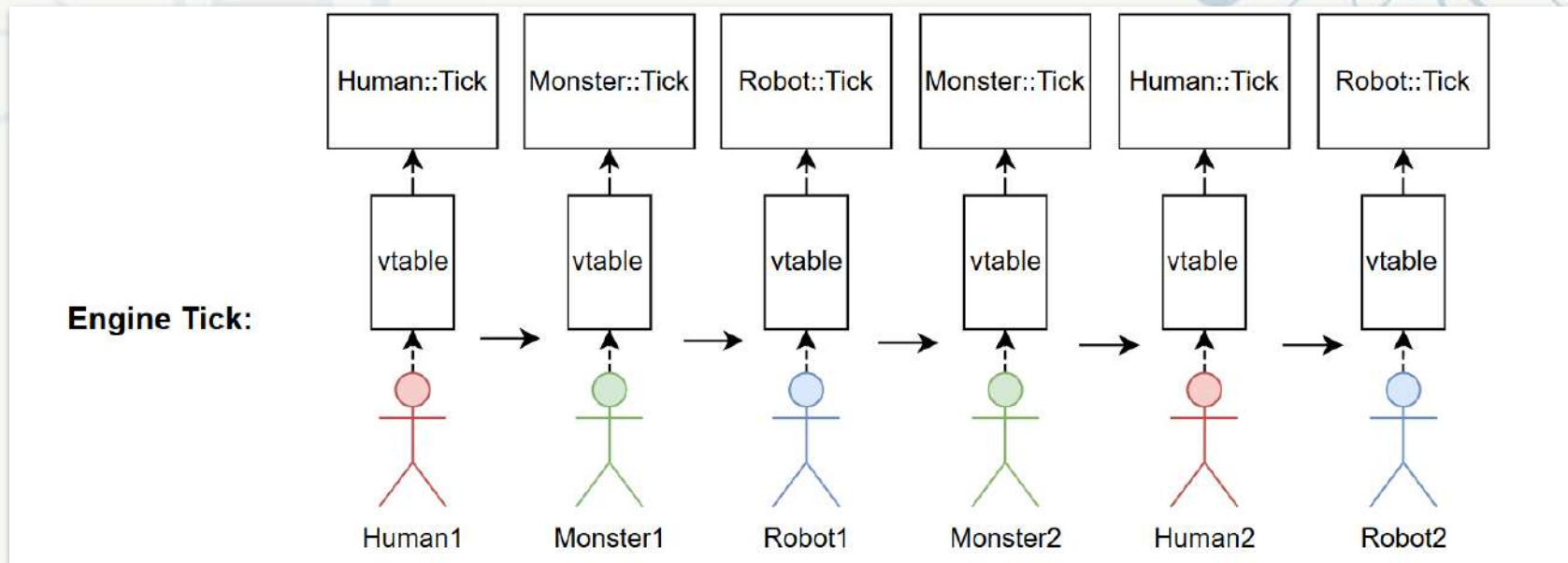
Без квантування часу:



З квантуванням часу:



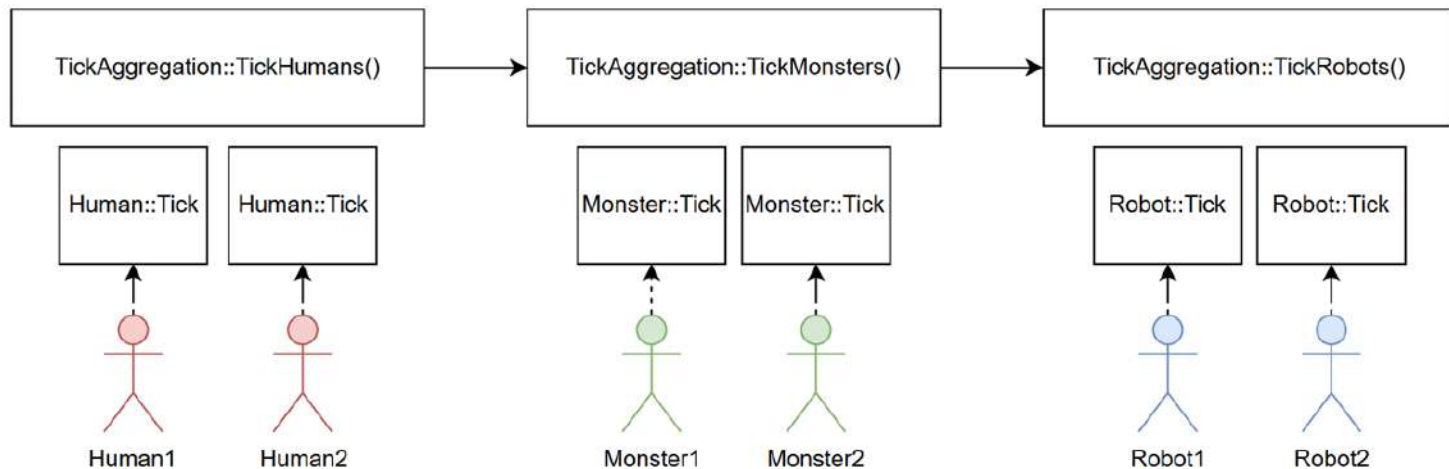
Агрегування оновлень



Неоптимізований варіант

Агрегування оновлень

Engine Tick:



Використання агрегування

Адаптивний контроль вагомості

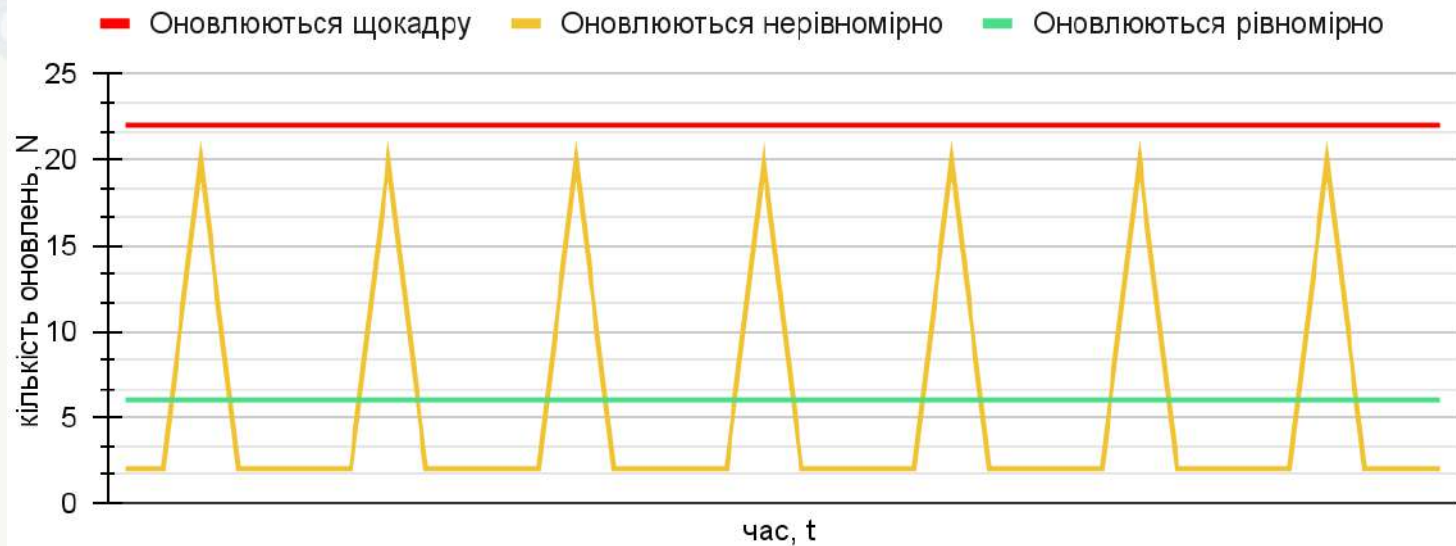
1. Визначити критерії вагомості
2. Обчислити значення вагомості x
3. Перетворити значення у частоту оновлень – $f(x)$

$$f(x) = \frac{x - a_{min}}{a_{max} - a_{min}} (b_{max} - b_{min}) + b_{min}$$

$[a_{min}; a_{max}]$ – допустимі значення вагомості

$[b_{min}; b_{max}]$ – допустимі значення частот оновлень

Бюджетування оновлень



Демонстраційний проєкт



Lyra Starter Game



Вимірювання продуктивності



Тестовий сценарій:

- 2 команди
- 10 ботів vs 9 ботів + 1 гравець
- умова перемоги: 50 очок
- гравець постійно рухається
- прогрес завдяки ботам
- обмеження FPS – 120
- кількість повторень: 3 рази



Тестовий стенд:

- CPU: AMD Ryzen 7950X 16c/32t 3.00 GHz
- RAM: 192 GB DDR5
- GPU: NVIDIA GeForce RTX 4080 16 GB

Метрики продуктивності

Показник	Символ	Значення
Середній час основного потоку	GT , мс	8,59
Кадрів > 10 мс (~ 100 гц)	N₁₀ , шт	1703
Кадрів > 15 мс (~ 66 гц)	N₁₅ , шт	26,43
Кадрів > 30 мс (~ 33 гц)	N₃₀ , шт	0,76
Мін. використання ОЗП	M_{min} , МБ	2065
Середнє використання ОЗП	M_{avg} , МБ	2147
Макс. використання ОЗП	M_{max} , МБ	2192
Середня частота кадрів	FPS , Гц	105,7

Значення для порівняння:

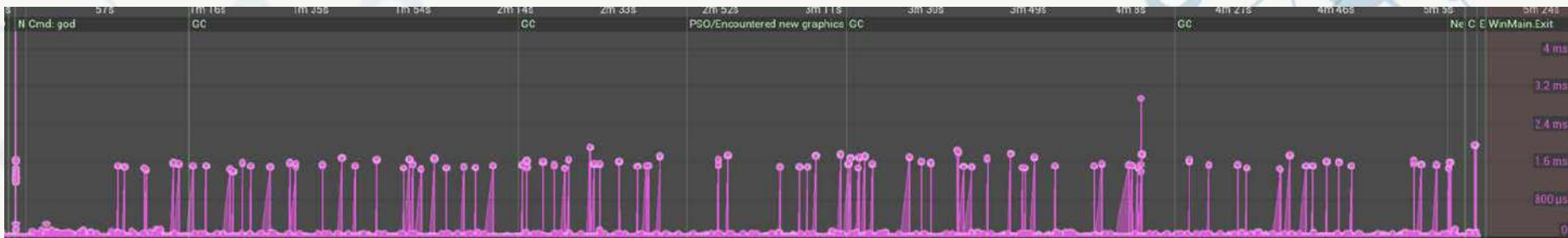
120 FPS ~ 8.33 мс/кадр

60 FPS ~ 16.66 мс/кадр

30 FPS ~ 33.33 мс/кадр



Профілювання створення акторів



Витрати часу на створення акторів

Timers		Counters	
SpawnActor			
Group by	Flat	Mode	Instance
Name	Name	Count	Excl
All (41 / 34,666)		2,728	421.2 ms
UWorld_SpawnActor		1,364	212 ms
SpawnActor_B_Hero_ShooterMannequin_C		119	180.1 ms
SpawnActor_B_WeaponFire_C		258	7.9 ms
SpawnActor_B_WeaponImpacts_C		379	7.5 ms
SpawnActor_GCNI_Character_DamageTaken_C		58	3.4 ms
SpawnActor_B_Pistol_C		128	2.7 ms
SpawnActor_B_WeaponDecals_C		192	2.5 ms
SpawnActor_B_Shotgun_C		62	1.4 ms
SpawnActor_B_Manny_C		62	894.2 µs
SpawnActor_B_Quinn_C		57	796.7 µs
SpawnActor_B_Rifle_C		36	773.5 µs
SpawnActor_GCNI_Death_C		6	470.8 µs
SpawnActor_GCNI_Character_Heal_C		2	276 µs
SpawnActor_GCNI_Teleporter_Activate_C		2	170.8 µs
SpawnActor_SpectatorPawn		1	141.9 µs
SpawnActor_GCNI_Launcher_Activate_C		1	100.5 µs
SpawnActor_GCNI_MatchDecided_C		1	71.5 µs
SpawnActor_AbstractNavData		0	0

Критичне місце:
створення 119 акторів персонажів
(витрачено: 180 мс за сесію)

Профілювання оновлень

Кількість кадрів: 24158

Тривалість сесії: 4 хв 2 сек

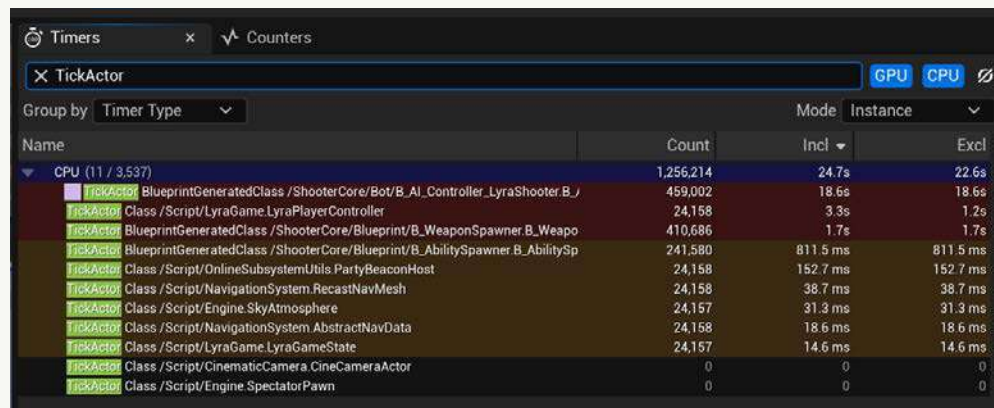
Основні споживачі:

Поведінка ботів: ~ 0.77 мс/кадр

Рух персонажів: ~ 1.06 мс/кадр

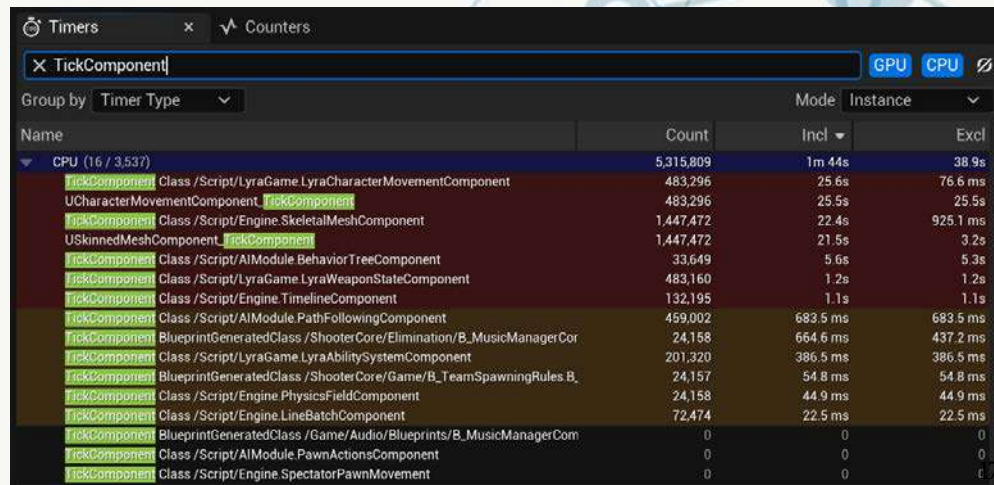
Анімації: ~ 0.93 мс/кадр

Точки видачі: ~ 0.11 мс/кадр



The screenshot shows the Unreal Engine Profiler interface with the 'Timers' tab selected. The search filter is 'TickActor'. The table displays performance data for various actors, sorted by count. The 'CPU' column shows the number of ticks, while 'Incl' and 'Excl' columns show time in milliseconds.

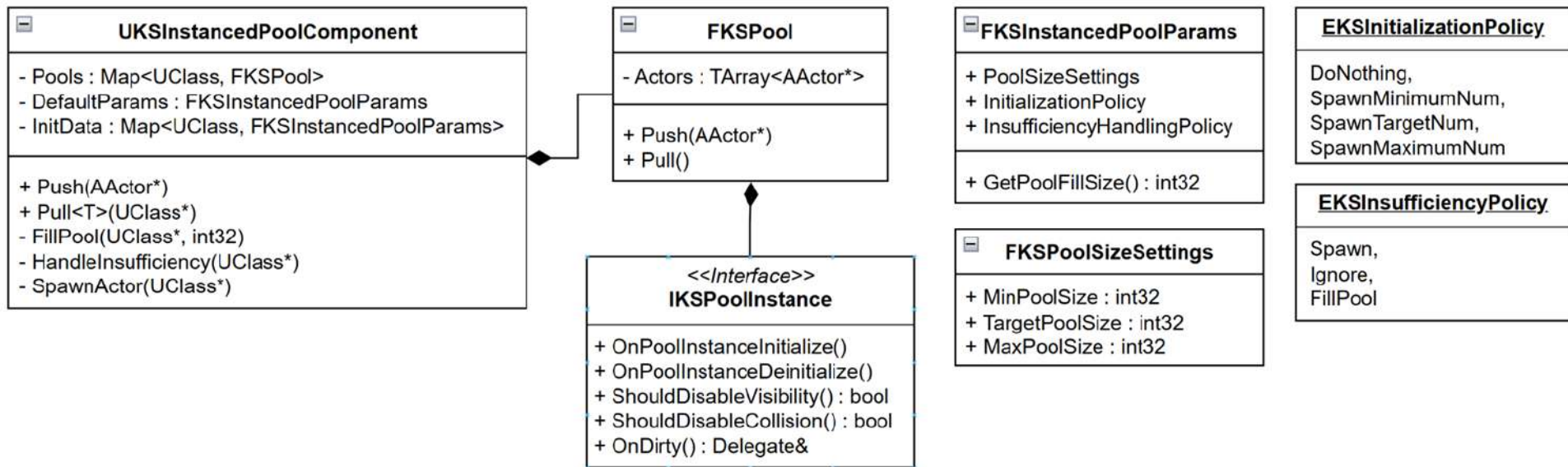
Name	Count	Incl	Excl
CPU (11 / 3,537)			
TickActor: BlueprintGeneratedClass /ShooterCore/Bot/B_AI_Controller_LyraShooter_B_	1,256,214	24.7s	22.6s
TickActor: Class /Script/LyraGame.LyraPlayerController	459,002	18.6s	18.6s
TickActor: BlueprintGeneratedClass /ShooterCore/Blueprint/B_WeaponSpawner_B_WeaponSpawner	24,158	3.3s	1.2s
TickActor: BlueprintGeneratedClass /ShooterCore/Blueprint/B_AbilitySpawner_B_AbilitySpawner	410,686	1.7s	1.7s
TickActor: BlueprintGeneratedClass /ShooterCore/Blueprint/B_AbilitySpawner_B_AbilitySpawner	241,580	811.5 ms	811.5 ms
TickActor: Class /Script/OnlineSubsystemUtils.PartyBeaconHost	24,158	152.7 ms	152.7 ms
TickActor: Class /Script/NavigationSystem.RecastNavMesh	24,158	38.7 ms	38.7 ms
TickActor: Class /Script/Engine.SkyAtmosphere	24,157	31.3 ms	31.3 ms
TickActor: Class /Script/NavigationSystem.AbstractNavData	24,158	18.6 ms	18.6 ms
TickActor: Class /Script/LyraGame.LyraGameState	24,157	14.6 ms	14.6 ms
TickActor: Class /Script/CinematicCamera.CineCameraActor	0	0	0
TickActor: Class /Script/Engine.SpectatorPawn	0	0	0



The screenshot shows the Unreal Engine Profiler interface with the 'Timers' tab selected. The search filter is 'TickComponent'. The table displays performance data for various components, sorted by count. The 'CPU' column shows the number of ticks, while 'Incl' and 'Excl' columns show time in milliseconds.

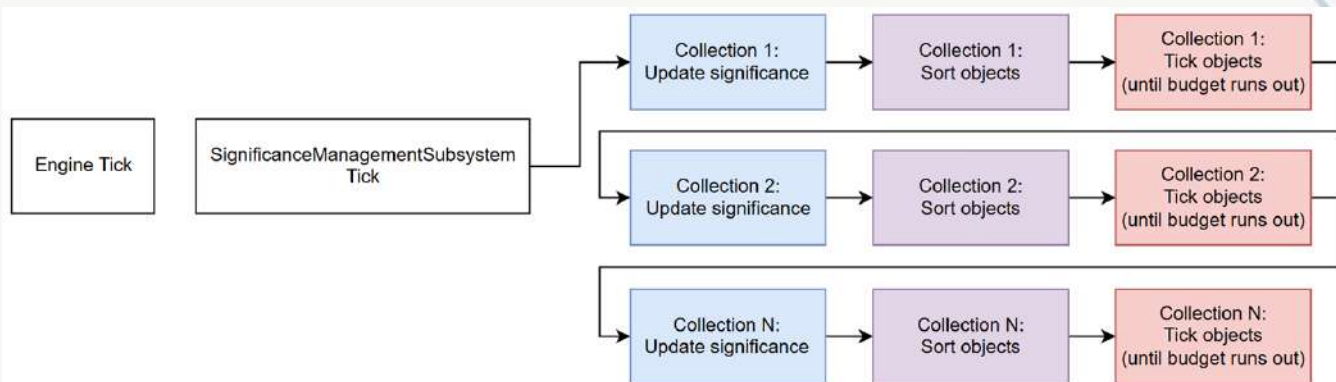
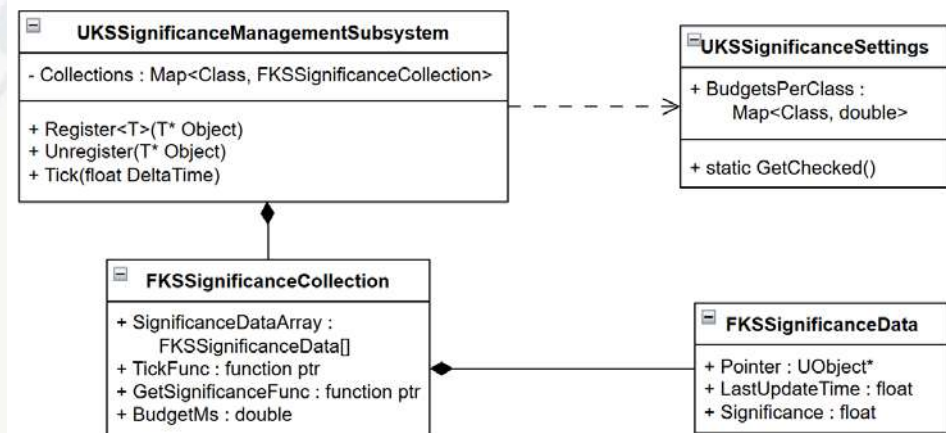
Name	Count	Incl	Excl
CPU (16 / 3,537)			
TickComponent: Class /Script/LyraGame.LyraCharacterMovementComponent	483,296	25.6s	76.6 ms
UCharacterMovementComponent: TickComponent	483,296	25.5s	25.5s
TickComponent: Class /Script/Engine.SkeletalMeshComponent	1,447,472	22.4s	925.1 ms
USkinnedMeshComponent: TickComponent	1,447,472	21.5s	3.2s
TickComponent: Class /Script/AIModule.BehaviorTreeComponent	33,649	5.6s	5.3s
TickComponent: Class /Script/LyraGame.LyraWeaponStateComponent	483,160	1.2s	1.2s
TickComponent: Class /Script/Engine.TimelineComponent	132,195	1.1s	1.1s
TickComponent: Class /Script/AIModule.PathFollowingComponent	459,002	683.5 ms	683.5 ms
TickComponent: BlueprintGeneratedClass /ShooterCore/Elimination/B_MusicManagerCor	24,158	664.6 ms	437.2 ms
TickComponent: Class /Script/LyraGame.LyraAbilitySystemComponent	201,320	386.5 ms	386.5 ms
TickComponent: BlueprintGeneratedClass /ShooterCore/Game/B_TeamSpawningRules_B_	24,157	54.8 ms	54.8 ms
TickComponent: Class /Script/Engine.PhysicsFieldComponent	24,158	44.9 ms	44.9 ms
TickComponent: Class /Script/Engine.LineBatchComponent	72,474	22.5 ms	22.5 ms
TickComponent: BlueprintGeneratedClass /Game/Audio/Blueprints/B_MusicManagerCom	0	0	0
TickComponent: Class /Script/AIModule.PawnActionsComponent	0	0	0
TickComponent: Class /Script/Engine.SpectatorPawnMovement	0	0	0

Плагін “Extensive Pool System”



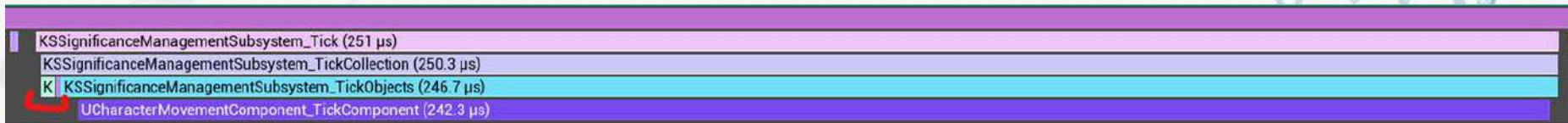
Замість повторного створення персонажів, вони кладуться до пулу

Плагін “Powerful Tick Management”



Плагін “Powerful Tick Management”

Застосовано для руху персонажів у просторі.



Name	Count	Incl	Excl
All (5 / 34,598)	133,150	49.3s	286 ms
KSSignificanceManagementSubsystem_Tick	26,630	16.4s	23.8 ms
KSSignificanceManagementSubsystem_TickCollection	26,630	16.4s	4.6 ms
KSSignificanceManagementSubsystem_TickObjects	26,630	16.3s	144.7 ms
KSSignificanceManagementSubsystem_UpdateSignificance	26,630	85.2 ms	85.2 ms
KSSignificanceManagementSubsystem_SortBySignificance	26,630	27.7 ms	27.7 ms

Результати оптимізації

	GT, мс	N ₁₀ , шт	N ₁₅ , шт	N ₃₀ , шт	M _{min} , МБ	M _{avg} , МБ	M _{max} , МБ	FPS, Гц
Базове значення	8,59	1703	26,43	0,76	2065	2147	2192	105,7
Оптимізоване значення	7,72	671	20,44	0,82	2064	2175	2240	117,4
Δ, %	-10,1	-60,6	-22,7	+7,9	-0,05	+1,3	+2,2	+11,1

Висновки

- Розроблено плаґін “Extensive Pool System”
- Розроблено плаґін “Powerful Tick Management”
- Плаґіни інтегровано у проєкт Lyr a
- Оптимізовано 10.1% часу виконання основного потоку
- Знижено кількість довгих кадрів на 22.7%
- Практична цінність плаґінів
- Можливість розвитку подальших досліджень

Дякую за увагу!
