

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Механізми захисту у багатокористувацьких онлайн ігрових додатків
(тема)

Виконав: Лаухін Ю.Р.
(прізвище, ініціали)

студент 2 курсу, групи БІКСм-18-1

Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник доц. Балагура Д.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Халімов Г.З.
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Лаухіну Юрію Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Механізми захисту у багатокористувацьких онлайн ігрових додатків
затверджена наказом по університету від "04" листопада 2019 р. № 1649Ст
2. Термін подання студентом роботи до екзаменаційної комісії _____
3. Вихідні дані до роботи Теоретичні данні про розробку ігрових додатків
4. Перелік питань, що потрібно опрацювати в роботі
1. Особливості розробки онлайн ігрових додатків
2. Найпоширеніші атаки на онлайн ігрові додатки
3. Дослідження механізмів захисту від атак на ігрові додатки
4. Модель порушника
5. Опис програмної реалізації онлайн ігрового додатка та систем захисту від атак
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Презентаційний матеріал у вигляді слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Отримання завдання</i>	<i>25.01.19</i>	
2	<i>Пошук літератури</i>	<i>26.01.19- 15.05.19</i>	
3	<i>Аналіз зібраних даних</i>	<i>16.05.19- 19.09.19</i>	
4	<i>Розробка онлайн ігрового додатку та систем захисту від найпоширеніших атак</i>	<i>20.09.19- 10.10.19</i>	
5	<i>Аналіз отриманих результатів</i>	<i>11.10.19- 18.10.19</i>	
6	<i>Оформлення пояснювальної записки</i>	<i>19.10.19- 16.11.19</i>	

Дата видачі завдання _____ 20__ р.

Студент _____
(підпис)

Керівник роботи (проекту) _____ доц. Балагура Д.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Атестаційна робота містить стр. 88, рис. 10, дод. 1, дж. 16.

ІГРОВЕ ЗАСТОСУВАННЯ, ЗАХИСТ ІГРОВИХ ЗАСТОСУВАНЬ,
ШАХРАЙСТВО, ОНЛАЙН, БАГАТОКОРИСТУВАЦЬКІ ЗАСТОСУВАННЯ

Об'єкт дослідження – багатокористувацькі онлайн ігрові застосування.

Предмет дослідження – атаки на багатокористувацькі онлайн ігрові застосування.

Основним завданням роботи є дослідження атак на багатокористувацькі онлайн ігрові застосування , реалізація AimBot та Bot з записом дій користувача.

РЕФЕРАТ

Аттестационная работа содержит стр.88, рис. 10, доп. 1, ист. 16.

ИГРОВОЙ ПРИЛОЖЕНИЕ, ЗАЩИТА ИГРОВЫХ ПРИЛОЖЕНИЙ,
МОШЕННИЧЕСТВО, ОНЛАЙН, МНОГОПОЛЬЗОВАТЕЛЬСКИЕ
ПРИЛОЖЕНИЯ

Объект исследования – многопользовательские онлайн игровые приложения.

Предмет исследования – атаки на многопользовательские онлайн игровые приложения.

Основной задачей работы является исследование атак на многопользовательские онлайн игровые приложения, реализация атаки с изменением исходного кода игрового приложения.

ANNOTATION

The attestation work contains pages 88, figures 10, appendixes 1, bibliographic titles 16.

GAME APP, GAME APPLICATION PROTECTION, FRAUD, ONLINE, MULTI-USER APPLICATIONS

Object of study - multiplayer online gaming applications.

The subject of the research is attacks on multiplayer online gaming applications.

The main objective of the work is the study of attacks on multiplayer online gaming applications, the implementation of an attack with a change in the source code of the gaming application.

ЗМІСТ

УМОВНІ ПОЗНАЧЕННЯ, СИМВОЛИ, ОДИНИЦІ СКОРОЧЕННЯ І ТЕРМІНИ	11
ВСТУП	12
1 ТЕОРЕТИЧНІ АСПЕКТИ СТВОРЕННЯ ТА ВИПУСКУ ВІДЕОІГОР....	13
1.1 Відеоігри. Класифікація відеоігор.....	13
1.2 Етапи створення відеоігор.....	16
1.2.1 Написання коду.....	17
1.2.2 Розробка контенту	18
1.2.3 Розробка ігрової механіки	19
1.2.4 Тестування.....	20
1.3 Фінансові сторони створення відеоігор.....	20
1.4 Ліцензування в сфері створення відеоігор	21
1.4.1 Ліцензія на використання сторонньої торгової марки	21
1.4.2 Ліцензування музики.....	22
1.4.3 Ліцензія ігрового рушія.....	24
1.4.4 DRM ліцензія	25
1.5 Проблеми порушення авторських прав в ігровій індустрії	25
1.5.1 Захист від несанкціонованого доступу	26
1.5.2 Захист від модифікацій	28

2 БАГАТОКОРИСТУВАЦЬКІ ОНЛАЙН ВІДЕОІГОРИ ТА ШАХРАЙСТВО В НИХ	29
2.1 Багатокористувацький додаток.	29
2.2 Класифікація багатокористувацьких додатків.....	30
2.2.1 По технічній реалізації.....	30
2.2.2 За правилами	31
2.2.3 Класифікація за типом клієнту.....	31
2.2.4 Типи багатокористувацької гри в 3D-шутерах.....	33
2.2.5 Типи багатокористувацької гри в стратегіях в реальному часі	34
2.2.6 За організацією зв'язку.....	35
2.3 Шахрайство в онлайн іграх	38
2.3.1 Причини існування.....	39
2.3.2 Відмінність від індивідуальних ігор.....	40
2.3.3 Технічні причини.....	40
2.3.4 Невизначеність поняття шахрайства	41
2.3.5 Основні типи шахрайства в онлайн-іграх.....	42
2.3.5.1 Загальні для різних жанрів	42
2.3.5.2 В шутерах	45
2.3.5.3 У стратегіях реального часу	49
2.3.5.4 У MMORPG.....	49
2.3.6 Протидія шахрайству	50
2.3.7 Легалізація шахрайства.....	53

3 ДОСЛІДЖЕННЯ НЕЛЕГАЛЬНОГО ДОСТУПУ ДО РЕСУРСІВ У ВІДЕОІГОРАХ	54
3.1 Види нелегального доступу	54
3.2 Класифікація шахрайства у відеоіграх	56
3.3 Технічні засоби захисту авторських прав.....	58
3.4 Сучасні методи захисту від несанкціонованого доступу і мета його застосування	60
4 РЕАЛІЗАЦІЯ ІГРОВОГО ЗАСТОСУВАННЯ.....	62
4.1 Загальні відомості	62
4.2 Функціональне призначення.....	62
4.3 Опис логічної структури	62
4.4 Використовувані технічні засоби	63
4.5 Виклик і завантаження	63
4.6 Вхідні дані.....	63
4.7 Вихідні дані.....	63
5 РЕАЛІЗАЦІЯ АТАК ТА МЕХАНІЗМІВ ЗАХИСТУ ДЛЯ БАГАТОКОРИСТУВАЦЬКОГО ОНЛАЙН ІГРОВОГО ЗАСТОСУВАННЯ.....	64
5.1 Теоретичні основи.....	64
5.2 Підготовка програмного та апаратного обладнання	64
5.3 Блокування можливості для створення декількох вікон додатку водночас	65
5.4 Створення AimBot.....	66
5.5 Створення бота, з записом дій користувача	73
5.6 Висновки проведених атак.....	75

	10
ВИСНОВКИ.....	77
ПЕРЕЛІК ПОСИЛАНЬ.....	78
ДОДАТОК А.....	80

УМОВНІ ПОЗНАЧЕННЯ, СИМВОЛИ, ОДИНИЦІ СКОРОЧЕННЯ І ТЕРМІНИ

DRM	– Digital rights management або Керування цифровими правами
CPU	– Центральний процесор
GHZ	– Гігагерц
SSD	– Твердотілий накопичувач
P2P	– Peer-to-peer
LAN	– Local area network
DGS	– Dedicated Game Server

Ліцензування – видача та переоформлення ліцензій, видача дублікатів ліцензій, ведення ліцензійних справ та ліцензійних реєстрів, контроль за додержанням ліцензіатами ліцензійних умов, видача розпорядження про усунення порушень ліцензійних умов, а також розпорядження про усунення порушень законодавства у сфері ліцензування.

Авторські права – це сукупність правових норм, які регулюють відносини, що виникають внаслідок створення і використання творів літератури, мистецтва й науки.

ВСТУП

Комп'ютерні ігри вже давно стали частиною нашого відпочинку та заповнили світ. Деякі проводять так свій вільний час та відпочиває від сірих буднів на роботі. Інші настільки захоплюються цим віртуальним світом, що нерідко перетворюють свій відпочинок у великий заробіток.

У сучасному світі в відеоіграх існує дві проблеми. Шахраї та «пірати».

Перші починають шахраювати в відеоіграх, тому що гра для них занадто важка, або тому що бажають отримати більше привілеїв та стати кращим не дотримуючись правил гри.

Другі порушують авторські права та починають розповсюджувати ігрові додатки без згоди розробників. Найчастіше безкоштовно або за дуже малі кошти, які йдуть у карман зловмисника, а не розробнику.

У роботі будуть розглянуті деякі способи захисту від шахрайства та механізми захисту авторських прав.

1 ТЕОРЕТИЧНІ АСПЕКТИ СТВОРЕННЯ ТА ВИПУСКУ ВІДЕОІГОР

1.1 Відеоігри. Класифікація відеоігор

Сьогоднішня комп'ютерна гра - це багатофункціональна програма, яка використовується в розважальних і освітніх цілях. Відеоігри - це ігри, в яких гравець взаємодіє з віртуальним середовищем, створеної комп'ютером. Стан віртуального середовища надається гравцеві за допомогою різних способів передачі інформації, таких як музичний, візуальний, тактильний. Всі комп'ютерні ігри є мультимедійними. В цей тип ігор можна грати як на локальному комп'ютері, так і в приставці, а також з іншими гравцями через локальну або глобальну мережу.

У 2011 році уряд США і Національний фонд мистецтв США офіційно визнали відеоігри мистецтвом. [1].

Комп'ютерні ігри класифікуються за жанрами та кількістю гравців.

В основі сучасного поділу відеоігор на жанри лежить той вид діяльності, в якому гравець бере участь в іграх цього жанру. Відеоігри діляться на ігри на рух, планування, сюжет і спілкування. Проте, жанрова класифікація найбільш часто використовується.

Action. В іграх цього жанру для подолання ігрових обставин необхідно використовувати рефлексі і швидкість реакції. Це один з основних жанрів і в той же час найпоширеніший. Також в іграх цього жанру потрібно багато пересуватися, іноді битися, стріляти або навіть тікати від переслідування. Для екшн-ігор, в яких присутні важливі елементи пригодницьких ігор, застосовується термін «пригодницький бій». Крім того, вони часто виділяються в напрямку аркадних ігор, ігровий процес характеризується простотою і легкістю розробки.

Цей жанр поділяється на велику кількість піджанрів, основними є:

Стрілянина. Основою ігрового процесу є стрілянина з гвинтівок по цілям, зазвичай ворожим. Залежно від перспективи вони поділяються на шутери від першої особи (серія Battlefield) або сторонні (Max Payne). Існують варіації як тактичні, в яких ігровий персонаж виступає в складі команди (Rainbow Six Siege), аркади (Alien Shooter), стелс-екшену, метою якого є приховані дії для виконання завдань без безпосереднього знищення противників (серія Hitman)). Shoot 'em up відноситься до ігор, в яких основою ігрового процесу є знищення великої кількості ворогів, а сама стрілянина превалює над тактикою і точністю.

Файтинги — імітують ближній бій на спеціальних аренах. Приклади: серія Mortal Kombat, Street Fighter, Injustice.

Beat 'em up — подібні на файтинги, з тою різницею, що персонажі вільно переміщуються за ігровим світом (а не спеціальною ареною) і проводять бої проти багатьох противників одночасно. Приклади: Golden Axe, Battletoads, Double Dragon.

Платформери — персонаж мусить рухатися, стрибаючи по платформах долати перешкоди. Приклади: Mario, Spyro the Dragon, Megaman.

Лабіринти — персонаж рухається лабіринтом з метою знайти вихід, зібрати предмети і/або уникнути пасток і небезпек. Часто в іграх цього жанру є обмеження на час. Приклади: Pac-Man, Boulder Dash.

Стратегія. Сенс стратегічних ігор полягає в тому, щоб планувати дії і формулювати конкретну стратегію для досягнення конкретної мети, наприклад, виграти військову операцію. Гравець управляє не одним персонажем, а цілим загоном, бізнесом або навіть всесвітом. За реалізації ігрового часу стратегічні відеоігри діляться на дві основні різновиди:

Покрокові стратегічні ігри — гравець та його противник здійснюють дії один за одним, покроково, маючи змогу за один ігровий хід виконати певну кількість операцій. Приклади: Heroes of Might and Magic, Цивілізація, Xcom.

Стратегічні ігри в реальному часу — і гравець і противник виконують свої дії одночасно, проте часто масштаб часу відрізняється від реального. Наприклад, будівництво триває кілька секунд, а ігрова година складає кілька хвилин реального часу. Приклади: StarCraft, Warcraft, Age of Empires.

Tower Defense — похідний жанр, в якому гравець керує оборонними баштами аби не пропустити хвилі ворогів.

MOBA — похідний жанр, в якому гравець керує одним персонажем з метою захистити свою базу від ворогів та знищити ворожу.

MMORTS — багатокористувацькі он-лайн стратегії в реальному часі, орієнтовані на суперництво/співпрацю з іншими реальними гравцями, щоб досягнути спільної мети.

Рольова гра. Гравець пов'язаний з конкретним персонажем або лідером команди, який діє відповідно до правил своїх ролей. Наприклад, кожен гравець бере на себе управління одним персонажем і грає свою роль, будь то поліцейський або таксист. Метою ігрового процесу є виконання різних завдань (квестів) для розвитку одного персонажа або групи. Можливі дії залежать від обраного зображення персонажа, заздалегідь визначеного або сформованого самим гравцем.

Іноді рольові відеоігри поділяються за дизайном і побудовою сюжету. Так існує умовний поділ на рольові ігри західного зразка та східного.

Симулятор. У широкому сенсі всі ігри є симуляторами. Якщо коротко, це відеоігри, призначені для представлення реальності за допомогою демонстрації певних явищ і властивостей реального світу в віртуальному середовищі. Існує безліч піджанрів, таких як технічні (управління складними технічними пристроями, авіаційні технології та інші, наприклад, гра Project Car), аркада (відрізняється від аркади наявністю спрощеної фізичної моделі. Наприклад, Need For Швидкість), спорт, економ, знайомства та інші.

Пригоди. У пригодницьких відеоіграх гравець управляє ігровим персонажем, який переміщається по сюжету і виконує визначені завдання сценарію, спираючись на свою увагу і логіку, шукає підказки і вирішує головоломки. В рамках жанру виділяються основні піджанри: інтерактивна література, інтерактивні фільми та візуальні романи. Часто, за аналогією, пригодницькі фільми - це ті відеоігри, сюжетна лінія яких динамічно розгортається, насичується яскравими подіями, що швидко змінюються оточеннями, а персонажі демонструють винахідливість і мужність. Приклади: Uncharted або Tomb Raider.

Інші різновиди. Настільна гра — електронні реалізації настільних ігор. Серед них існують як реалізації класичних ігор (шахи, преферанс), так і специфічних як Magic: The Gathering.

Головоломки — вимагають від гравця вирішення логічних завдань, передбачення можливих ситуацій. Приклади: Tetris, Angry Birds.

Games with a Purpose — програми, за допомогою яких люди використовують свої знання для вирішення проблем (передусім у наукових дослідженнях), при цьому граючись. Такі ігри є різновидом людино-орієнтованого методу комп'ютерного моделювання [2].

1.2 Етапи створення відеоігор

Розробка відеоігор має ряд послідовних етапів, в цілому їх три: розробка програмного забезпечення, розробка контенту (креслення, моделі, музика) і розробка ігрової механіки. Їм передує проектування (pre-production) - генерація ідей для майбутнього ігрового процесу для ігрового дизайнера, вибір жанру, теми, особливостей ігрового процесу, розробка сценарію і зображень персонажів за допомогою оточення. Менеджер координує діяльність різних людей, що беруть

участь в розробці, складає план їх роботи, встановлює терміни її реалізації, планує витрати. Закінчена гра, в свою чергу, повинна пройти серію етапів, під час яких вона досягає гравців і підтримує інтерес до себе. Індустрія відеоігор включає в себе безліч людей з різними професіями та ролями: програмісти ігор, художники, моделісти і аніматори, які створюють графічний контент, композитори і звукорежисери, які створюють звуковий дизайн і музику, що часто здається окремою проблемою. Виробники несуть відповідальність за успішне завершення проекту.

1.2.1 Написання коду

Оскільки відеогра є комп'ютерною програмою, її робота, технічні характеристики, контент і ігровий процес забезпечуються програмним кодом. Розробка ігор включає ті ж етапи, що і розробка програмного забезпечення, але включає в себе більше роботи з контентом і ігрової механіки.

Сучасні ігри в основному засновані на готових програмних модулях - ігрових движках, в яких вже реалізовані базові функції, які можуть поєднувати графіку, звук, об'єкти і їх руху. Щоб налаштувати движок для реалізації конкретного плану, програмісти уточнюють його, додаючи необхідні функції. Існують як безкоштовні ігрові движки, доступні кожному, так і ті, яким для їх використання потрібна ліцензія. Крім того, драйвери розрізняються за ліцензіями. Для сторонніх розробників їх використання може бути набагато дешевше.

Деякі двигуни призначені для створення ігор певного жанру, інші універсальні. Не всі драйвери можуть забезпечувати однакові ігрові можливості і рівень графіки. Деякі з драйверів дозволяють створювати ігри для різних платформ, тому Unreal Development Kit підтримує розробку інтерактивних робіт для ПК, Xbox 360, PlayStation 3, Wii і Android.

Деякі ігри створюються в спеціальних програмах, які вже мають початкові ресурси, дії і не вимагають знання мов програмування. Прикладами таких програм є Game Maker, Construct, RPG Maker.

1.2.2 Розробка контенту

Відеогра передбачає створення графіки, звуків і внутрішньоігрових текстів. Концепт-арт виконується на папері або на комп'ютері, як правило, в декількох варіантах. Грунтуючись на концептуальному мистецтві, художники стверджують і створюють дво- або тривимірні моделі персонажів, об'єкти і декорації. Для цього художники працюють в програмах, призначених для роботи з графікою.

Для підтримки руху моделей вони анімовані в інших спеціальних програмах. Створюються різні набори ходів, які будуть відтворюватися в залежності від конкретних дій гравця, що використовує код. У разі двовимірної графіки це набори спрайтів, де кожне зображення являє собою окремий кадр. Захоплення руху живих акторів може бути використаний для реалізації реалістичних рухів або емоцій. Як тільки датчики виявляють рух, вони передаються комп'ютерного персонажу як людина або монстр.

Візуальні ефекти роблять гру більш цікавою і задають стиль. Деякі з них додають реальності, такі як відкидання тіней, заломлення світла, постріли і вибухи. Інші вказують стану персонажа і дії, які визначають стиль гри. Деякі ігри повністю в стилі коміксів чи фільмів. За реалізацію картинки і звуку відповідають графічні і звукові движки.

Для звукового дизайну написана музика, а персонажі озвучені. Крім того, для повноцінного звукового оформлення потрібні такі ефекти, як кроки, звук пострілів. Вони можуть бути обрані з безкоштовних бібліотек або записані індивідуально. Деякі композитори спеціалізуються на створенні музики для ігор. Музика може бути виконана цілими професійними оркестрами, мати музичний

супровід. Діалоги персонажів часто озвучуються спеціальними акторами в звукових студіях.

Іноді ігри включають відео вставки, створені в 2D або 3D анімаційних програмах. Іноді відео актори знімають живих акторів і створюють декорації. Є актори, які спеціалізуються на зйомках таких відео-вставок або озвучці. Сюжет, діалоги, додаткові тексти написані сценаристами і відповідальними авторами.

1.2.3 Розробка ігрової механіки

Ігрова механіка визначає насиченість ігрового процесу, правила, за якими грається відеогра. Основою механіки є ігрові об'єкти, такі як персонажі, об'єкти, якими вони можуть маніпулювати, декорації. Частиною ігрової механіки є управління тим, як гравець управляє персонажем і ігровим світом. Наприклад, як встановити напрямок руху, як взаємодіяти з віртуальними об'єктами. Крім того, на етапі розвитку механіки створюється призначений для користувача інтерфейс, який інформує гравця і дозволяє взаємодіяти з ігровим світом.

Зазвичай ігри поділяються на рівні (локації), щоб комп'ютер не завантажувався обчисленнями по всьому світу гри. В останніх іграх світ часто моделюється так, що немає чіткого поділу на локації. Дизайнер рівнів поміщає готові об'єкти в ігровий світ і продумує їх руху. Розташування рівнів визначає, наскільки цікавою буде гра, які можливості гравець буде мати в конкретній ситуації.

Фізичний двигун відповідає за взаємодію об'єктів, яке відбувається без контролю гравця. Наприклад, він реалізує закони інерції, гравітації, поведінки рідин, властивості об'єктів. Штучний інтелект (ШІ) відповідає за поведінку персонажів, коли вони реагують на дії гравця. Багато ігрові події написані за сценарієм. Самі події придумані сценаристами, а сценарії - програмістами [3].

1.2.4 Тестування

Після роботи над кодом, контентом і механікою, по якій може функціонувати гра, її завершення завершено. Гра, яка не була повністю скомпільована, але в неї можна грати, називається альфа-версією. Він може містити значні помилки і недоліки, такі як відсутність функцій, музики або об'єктів. Тестери, які грають в цю гру, займаються виявленням проблем, намагаючись в повній мірі використовувати всі доступні в ній функції. На більш пізньому етапі випускається бета-версія, і потенційні покупці гри можуть її тестувати. У бета-версії є подальший пошук помилок, перевірка правильності взаємодії об'єктів в ігровому світі, управління. Можливі зміни макета, зміни ігрового балансу тощо [4].

1.3 Фінансові сторони створення відеоігор

З 2011 року деякі компанії по розробки ігор можуть отримати гранти від США на створення відеоігор, але за умовою що гроші підуть на створення об'єкта, який згодом стане національним надбанням. Розміри грантів варіюються від 10 до 200 тисяч доларів.

Окрім цього ентузіасти, які не хочуть пов'язувати свою діяльність по створенню відеоігор з великими компаніями можуть розраховувати на підтримку звичайних гравців або просто зацікавлених людей.

Рух почався зі стартом платформи KickStarter, де спільними зусиллями збирають кошти на реалізацію різних проектів. Розробники зрозуміли можливості цієї платформи, коли Тім Шейфер розмістив там свою нову гру Double Fine Adventure на початку 2012-го року. Тім розраховував зібрати 400000 \$, але в підсумку отримав майже три з половиною мільйони. За ним послідували inXile entertainment з Wasteland 2 і Stainless Games з Carmageddon: Reincarnation.

Початківцям розробникам на такий прийом розраховувати не варто, але невелику суму грошей на KickStarter зібрати цілком реально.

За офіційною статистикою KickStarter більш ніж на \$ 100 млн було зібрано на проекти ігор, що дало зелене світло 1,476 проектам. За 2012-2016 роки більш ніж 633,242 розробників пообіцяли більш ніж 4500 ігрових проектів, майже 1000 з них тільки в 2016 році .

При створенні комп'ютерної гри треба дуже ретельно планувати бюджет. Крім банальних витрат, наприклад, зарплатня робітникам (геймдизайнерам, програмістам, художникам, тестерам та іншим), оренда приміщень, де буде проходити робочий процес, середовище розробки, маркетинг, треба звернути увагу на такі області як, ліцензування програмного продукту (більш детально це буде описано у наступній главі), брендування, створення захищеного коду, застосування зовнішнього захисту авторських прав.

1.4 Ліцензування в сфері створення відеоігор

1.4.1 Ліцензія на використання сторонньої торгової марки

Власник свідоцтва на зареєстрований знак для товарів і послуг має право дати дозвіл (видати ліцензію) на використання зареєстрованого знака на підставі ліцензійного договору (п. 8 ст. 16 Закону України «Про охорону прав на знаки для товарів і послуг»). Ліцензійний договір на використання торговельної марки - це договір між ліцензіаром (власником) і ліцензіатом, в якому одна сторона (ліцензіар) передає другій стороні (ліцензіату) право на використання в господарській діяльності торговельної марки на чітко встановлених умовах, а саме: користування ліцензіатом правами на певній території, в певні терміни і за певну винагороду. Ліцензійний договір на торговельну марку передбачає такі види ліцензій, закріплені на законодавчому рівні:

Одинична ліцензія – видається лише одному ліцензіату і виключає можливість видачі ліцензіаром іншим особам ліцензій на використання об'єкта права інтелектуальної власності у сфері, що обмежена цією ліцензією, але не виключає можливості використання ліцензіаром цього об'єкта у зазначеній сфері.

Виняткова ліцензія – видається лише одному ліцензіату і виключає можливість використання ліцензіаром об'єкта права інтелектуальної власності у сфері, що обмежена цією ліцензією, та видачі ним іншим особам ліцензій на використання цього об'єкта у зазначеній сфері.

Невиключна ліцензія – не виключає можливості використання ліцензіаром об'єкта права інтелектуальної власності у сфері, що обмежена цією ліцензією, та видачі ним іншим особам ліцензій на використання цього об'єкта у зазначеній сфері.

У ліцензійному договорі також вказується перелік товарів і послуг торгової марки і територія на яку поширюється його дія. В Україні державна реєстрація ліцензійного договору не обов'язкова. Сторони можуть його зареєструвати за власним бажанням.

1.4.2 Ліцензування музики

Сьогодні розробники мають безліч різних варіантів. На багатьох сайтах продаються треки за ліцензією без відрахувань, є навіть безкоштовна музика різноманітних стилів. Крім того, є безліч композиторів, готових писати оригінальну музику за невелику ціну. Але іноді цього недостатньо. Якщо вам потрібні дуже специфічні пісні або саундтреки, який виділить вашу гру на фоні інших, то для вас підійде ліцензування.

Перший крок до ліцензованої музики - пошук відповідних пісень. Швидше за все, популярні пісні, які можна почути по радіо / телебаченню / Інтернет, фінансово важкодоступні. Великі звукозаписні компанії зазвичай запитують

величезні суми з оплатою вперед, що можуть дозволити собі не всі розробники. Краще за все шукати інді-виконавців або дрібні студії, і завдяки Інтернету, сьогодні це зробити як ніколи просто.

Існують багато музичних онлайн-магазинів, наприклад, iTunes або CD Baby. В них продається безліч композицій інді-виконавців в різних стилях, а на веб-сайтах є зручні функції пошуку. Ще один відмінний спосіб – пошук в відео YouTube. Багато виконавців випускають відео і записи концертів. Також можна пошукати в більш спеціалізованих онлайн-магазинах.

Коли виявили пісню, яку необхідно ліцензувати, наступний крок - знайти того, з ким потрібно домовитися про ліцензування. У більшості онлайн-магазинів дуже мало інформації про авторів пісень, але іноді вдається знайти їх імена та/або назви студій, яким належать права. За допомогою цієї інформації можна знайти офіційний сайт цього виконавця / студії. Це може бути складніше, ніж здається. Запити в Гуглі часто видають посилання на новинні, статті, інтерв'ю та відео, і тільки потім офіційні веб-сайти, якщо вони взагалі існують. У назвах маленьких студій часто використовуються звичайні слова. Пошук по таким словам призводить до купи посилань але ніяк не зв'язані з цими студіями сайти. Якщо не вдається знайти офіційну сторінку, то спробуйте пошукати сторінку на Facebook. У розділі «Інформація» (About) зазвичай є посилання на офіційний сайт або контактна інформація.

Деякі інді-виконавці можуть використовувати власні договори, але більшість буде чекати, що їх підготуєте ви. Тому непогано буде перед відправкою листа заздалегідь написати договір.

При написанні договору не варто забувати про таких аспектах:

Ексклюзивне використання пісні буде коштувати набагато дорожче, так що подумайте двічі.

Виконавець повинен зберігати всі права на композицію і при необхідності згадано. Якщо ви хочете мати право на копіювання композиції, готуйтеся заплатити за це.

Треба описати всі допустимі варіанти використання композиції. Чи буде вона поширюватися як частина саундтрека гри? Буде використовуватися для реклами гри або в трейлері? Або в безкоштовному демо? Чи буде вона якимось чином змінюватися?

Найкраще буде найняти для написання договору юриста. Але можна, щоб не писати договір «з нуля», скачати безкоштовний шаблон і заповнити під свої вимоги.

1.4.3 Ліцензія ігрового рушія

Існують багато ігрових рушіїв у сучасному світі. Більшість розробників щоб заощадити свої фінанси та, найважливіше, час, використовують вже створені рушії, замість створення свого.

Розглянемо ліцензії у найпопулярніших ігрових рушіях на теперішній час: Unity та Unreal Engine.

Unity має підписку і безстрокову ліцензія

Є 4 плану підписки:

- 1) Unity Personal -0 \$
- 2) Unity Plus -420 \$ / рік
- 3) Unity Pro -1500 \$ / рік
- 4) Unity Enterprise – ціна залежить від кількості персональних комп'ютерів у компанії.

Безстрокова ліцензія: Unity Pro – 3000 \$.

Вибір підписки залежить від вашого доходу від продукту створеного на Unity, якщо ваш дохід не перевищує 100000 \$, то можна використовувати

безкоштовну версію. 100000 \$ до 200000\$ – Unity Plus. Більше 200000 \$ – Unity Pro.[5]

Unreal Engine більш простий у плані вибору ліцензії.

Якщо проект не комерційний – то можна використовувати безкоштовну версію. В іншому випадку, 5% від вашого доходу треба відчислити розробникам Unreal Engine [6].

1.4.4 DRM ліцензія

DRM - це скорочення від Digital Rights Management, що означає управління цифровими правами. Це управління застосовується при виробництві цифрових продуктів, захищених різними технічними засобами, які запобігають незаконне розпорядження інтелектуальною власністю. Такий «цифровий замок» у вигляді ліцензійного ключа шифрування може бути дуже ефективним. Власник не зможе скопіювати придбаний цифровий продукт, передати його, продати і т. Д.

Ліцензія DRM - це цифровий ключ, який використовується для дешифрування раніше зашифрованого контенту, який також дає вам право використовувати цей контент. Такий ключ можна отримати, тільки якщо він оплачений.

Існують також часткові ліцензії DRM в формі різних «пробних періодів», обмежених кількістю запусків програми і іншими видами тимчасового доступу до функціональності продукту, після чого доступ до програми блокується, і нам пропонується купити повну версію. програми [7].

1.5 Проблеми порушення авторських прав в ігровій індустрії

Скидання DRM-ліцензії залежить від конкретики кожної ліцензії. Різний хакерський софт дозволяє обнулити лічильники запуску файлів із захистом DRM,

відв'язувати такі файли від певних сайтів, нейтралізувати перевірку ключа використовуваним програмним забезпеченням і інші.

1.5.1 Захист від несанкціонованого доступу

У більшості випадків використання DRM в комп'ютерних іграх обмежує можливість установки гри на певну кількість комп'ютерів, що вимагає реєстрації на онлайн-сервері. У більшості ігор з такими обмеженнями допускається установка трьох або п'яти комп'ютерів, причому деякі з них також враховують видалення гри, тим самим відновлюючи можливість іншої установки. Він не тільки обмежує користувачів, у яких вдома більше комп'ютерів, але також може бути проблемою при оновленні ОС або, наприклад, форматуванні жорсткого диска, що тягне за собою переустановку гри, що можна вважати новою установкою, і можливе відключення гри навіть при використанні на одному комп'ютері.

Для деяких програмних продуктів (включаючи ігри) часто використовується прив'язка до носія, наприклад, компакт-диск. Тобто для запуску гри потрібна наявність оригінального приводу CD-ROM, який захищений від копіювання стандартними засобами.

Процедура реєстрації програмного забезпечення полягає в тому, щоб ввести певну інформацію про споживача програмного забезпечення (заповнення реєстраційної карти) і відправити її виробнику по електронній пошті, через Інтернет. Після цього споживачеві надсилається реєстраційний код, після чого він стає зареєстрованим користувачем і отримує передбачувані привілеї (технічна підтримка, гарантійне обслуговування і т. Д.). У свою чергу, виробник програмного забезпечення оновлює статистичну інформацію про своїх клієнтів. Оскільки ім'я користувача не унікально, бажано пов'язати кожен екземпляр продукту для продажу з якимось неповторним значенням, серійним номером. Цей

номер вказується користувачем при заповненні реєстраційної картки та в подальшому використовується при спілкуванні з виробником. І на додаток до програмного забезпечення серійний номер цілком може виконувати допоміжну функцію - обмежувати нелегальне копіювання. Якщо додаток вимагає установки правильного серійного номера шляхом крадіжки (копіювання) носія поширення програми, який є однаковим для всіх користувачів, отримати робочу копію програми буде неможливо. А розподіл серійних номерів дозволяє знайти і покарати пов'язаного з цим номером користувача.

У деяких випадках після установки програми (незалежно від того, вводите ви серійний номер або без нього), щоб отримати доступ до всіх функцій програми, користувач повинен виконати іншу процедуру - реєстрацію або, як тепер називає це Microsoft, активацію. Така поведінка типово для більшості продуктів Shareware, а також для додатків, які, на думку розробників, не можуть запускатися до тих пір, поки користувач не надасть всю необхідну інформацію, навіть якщо він вже придбав ліцензію.

Коли ви вводите отриманий реєстраційний код, спрацьовує механізм захисту програмного забезпечення, який перевіряє правильність введеного коду. В цьому випадку вміння обчислювати правильні коди завжди має залишатися тільки в руках розробника. Щоб зловмисник, виправивши кілька байтів, не дозволив програмного забезпечення працювати так, як якщо б воно було належним чином зареєстровано або активовано, частина коду або даних, до яких можуть звертатися тільки законні користувачі, можуть бути зашифровані за допомогою стабільного алгоритму, і ключ шифрування розраховується з реєстраційним кодом. Тоді без відома реєстраційного коду отримати повну версію програми не вдасться. ASProtect (ASPack Software) і EXECryptor (SoftComplete Development) надають схожу функціональність.[8]

1.5.2 Захист від модифікацій

Захист від модифікації ігрових додатків застосовується у декількох напрямках:

Захист ігрових даних – дані, які фіксують ігровий процес користувача. Найкращий захист для таких даних, це зберігання у недоступному для користувача місті (щоб вони не буди випадково видалені) та шифрування. Для шифрування рекомендується використовувати сучасні алгоритми, наприклад, AES.

Захист пам'яті програми – дані, які знаходяться в RAM у момент функціонування додатка. Такий захист може впровадитись декількома способами. Один з них це правильне програмування скриптів. Розумне використання публічних змінних, тому що теоретично користувач може отримати доступ до любых публічних даних. Щоб захистити публічні змінні, можна використати дуже швидке и просто шифрування, яке буде зашифровувати дані при кожному їх записі, та розшифрування при кожному читанні.

Захист внутрішньоігрових покупок

Цей захист потрібен, щоб внутрішньоігрові покупки не можна було підробити. Наприклад, на Android існує програма Freedom, яка може підмінити сервіс внутрішньоігрових покупок и усі покупки у додатку будуть проходити без плати успішно. Щоб цього уникнути треба з сервісом покупок використати цифровий підпис. На сервері формується 2 ключа: закритий та відкритий. Закритим сервер підписує кожну відповідь про покупки. А в додатку за допомогою відкритого ключа перевіряється справжність підпису. Також, щоб уникнути атаки man-in-the-middle треба до запиту на сервер додавати деяку випадкову строку, яка додається до повідомлення.

2 БАГАТОКОРИСТУВАЦЬКІ ОНЛАЙН ВІДЕОІГОРИ ТА ШАХРАЙСТВО В НИХ

2.1 Багатокористувацький додаток.

Багатокористувацький додаток – режим комп'ютерної гри, під час якого грає більше однієї людини.

Першою багатокористувацькою грою вважається Tennis for Two (1958 рік). Гру було розроблено для аналогової ЕОМ і засобом візуалізації гри слугував осцилограф.

Перші мережеві ігри для декількох гравців вийшли 1973 році. Приблизно в один час створюються гри Maze War, Empire і Spasim. Зважаючи на те, що точна дата створення ігор не відома, вони до сих пір претендують на звання першої розрахованої на багато користувачів гри. Maze War - тривимірний шутер від першої особи з мережевою грою в режимі Deathmatch. Empire - розрахований на багато користувачів мережевий 2D-шутер (до 50 гравців). Spasim - розрахований на багато користувачів мережевий тривимірний космічний симулятор (до 32 гравців).

Вже перші приставки (Pong, Magnavox Odyssey) мали два контролера, що дозволяло грати удвох.

Онлайн-гра – це відеогра, яка використовує постійне підключення до Інтернету для забезпечення ігрового процесу. Концепції онлайн-ігор і онлайн-ігор схожі, але не ідентичні. Наприклад, World of Warcraft - це онлайн-гра (ви не можете грати в неї без підключення до інтернету), а Warcraft 3 - мережева гра (інтернет необхідний для багатокористувацьких ігор, але не потрібно для одиночних ігор) [9].

2.2 Класифікація багатокористувацьких додатків

2.2.1 По технічній реалізації

1) Розрахована на багато користувачів гра на одному комп'ютері:

a) Гравці беруть участь в грі одночасно. У спортивних симуляторах це природний режим. В інших іграх, наприклад, гонках, екран ділиться навпіл, і такий режим називається splitscreen.

b) Гравці беруть участь в грі по черзі (англ. Hotseat). Використовується в деяких покрокових стратегіях, наприклад, Heroes of Might and Magic. У симуляторах Формули-1 Grand Prix 2, 3 і 4 також є режим почергової гри, коли один гравець веде свій болід, а болідами інших керує комп'ютер, при цьому кілька разів за гонку управління передається від одного гравця іншому.

c) Один комп'ютер - кілька терміналів. На даний момент застосовується в основному в ігрових автоматах.

2) Гра по мережі. Кілька комп'ютерів з'єднані в обчислювальну мережу.

a) Через послідовні або паралельні порти.

b) Через модем.

c) Через локальну мережу і інтернет, по протоколам IPX або TCP / IP. (3D-шутери, стратегії в реальному часі, ...).

d) Онлайн ігри

– Через власний клієнт: Ultima Online, Аллоди Онлайн.

– Браузерні ігри: Tale, Бійцівський клуб, Magic, Livegames.

– Через електронну пошту. Див. РВЕМ.

– Через спеціальний веб-сайт: Мафія.

– Через IRC (наприклад, вікторини).

e) Через Bluetooth, IrDA, SMS (гри для мобільних телефонів).

Деякі ігри (в основному для ігрових приставок) підтримують мережеву гру, в якій на кожній машині можуть грати кілька гравців в режимі splitscreen.

2.2.2 За правилами

1) За одним «життя». Цей режим застосовувався в деяких іграх для перших ПК (наприклад, ZX Spectrum). В цьому режимі гравці зовсім не взаємодіють один з одним, кожен веде свою гру; коли гравець втрачає «життя», управління передається напарнику. Див., Наприклад, Digger.

2) Кооперативна гра. Мета гравців загальна: разом перемогти комп'ютерних суперників. Цей тип гри з'явився набагато раніше мережевих ігор.

3) Особисті змагальні ігри: інтереси гравців перетинаються, і кожен гравець намагається перемогти інших.

4) Командні ігри змагань: гравці об'єднуються в команди, що змагаються один з одним.

5) Ігри з повною свободою, коли об'єднання в команди і війна один з одним повністю лежать на совісті гравців. Зрозуміло, зради не забороняються. Див., Наприклад, MMORPG.

2.2.3 Класифікація за типом клієнту

У ролі клієнтського програмного забезпечення може виступати:

- 1) Особлива програма-клієнт (наприклад, Lineage 2, World of Warcraft).
- 2) Термінальна програма (MUD-клієнт)

MUD-клієнт - програмне забезпечення, що використовується для підключення до ігрового сервера MUD. Найпростіший MUD-клієнт - telnet, використовується рідко, зважаючи на відсутність специфічних функцій. Клієнти, розроблені для гри в MUD, зазвичай володіють наступними функціями: підтримка ANSI квітів, псевдонімів, тригерів і скриптів. Деякі клієнти дозволяють малювати

карту світу і швидко переміщатися по ній. Однак саме відсутність деяких функцій в MUD послужило поштовхом до створення першої BBMMORPG гри.

3) Браузер (BBMMORPG), (наприклад, Танки Онлайн, Травіан).

Браузерна багатокористувацька рольова онлайн-гра (BBMMORPG, англ. Browser Based Massive Multiplayer Online Role-Playing Game з англ. - «браузерна гра», «браузерна багатокористувацька онлайн рольова гра») – жанр комп'ютерних ігор, що є перетином масових багатокористувацьких рольової онлайн (MMORPG, з точки зору ігрового процесу) і браузерних ігор (з точки зору технічної реалізації). Від інших MMORPG відрізняється тим, що для гри потрібно тільки веб-браузер. Це має як плюси: гру не потрібно завантажувати і встановлювати, так і мінуси: гра має серйозні технологічні обмеження.

4) Поштовий клієнт (PBEM). Клієнт IRC (наприклад, у вікторині).

PBEM (від англ. Play By Electronic Mail, з англ. - «гри по електронній пошті») – різновид онлайн-ігор через інтернет, де в якості протоколу для обміну даними гри використовується електронна пошта (e-mail, netmail, QWK і т . п.). Являють собою покрокові гри різної спрямованості, в основному - стратегічні або рольові ігри. Як класичний приклад PBEM-гри можна привести шахи за листуванням.

5) Клієнт Jabber.

б) Браузерні ігри для соціальних мереж.

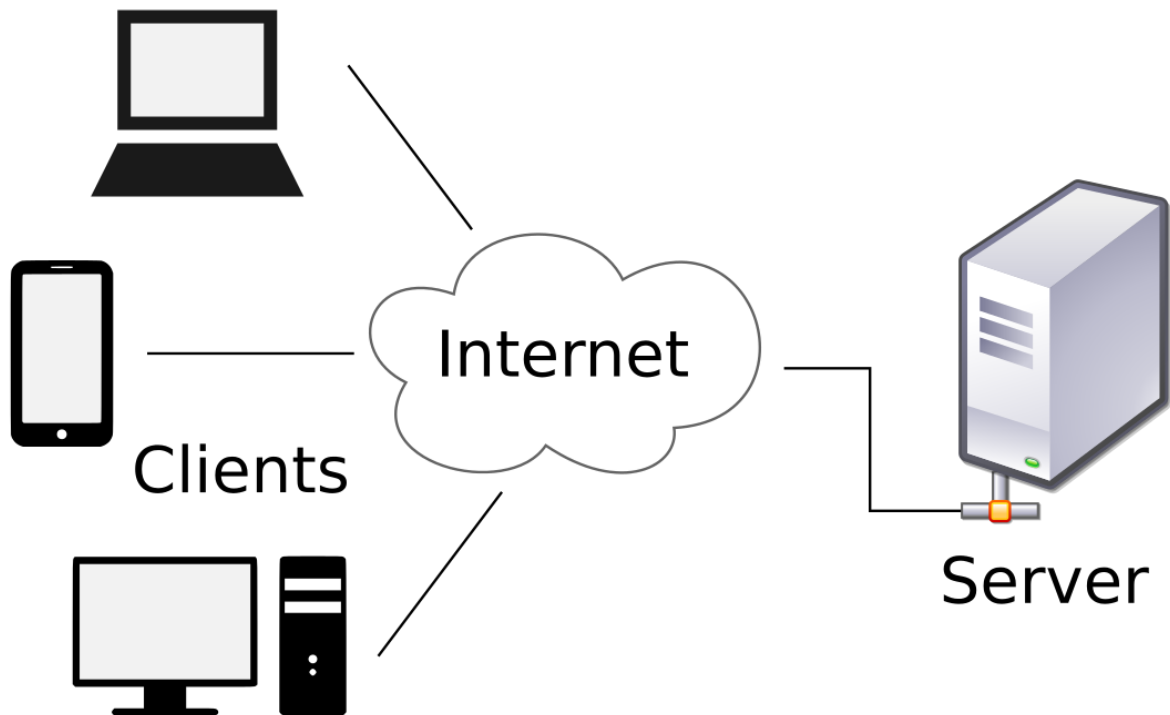


Рисунок 2.1 – Клієнт-серверна модель

2.2.4 Типи багатокористувацької гри в 3D-шутерах

1) Кооперативна гра. Зазвичай мається на увазі гра проти комп'ютерного противника у режимі сюжету. Наприклад, гра з керованими комп'ютером напарниками проти монстрів в режимі Invasion Unreal Tournament 2004 є кооперативною грою, а команди людей проти команди ботів – ні

2) Особисті.

a) Deathmatch - проста перестрілка між гравцями. За вбивство суперника дається очко (фраг). Перемагає той, хто набере більше Фрагів.

b) Last Man Standing - за смерть у гравця віднімається одне життя. Той, у кого скінчилися життя, вибуває.

c) І інші, більш екзотичні типи гри. Див., Наприклад, Aliens versus Predator.

3) Командні.

a) Командний deathmatch.

b) Контроль території (Domination). Одна з перших реалізацій - Unreal Tournament. У цьому режимі на рівні є кілька контрольних точок. Як тільки команда захоплює точку, та починає приносити їй бали. Перемагає той, хто набере більше балів. Подібний режим присутній в грі Team Fortress 2, де потрібно захопити спочатку центральну контрольну точку, а потім ворожі другорядну і головну.

c) Capture the flag. Ця гра схожа на однойменну гру на місцевості. Завдання - захопити ворожий прапор і принести його на свою базу.

d) Загарбник-захисник. Одна сторона нападає, друга обороняється.

e) Втеча. Схожа на попередню. Одна команда - охоронці, друга повинна втекти і в деякому складі (від 1 людини до 50% команди) дістатися до точки евакуації. Приклади - Evacuation в Alien vs. Predator 2, Escape в Counter-Strike, карта AS-Rook в Unreal Tournament.

f) Війна. Варіація режиму контроль території, коли дві команди повинні послідовно захоплювати територію і для перемоги необхідно захопити стартову територію противника. Такий режим гри зазвичай супроводжується використанням важкої техніки, на великих територіях і з великим (для 3D-шутерів) числом гравців (десятки). Прикладом таких режимів є Onslaught в Unreal Tournament 2004, Warfare в Unreal Tournament 3, Heavy Metal в F.E.A.R. 2: Project Origin.

g) І інші (наприклад, Assassination в Counter-Strike, Key Hunt в Nexuiz, Artifact Hunt в S.T.A.L.K.E.R.) [10].

2.2.5 Типи багатокористувацької гри в стратегіях в реальному часі

1) Кооперативна проти комп'ютерних суперників.

2) Особисті.

a) З повною свободою (англ. Free for all). Гравцям дається повна свобода укладати таємні союзи, зрада не забороняється. Варіант: як тільки залишається три людини, гра переходить на правила melee.

b) Melee - гравцям заборонено укладати союзи. Відповідно, юніти стріляють по всім іншим гравцям, а повідомлення, що передаються по чату, передаються всім іншим.

3) Командна гра з рівними умовами.

4) Командний сценарій (наприклад, двоє осаджують замок добре зміцнився третього гравця).

5) Різні модифікації, спрямовані на управління одним персонажем (а не армією).

a) Defense of the Ancients - карта для WarCraft III.

b) Command & Conquer: The Sole Survivor.

6) Екзотичні. Наприклад, карта Steal the Beacon в StarCraft імітує телегру.

2.2.6 За організацією зв'язку

Розглянемо одну з класифікацій – за організацією зв'язку.

Існує 5 видів зв'язку в багатокористувацьких іграх:

1) Local

2) LAN

3) Direct P2P

4) Client-Server Relay (UNet)

5) DGS

Розглянемо їх більш детально.

Local (“coach” multiplayer) – тобто, багатокористувацький додаток, в який можна грати тільки на одному клієнті поруч з іншими людьми. З точки зору багатокористувацького додатку це самий простий тип онлайн відеогри, тому що усе що мається, це ресурси у клієнта під час роботи. Тому розробка таких ігор не дуже відрізняється від звичайної розробки, за винятком того треба обробляти декілька пристроїв введення. Тому перевага даного типу, що в ньому нема онлайн складової. Тож немає і затримок мережі, не треба нікуди відсилати біти. Відсутня необхідність вкладати значні ресурси у безпеку додатку.

Недоліки: якщо усі грають на одній машині, то зазвичай ви не зустрінете гри, яка підтримує більш ніж 4 гравця одночасно, особливо якщо всі гравці користуються одним екраном. Також, завдяки тому, що усі гравці повинні знаходитись поруч під час гри, аудиторія не може зростати високими темпами.

LAN (Local Area Network Games) – це термін, що використовується для опису декількох комп'ютерів, підключених один до одного в межах відносно невеликої площі. Сьогодні цей тип підключення набуває більшої популярності завдяки VR (Virtual Reality). Щоб користувачам було максимально комфортно грати, зазвичай використовують сервер, який не підключено до інтернету, але він має задачу з'єднати багато користувачів разом.

Недолік як і у локального типу гри: гравці повинні бути розташовані на невеликій площі.

Direct P2P (P2P) – може бути 2 різновиду P2P. Перший – прямий P2P. Кожен з гравців намагається напряму з'єднатися з кожним гравцем. Це самий очевидний спосіб реалізації, але й самий вимогливий до обробки, тому що кожен з клієнтів намагається підтримувати багато з'єднань, повинен синхронізувати стан гри и логіка реалізації цього дуже складна.

Другий – Клієнт-сервер хост. Це концепт, в якому один з гравців стає сервером. Тож усі інші гравці підключаються до клієнту-серверу, який централізує

весь трафік. Недолік такого концепту в тому, що саме цей клієнт має обробляти усі дані, що вводить додаткові вимоги щодо ресурсів цього клієнта та при цьому надає, йому так звану Хост перевагу, тому що йому не потрібно отримувати дані через Інтернет, у нього нема затримок, коли у всіх інших є. Також якщо хост виходить з гри, то гра або закінчиться, або треба мігрувати усіх на іншого хоста. Мігрування – саме по собі це не такий важкий процес для розробника, але це дуже не приємний процес для користувачів, тому що гра повинна стати на паузу, треба відновити усі з'єднання, синхронізувати усіх та сподіватися, що все буде добре. Тож мігрування зі збереженням прогресу це не зручно для користувача та дуже важно для розробника.

Client-Server Relay (UNet)– Клієнт-сервер хост з маршрутизатором це різновид P2P. Через топологію мережі все одно треба сподіватись, що користувачі зможуть з'єднатися один з одним, тому що завдяки брэндмауеру це дуже важко. За статистикою лише 80-90 відсотків користувачів можуть з'єднатися один з одним. Тож для 10-20 відсотків користувачів буде поганий досвід гри, і розробник повинен зробити усе, щоб його покращити. Зазвичай це роблять з допомогою безпечного обслуговування маршрутизатору. Тож це добавляє ще більшу затримку, тому що клієнту треба відсилати дані кудись до хмарного сервісу, який потім перешле ці дані до іншого клієнту.

Для більшості різних ігрових додатків P2P з'єднання буде дуже важким для успішної реалізації. Затримка завжди буде різною, також з'єднання взагалі може розриватися.

Безпека – важкий виклик, тому що коли гра хоча б трохи успішна і весь код функціонує стороні користувача, він буде зламаний. І розробник має увесь час намагатися змінити ту частину, яку зламали.

Розширення – ще один важкий виклик. Коли розробник залежить тільки від апаратної частини клієнта, то наскільки б не була потужна апаратна частина, завжди є ліміт у 10-20 людей, які зможуть грати разом.

Ціна – P2P не безкоштовна. Якщо розробник намагається створити гру з невеликими затримками, то потрібно з'єднувати людей які знаходяться умовно близько один до одного. Тому треба реалізовувати сервер для пошуку інших гравців.

DGS – Dedicated Game Server (Виділений ігровий сервер). У цій моделі розробник використовує виділений ігровий сервер, який він контролює. То ж розробник сам вирішує наскільки потужне апаратне забезпечення йому потрібне. Скільки серверів по усьому світу йому потрібно. Тож розробник сам вирішує наскільки швидкий та стабільний досвід від гри користувач отримує.

Також є можливість створити таку логіку на сервері, яка буде потребувати авторизації користувачів, таким чином, коли шахраї будуть шкодити іншим гравцям, у розробника є більше контролю, щоб пошкодити планам злодія.

GS надає найкращу можливу якість з'єднання. Розробник може розширяти гру до підтримки більше ніж сто гравців. Найлегший спосіб ідентифікувати шахраїв. Але це й найбільш дорога модель для багатокористувацького додатка. Тому що розробнику потрібно не тільки сервіси для гри, які повинні функціонувати постійно і не безкоштовні, але й виділений ігровий сервер, який теж не безкоштовний.

2.3 Шахрайство в онлайн іграх

Шахрайство в онлайн-іграх - це дія, яка змінює поведінку гравця по шляху! Використання спеціально створених або модифікованих програм або обладнання в онлайн-іграх, щоб дати одному гравцю перевагу перед іншими. Залежно від гри

різна діяльність розглядається як шахрайство, і ця діяльність розглядається як шахрайство з боку ігровий поліції або ігрового співтовариства. У роботах Гейзінга шахрайство визначається як дія, для якого не вигідно підкорятися правилам гри, в той же час вигідно таємно обіймати їх, щоб використовувати в своїх інтересах суперників [11]. На відміну від шахрайства в особистих іграх, це проблема для ігрових спільнот, і боротьба з нею є важливим аспектом ігрового дизайну. Сильний резонанс в ігровому співтоваристві Проблема шахрайства в значній мірі пов'язана з безпрецедентною поширеністю шахрайства на безкоштовних ігрових серверах Counter-Strike.

Згідно оглядам, шахрайство існує в більшості багатокористувацьких онлайн-ігор, але його важко виміряти. Інтернет і Darknet можуть надати гравцям методику, необхідну для обману онлайн-ігор, іноді в обмін на ігрові цінності.

2.3.1 Причини існування

Шахрайство в онлайн ігрових додатках існує з єдиною метою — легко і швидко отримати те, на завоювання чого «чесним» шляхом піде у багато разів більше часу, зусиль чи грошей (причому ще неясно, чи буде бажаний результат взагалі досягнуто). Незалежно від очікуваного результату, а це може бути рекордний рахунок, час, просто перемога над суперниками — його досягнення приносить певне задоволення, і на нечесний шлях потенційних шахраїв штовхає тяга до самоствердження у купі з невпевненістю в можливості (або небажанням) отримати його «чесним» шляхом. До причин можна додати і бажання прославитися, але це для шахрая є досяжним лише частково: при занадто близькому знайомстві з «чесними» гравцями він неминуче буде викритий.

Найбільш поширеним джерелом коштів для шахрайства є наявність вихідного коду для виконуваних файлів гри, який може бути легально доступним через те, що виконувані коди були відкриті розробниками і незаконно доступні -

через «витоку» виконуваних кодів при вони були вкрадені та розповсюджені через Інтернет.

2.3.2 Відмінність від індивідуальних ігор

В іграх, розрахованих на одного користувача, гравець, використовуючи «нечесні» прийоми, обманює тільки себе (досвідчені гравці говорять, що занадто легка перемога не приносить задоволення), і тому шахрайство в них нешкідливо. Часто розробники використовують чит-коди для тестування гри і надають її гравцям. Зрозуміло, розробники стежать за тим, щоб ці коди не працювали в онлайн-грі.

Онлайн-гра завжди в тій чи іншій формі конкурує між гравцями, і досягнення високих результатів завжди пов'язане з великими зусиллями і часом (так, на розвиток персонажа в середній MMORPG можна витратити від кількох місяців до кількох років). Не дивно, що «звичайні» гравці «легко і швидко» сприймають можливість досягти таких же, якщо не найбільших результатів, як несправедливість, з якою потрібно боротися всіма можливими засобами.

Найбільш важливим аргументом в боротьбі з шахрайством є наслідок наступного морального аспекту: шахрайство становить загрозу для ігрових спільнот. Чесні гравці обурені можливістю шахрайського виграшу і, за відсутності ефективної протидії з боку управління сервером, вони можуть просто покинути гру.

2.3.3 Технічні причини

Другорядними критеріями, які визначають шахрайство, а також в значній мірі визначають ступінь його поширення, є технічні причини, які допускають безкарність шахрайства:

Відсутність або недорозвинення античит-програм на сервері (потрібно боротися з програмами).

Відсутність контролю гравця - на вашому власному комп'ютері гравець може робити все що завгодно.

Відсутність явних доказів шахрайства. Якщо хтось отримує високий рейтинг, він може бути як сильним гравцем, так і шахраєм.

Відсутність заходів протидії. У ряді онлайн-ігор у середнього гравця немає інструментів, що дозволяють швидко впливати на гравця, викритого в шахрайстві.

Шахрайство в онлайн-іграх, таких як спортивний допінг: обидва дають несправедливу перевагу один одному, і від них практично неможливо позбутися - їх можна тільки мінімізувати.

2.3.4 Невизначеність поняття шахрайства

Перевага шахрайства полягає в тому, що поняття шахрайства (як і багато інших понять в життєвій практиці) не може бути формально визначено і, отже, заковано відповідним чином. Стандартне визначення «шахрайство - отримання несправедливої переваги в грі» - не є точним, оскільки не визначає поняття «несправедливого».

Оскільки з моральної точки зору (яка наказує боротьбу з шахрайством) класифікація конкретної діяльності як шахрайства визначається цілями цієї діяльності, а не методами, немає чіткого поділу між «добром» і «шахрайством». Погані вчинки. Одні і ті ж дії можуть бути зроблені як для настройки гри під себе, так і для отримання переваги. Наприклад, можна прибрати ефект туману, як для збільшення швидкості гри, так і для спостереження за противниками на більшій відстані. Тобто дії гравця як шахрая можуть бути класифіковані тільки з будь-якої ймовірністю. Це можуть бути дії, пов'язані зі змінами налаштувань управління і

графіки («білі» методи), і можуть бути явним шахрайством, таким як використання помилок («чорні» методи).

Все між ними сіре. Такі суперечливі дії включають:

- 1) переналаштування драйверів;
- 2) тонке налаштування графіки через консоль;
- 3) скрипти, які виконують багато команд за одне натискання кнопки;
- 4) мультитовариство.

Навіть до використання помилок гри ставляться по-різному: одні помилки (наприклад, rocket jump) перетворюються в стандарт (стають фічами), інші виявляються поза законом.

Вирішення цієї проблеми адміністраціями серверів повністю аналогічно відповідним діям в інших сферах життя, які піддаються формалізації: адміністратори кодифіковані очевидні ситуації (явно повідомляють, що дозволено і що заборонено), а в інших застосовують прецедентне право - розбирають кожен випадок індивідуально.

2.3.5 Основні типи шахрайства в онлайн-іграх

2.3.5.1 Загальні для різних жанрів

Використання помилок гри

В цей розділ входять як явні баги, так і просто недоліки ігрової моделі.

Відомий приклад — глушник BFG9000 в Doom. За задумом розробників, BFG, перш ніж вистрілити, він має довго заряджатися і видавати характерний звук. Гравці знайшли спосіб заглушити цей звук.

У гоночних іграх пілот може з'їхати з траси з двох причин: або він спробував зрізати поворот (за це гонщику показують чорний прапор), або просто не впорався з керуванням (в цьому випадку гонщик сам себе покарав). Для того, щоб

визначити, зрізання це, або виліт, застосовуються евристики на зразок «машина на швидкості не менше X в'їхала в зону Y ». Якщо на деякій трасі зона чорних прапорів розташована не дуже вдало, то може статися, що зрізання повороту пройде безкарно.

Деякі помилки ігрової механіки стали стандартом (наприклад, Rocket jumping в Quake).

На змаганнях помилки гри «виправляються» посиленням спортивним регламентом. Наприклад:

Змагання серії Fast Prix Fast Track Racing: дозволені тільки ті повтори, в яких автомобіль ніколи не перетнув трасу. Зазвичай правило: «в будь-два рази на трасі повинно бути не менше двох коліс; дорога вважається асфальтовим полотном ». Зрозуміло, що ця техніка не підходить навіть для коротких гонок.

У Dota один сильний персонаж може серйозно змінити баланс сил. Тому змагань суворо заборонено передавати артефакти (крім деяких дешевих) іншим гравцям. Згодом можливість використовувати чужі речі заблокувалася самою грою.

Окремим випадком використання недоліків гри є використання її поведінки при втраті зв'язку, якщо результат змагання залежить від факту відключення. У більшості сучасних ігор ця проблема вирішується: змагання організовано таким чином, що будь-який з гравців, відключилися, завдає шкоди тільки собі і своїй команді (в важливих змаганнях призначається перегравання). Обман, змова і шпигунство

Ці види шахрайства виходять за рамки ігрового процесу і зовсім нічим не відрізняються від аналогічних дій в інших сферах життя. Приклади:

1) Один гравець може навмисно програти іншому в шахи, щоб підвищити рейтинг напарника (зниживши при цьому свій).

2) Спостерігач, який розглядає рівень в режимі вільного польоту, може підказувати напарникам положення суперників.

3) У грі, де можна вільно вибирати сторону, кілька дуже сильних гравців йдуть в одну команду, не даючи супротивникам жодного шансу виграти. Подібна тактика спрямована проти спільної мети більшості членів команди на публічному сервері — «побігати, постріляти» і отримати задоволення (передбачається, що згаданих сильних гравців меншість), і саме тому сприймається ними як нечесна.

4) У стратегіях в реальному часі набір учасників під час ходу гри, як правило, неможливий. Тому хтось може покинути гру, послабивши команду.

5) «Стримінг» — гравець в матчі може підслухати трансляцію цього матчу і зробити нечесні дії. Для протидії цьому в онлайн-трансляціях часто застосовують затримку.

Деякі способи обману:

- 1) фішинг;
- 2) провокування гравця зробити небезпечну дію на своєму комп'ютері (наприклад, натиснути Alt+F4 або ввести команду unbindall).

Ті, з даних способів, яким можна запобігти технічними засобами (огляд карти третіми особами, «накрутка» рейтингу), запобігаються або робляться неефективними. Інші ж, можливо відстежувати і запобігати тільки поза ігровими засобами — наприклад, за допомогою адміністраторів сервера (для цього в гру вбудовують можливість «збору доказів» — скріншотів, логів, демо-роликів) або просто вибором надійних партнерів.

Симуляція високого рівня допомагає в іграх з моделлю зв'язку «рівний з рівним».

Вихідні пакети отримують тимчасову мітку на 0,1-0,2 секунди раніше, ніж вони реально вийшли. З точки зору часу в середині гри, шахрай передбачає дії суперників. З боку це виглядає так: гра затримується на короткий час (як ніби у

супротивника з'явилися затримки зв'язку); коли затримка проходить, ваш персонаж вже вбитий.

Є лише два способи боротися з цим: заносити подібних «лаггерів» в чорні списки і перевіряти програму на шахрайство. Існують протоколи, спеціально розраховані на боротьбу зі штучними затримками пакетів, але їх недолік — істотне уповільнення гри в критичні моменти.

У деяких іграх FPS при грі онлайн, таких як Quake, ви часто можете бачити різні звуки гравців, ця різниця впливає на точність стрільби. Гравець з меншою кількістю пінгів отримує перевагу, однак гравець з меншою кількістю пінгів не є читом. Існують програми, які штучно збільшують затримку, тому гравець може збільшувати пінг і грати на рівних. Тим не менш, багато людей відмовляються ставити затримки на тій підставі, що «вони не винні в поганому інтернеті інших гравців». Таким чином, конкурентна складова падає і гра вважається некоректною, що викликає масу неприємних емоцій. Але слід зазначити, що пінг безпосередньо залежить від провайдера програвача, а також від кількості FPS - від потужності комп'ютера. Тому він не зобов'язаний штучно підвищувати свій пінг тільки тому, що у конкурентів гірше інтернет, точно так же, як ніхто не зобов'язаний встановлювати на свій комп'ютер відеокарту гіршої якості, тому що конкуренти не дають прийнятний номер FPS.

Гравців з поганим інтернетом називають гравцями які відмовляються ставити затримки, «пінгчитерамі».

2.3.5.2 В шутерах

Wallhacking дозволяє гравцеві бачити крізь тверді або непрозорих об'єктів і / або маніпулювати, чи навіть видаляти текстури, щоб заздалегідь знати, коли супротивник збирається прийти через той або інший прохід. Такий ефект можна

досягти шляхом створення прозорих текстур, або модифікації ігрових карт для вставки багатокутного отвору, в тверду стіну.

Wallhacking спирається на той факт, що сервер FPS зазвичай посилає сиру, позиційну інформацію для всіх гравців у грі, і залишає її для клієнта в 3D візуалізації, щоб приховати супротивників за стінами. Wallhacking використовується ширше, ніж Aimbot, оскільки він забезпечує майже непомітне пересування, що також набагато надійніше, бо ви будете знати, де знаходиться супротивник.

Мап-хак - це загальний термін, який відноситься до способу, або сторонніх програм, що дозволяє користувачеві бачити більшу кількість рівнів, ніж передбачалося розробником. Мапхакером є користувач, який свідомо користується таким способом проходження гри, або використовує програму в контексті відповідної гри.

Загальним аспектом стратегії в реальному часі є часткове обмеження гравця або повна нездатність бачити окремі ігрові об'єкти (зазвичай продукти і конструкції) поза увагою; ця концепція контролюється механізмом, відомим як туман війни. Злом карти зазвичай дозволяє користувачеві обійти цей механізм або повністю видалити його, помістивши в туман об'єкти, які зазвичай не було видно. У багатокористувацьких режимах це дає чітку перевагу перед іншими гравцями. Перевага може бути значним для стратегій реального часу середнього розміру, заснованих на використанні ножиць для паперу, механізму для балансування окремих об'єктів, різних сильних і слабких сторін.

Екстрасенсорні здібності в відеоіграх відображають контекстну інформацію, таку як здоров'я, ім'я, екіпірування, статус та / або орієнтація інших учасників гри, які зазвичай приховані від інших гравців. Це можна зробити, читаючи пам'ять програми зовнішньої програмою або перехоплюючи і декодуючи пакети. Вони подорожують між клієнтом і сервером. Перевага цього читання в тому, що воно

дозволить користувачеві отримати повну інформацію про своїх супротивників, що дозволить їм приймати рішення про вибір або уникнення менш або більш небезпечних супротивників відповідно. Це надзвичайно важко виявити і довести, але це, безумовно, вважається шахрайством.

Aimbot (іноді званий «авто-мета») — є одним з видів комп'ютерного бота, використовуваного в багатокористувацькому шутері від першої особи. Найбільш поширеним у шутері від першої особи, є мод TriggerBot, який стріляє автоматично, коли супротивник з'являється в полі-огляду .

Aimbotting заснований на тому факті, що клієнтський комп'ютер кожного гравця одержує інформацію про всі інші гравців: чи бачать вони супротивників з позиції гравця на ігровому полі чи ні. Орієнтація - це просто питання визначення місця розташування будь-якого супротивника щодо позиції гравця і націлювання зброї гравця на ціль. Орієнтування працює, чи знаходиться ворог за стінами або занадто далеко, щоб бачити прямо. Гра дозволяє м'ячу проникати на поверхню зі зменшеним втратою. Кодом можна маніпулювати так, щоб кулі проникали в нескінченну кількість матеріалу, і пошкодження не було виправлено, щоб гарантувати вбивство супротивника. Це серйозно збільшує несправедливу перевагу Aimbot. Деякі боти також використовують стратегію «перший постріл»: вони дізнаються, де ви знаходитесь, і почнуть стріляти в вас.

Деякі сервери дозволяють неактивним глядачам спостерігати за грою з точки зору активних гравців. Якщо ви граєте, націлених Aimbot, то для глядача буде абсолютно очевидно, наскільки неприродно точним є відстеження позиції. Деякі боти намагаються приховати цей очевидний вибір мети, використовуючи метод Silent Meta, який в основному використовується в Call Of Duty. Використовуючи цей метод, боти можуть вбивати ворогів непоміченими. Деякі боти і тригери кидаються в очі, в той час як інші намагаються приховати від глядачів те, що вони в даний час використовують чит, за допомогою ряду методів,

таких як затримки стрільби, щоб приховати той факт, що він стріляє, коли ворог знаходиться на гарматі. Деякі додатки Triggerbot можна легко включати і вимикати за допомогою миші або клавіатури.

Штучні лаги. В ігровій моделі відставання між рівноправними вузлами - це те, що відбувається, коли потік даних між одним або декількома гравцями сповільнюється або переривається, викликаючи заїкання в русі, і противники починають здаватися хаотичними. За допомогою перемикача затримки програваач може перервати завантаження з клієнта на сервер. Мета полягає в тому, щоб отримати перевагу над іншим гравцем без взаємності. Противники уповільнюють або зупиняють рух, дозволяючи користувачеві легко перехитрити їх за допомогою перемикача люфту. З точки зору противника, гравець, використовуючи пристрій, може переміщуватися, стати невидимим або непереможним, в той час як противники відчують затримки анімації.

Термін «затримка перемикування» охоплює безліч методів, які порушують мережеве з'єднання між клієнтом і його сервером. Один спосіб досягається шляхом підключення фізичного пристрою, званого «перемикачем затримки». Розробники ігрових приставок почали вводити захист від перемикачів затримки у вигляді детекторів напруги, які виявляють зміни напруги при натисканні на перемикач. Деякі виробники взяли контрзаходи, щоб обійти або обдурити детектор. Цього також можна домогтися, просто від'єднавши інтернет-кабель, що йде до клієнта, що призведе до порушення з'єднання інтернет-плеєра. Інші методи, звані програмним забезпеченням або бездротовим комутатором, пов'язані з використанням комп'ютерних програм. У цьому методі шахрай запускає додаток на комп'ютері, підключеному до тієї ж мережі, що і клієнт. Додаток розриває з'єднання між клієнтом і сервером. Однак ніхто не може робити це необмежену кількість часу.

2.3.5.3 У стратегіях реального часу

Чити можна розділити на ті, які змінюють онлайн-трафік (використання помилок) і не змінюють, а лише змінюють відображення існуючої інформації на клієнті (злом карти). Інші можуть бути виявлені тільки програмою захисту від злomu на стороні клієнта.

MapHack

Вся карта відкривається гравцеві, як в одиночній грі. Це може бути або визначено клієнтської анти-хакерської програмою, або іноді видає поведінку гравця (занадто швидка реакція на речі, які він може не знати).

Native hack - повідомлення, коли у противника є певний юніт / будівля / апгрейд, який переглядає його кількість грошей / юнітів.

2.3.5.4 У MMORPG

Один вид експлуатації помилок дублювання ігор / генерація предметів. Хакери постійно шукають «дірки» в захисті гри, завдяки яким ви можете отримати віртуальну власність безкоштовно.

Мультиводство - це ситуація, коли у одного гравця буде кілька ігрових персонажів. Власники гри по-різному ставляться до мультиводства: деякі забороняють це, інші заробляють на цьому, вимагаючи оплату за кожен аккаунт. Оскільки в більшості випадків мультиводства служить для подолання ігрових обмежень, які створюють незручності, але не впливають на баланс сил (наприклад, в Ultima Online гравці з «поганий» репутацією не можуть ходити в міста), він ближче до «білих» методи.

Причиною використання ботів є наявність рутинних дій в грі, логічною ідеєю яких є їх автоматизація. У MMORPG боти дозволяють виконувати ряд важливих для ігрового процесу дій набагато швидше і ефективніше, ніж це можливо без них, і тому дають недосяжне перевага «звичайними засобами». Отже,

в більшості MMORPG використання ботів заборонено ліцензійною угодою і погрожує закрити обліковий запис, а автори та адміністратори використовують різні методи для моніторингу таких дій.

Моральна сторона використання ботів неоднозначна: в частині, яка служить для автоматизації механічних дій («молоти»), це явно «біла» техніка, яка вказує на недосконалість гри з точки зору ігрового дизайну (і дійсно: з Розробники онлайн-ігор розробили «Штатний розклад »для автоматизації багатьох дій, які перша MMORPG повинна була виконати вручну). Якщо автоматизовані дії є основною частиною ігрового процесу (наприклад, «прокачування» персонажа), «звичайні» гравці не зможуть конкурувати з ботами, тобто такі боти - «легкий» спосіб домогтися перемоги.

Повна протилежність мультфільму, змінює правила гри, - це коли кілька людей по черзі грають одного і того ж персонажа. По суті не відрізняється від використання ботів; Різниця полягає в тому, що в обмін на повну невразливість проти антиботаніческого обладнання гравець або отримує персонажа, яким він володіє тільки частково, або несе значні грошові витрати на реальні гроші (якщо він викупує персонажа зі своїх наступників).

2.3.6 Протидія шахрайству

Більшість видів шахрайства є результатом технічних або ідеологічних недоліків гри: гравці можуть впливати на поведінку гри понад можливостей ігрового процесу, в грі є помилки або відсутні будь-які необхідні функції і т. Д. Вони мають справу з ці види шахрайства, усунення їх причин або створення бар'єрів для шахрайства (злом, контроль активності).

Найважливіші розрахунки виконуються на сервері.

Антич-методи:

Шифрування ігрових пакетів. Це рідко реалізується (або за допомогою слабкого алгоритму підпису), так як збереження процесорних годин, витрачених на шифрування, може зробити гру швидше або красивіше.

Контроль цілісності ігрового коду і даних. У більшості випадків він використовується при підключенні до сервера (щоб уникнути проблем, наприклад, через різницю в клієнтських версіях), але це також необхідно під час роботи клієнта - дані можуть бути змінені в уже запущеному процесі (ця перевірка часто не використовується).

Контроль активності гравців (разом з грою працює програма-агент, намагається виявити «підозрілу» активність на комп'ютері користувача. Приклад — Cheating-Death. Такі програми можуть також дозволяти адміністратору приховано спостерігати за гравцем). Інша програма, мутатор AntiTCC для Unreal Tournament 2004, стежить за точністю наведення гравця на противника і може навіть відключити його від сервера, в разі виникнення підозр (наприклад, в разі різкого точного наведення або безпомилкового тривалого проходження за противником). Відомі рішення: DMW Anticheat, PunkBuster, Valve Anti-Cheat, Warden.

Оновлення онлайн-ігор: гравці не можуть потрапити на сервер без відповідних оновлень.

Типи шахрайства, що виходять за рамки ігрового процесу, а також проблеми, які нездоланні для гри, можуть вирішуватися тільки поза грою, зокрема, адміністративні:

Спостереження за «підозрілим» гравцем і / або збір доказів (приклад: повторення вбивцею кількох секунд часто дозволяє вам визначити, чи використовував вбивця стрілка; якщо це так, ви можете записати демонстрацію з «наглядом» і відправ це адміністратору). Це можуть зробити як звичайні гравці, так і адміністратори.

Жорсткі заходи для шахраїв. Наприклад, в іграх Blizzard шахрайства забороняється входити на якийсь сервер, а в мережевих іграх на движку Source при використанні певних типів шахрайства на захищених серверах Valve Anti-Cheat забороняє користувачам грати на таких серверах. Однією з крайніх заходів впливу на шахрая є блокування інтернет-провайдера (якщо доведено, що дії шахрая призводять до несанкціонованої зміни налаштувань сервера і / або його шкідливій роботі), тобто прирівнюють його дії до спроби злому. , Драконівські заходи мають один недолік: невинних можна прийняти за покарання.

Відповідні інструменти (скріншоти, записи, демонстрації) вбудовані в гру для збору доказів шахрайства. Через складність фальсифікації доказів сервер може реєструвати себе, легко фальсифікуючи дані (наприклад, всі репліки символів, так званий податок).

В важливих змаганнях можливість шахрайства зводиться до мінімуму за рахунок участі в комп'ютерних клубах за апаратне і програмне забезпечення під контролем організаторів.

У 2008 році група шахраїв створила і розмістила в Інтернеті три види вірусів під загальною назвою Barrakuda (під назвами SAMP-SPY, sampmanager, барракуга). Вірус був розроблений, щоб зруйнувати комп'ютери нечесних гравців, що грають на серверах мережевих ігор GTA San Andreas Multiplayer (SA-MP). При запуску завантаженої шахрайської програми на комп'ютері користувача всі відомі антивірусні програми були відключені, після чого в системі були заражені всі файли з розширенням .EXE, видалена група системних папок, відключено відновлення системи і Система користувача була зупинена, в результаті чого Windows повністю порушена без відновлення даних. За даними інтернет-джерел, вірус був завантажений більше 15 тисяч разів [12].

2.3.7 Легалізація шахрайства

Широке поширення і труднощі визначення фактів і поняття шахрайства призводять до легалізації технічного шахрайства (на відміну від соціального).

Легалізація проводиться наступними засобами:

- 1) Надання гравцеві можливості помітити себе як шахрая, точніше вже як гравця, який легально застосовує технічні засоби, щоб модифікувати функціонал гри. Такі гравці беруть участь в грі в своїй категорії, подібно ваговій категорії в фізичних видах боротьби. При цьому приховане застосування технічних засобів в бою зі звичайними гравцями є неприпустимим так само, як і в іграх без легалізації.
- 2) Скасування заборони на зміну клієнтської частини гри.
- 3) Публікація вихідних кодів клієнтської частини і API сервера.
- 4) Створення та підтримка середовища програмування клієнтської частини і взагалі інфраструктури розробки.

3 ДОСЛІДЖЕННЯ НЕЛЕГАЛЬНОГО ДОСТУПУ ДО РЕСУРСІВ У ВІДЕОІГОРАХ

3.1 Види нелегального доступу

Існує 5 видів нелегального доступу в ігровому додатку:

- а) Доступ до програмного коду
- б) Доступ до текстур та шейдерів
- в) Доступ до моделей
- г) Доступ до збережених даних
- д) Доступ до пам'яті

Розглянемо їх більш детально.

Якщо зловмисник отримає доступ до програмного коду, то він зможе не тільки прочитати логіку гри та деякі її секрети, але також вкрати та використовувати ваші алгоритми та технології, що є інтелектуальною власністю.

Багато розробників не захищають власний код своїми силами, а покладаються на сили ігрового рушія. Але не всі ігрові рушії успішно справляються з цією задачею. Наприклад, Unity дуже погано приховує програмний код, навіть не змінює його структуру чи назви змінних. Тому доступ до нього можна отримати зробивши звичайну декомпіляцію .NET бібліотек.

Існує багато способів захисту програмного коду, один з них, для розробників на рушії Unity – це використання Unity 3D Obfuscator. Ця утиліта шифрує вашу інтелектуальну власність та робить неможливою атаку через декомпіляцію бібліотек [13].

Доступ до текстур і шейдерів – цей доступ дає зловмиснику можливість отримати текстури, матеріали гри та їх налаштування. Текстури можна використовувати, для створення своїх рішень або звичайного плагіату.

Шейдери є наборами інструкцій, і ці інструкції виконуються одночасно для кожного пікселя на екрані. Це означає, що код повинен працювати по різному в залежності від положення пікселя на екрані.

Текстура – це зображення, що відтворює візуальні властивості будь-яких поверхонь або об'єктів. На відміну від малюнка, до текстурі не застосовуються норми і вимоги композиції, оскільки текстура сама по собі художнім твором не є, хоча і може іноді виступати домінантою в художньому творі.

Отримання доступу до текстур більш складний процес та для кожного ігрового рушія потрібен індивідуальний підхід.

Якщо розробник використовує ігровий рушій Unity, то існує програмне забезпечення Unity Assets Explorer, яке дозволяє «викачати» з ігрового додатка усі текстури, матеріали та шейдери.

Доступ до моделей – більшість ігор на теперішній час використовують 3D графіку. А любий 3D об'єкт використовує моделі.

Модель об'єкта – це новий об'єкт, що відображає властивості даного об'єкта, суттєві для його дослідження.

Моделювання – процес створення та дослідження об'єктів на основі їх моделей.

Найчастіше більшість моделей зберігається по частинах, або навіть генерується тільки коли ігровий додаток запущений. Тому, щоб дуже довго не шукати окремі файли та намагатися їх правильно скласти зловмисники використовують наступний метод.

При запуску ігрового додатку на вашому персональному комп'ютері, усі дані, моделі персонажів та оточення завантажуються у відео пам'ять. І зловмисник може з цієї пам'яті отримати 3D моделі. Це дуже важкий та довгий процес. Але вже є рішення, які значно спрощують цей процес. Наприклад програмне забезпечення 3D Ripper DX. Воно дозволяє витягти всю інформацію з відео

пам'яті та зберегти у форматі, зрозумілому 3D-редакторам (наприклад, 3D Studio Max).

Доступ до збережених даних – це доступ до даних, які розробник зберігає на пристрої користувача. Такі дані можуть бути потрібні, для правильної загрузки прогресу гравця.

Зміна збережених даних може не тільки привести до не коректного функціонування ігрового додатка, але і зовсім зламати його. Частіше знищення таких даних надає значно меншої шкоди ніж її модифікація.

Будь-який ігровий додаток зберігає дані на пристрої, тож їх завжди можна знайти. Наприклад, більшість даних що розробник зберігає, використовуючи ігровий рушій Unity, перебуває у системному реєстрі. Щоб уникнути їх простого читання, ці дані, перед тим як зберегти на пристрої, шифрують. А після читання – розшифровуються.

Таким метод має свої недоліки. Головний недолік – це зниження продуктивності додатка. Також зашифровані дані можуть займати біль місця на пристрої користувача ніж звичайні, не зашифровані.

Доступ до пам'яті – най критичніший вид нелегального доступу, для правильного функціонування ігрового додатка. Коли завантажується гра, завжди усі дані загрузаються у оперативну пам'ять і у зловмисника є можливість змінити ці дані. Так користувачі в один момент можуть стати багатими, або навіть безсмертними [14].

3.2 Класифікація шахрайства у відеоіграх

Шахрайство в грі - це дія, яка змінює поведінку гравця, використовуючи спеціально створені або змінені програми або обладнання в онлайн-іграх, щоб дати одному гравцю перевагу над іншими.

Шахрайство у відеоіграх класифікується за кількістю користувачів, які можуть одночасно грати в гру.

Шахрайство з одним гравцем часто направлено на легке досягнення певних цілей. Наприклад, зробити персонажа багатим за один раз або нескінченну кількість раундів в зброю. Такі шахрайства тільки змінюють досвід гри одного гравця, тому розробники не несуть відповідальності за те, що всі користувачі отримують поганий досвід.

Шахрайство з розрахованими на багато користувачів додатками вже більш критично, і розробники ігрових додатків повинні відстежувати і / або карати таких шахраїв.

Основні види шахрайства в онлайн-іграх:

Використання ігрових помилок.

Цей розділ включає в себе як очевидні помилки, так і просто недоліки ігрової моделі. Окремим випадком використання недоліків гри є використання її поведінки при втраті зв'язку, якщо результат змагання залежить від факту відключення. У більшості сучасних ігор ця проблема вирішується: змагання організовано таким чином, що будь-який з гравців, відключилися, завдає шкоди тільки собі і своїй команді (в важливих змаганнях призначається перегравання).

Обман, змова і шпигунство

Ці види шахрайства виходять за рамки ігрового процесу і нічим не відрізняються від аналогічних дій в інших сферах життя. приклади:

Один гравець може свідомо втратити іншого шахіста, щоб підвищити свій рейтинг за командою (при цьому знизивши свій власний).

Спостерігач, який досліджує рівень в режимі вільного польоту, може повідомити своїм товаришам по команді становище своїх супротивників.

Деякі способи обману:

а) фішинг;

б) провокування гравця зробити небезпечну дію на своєму комп'ютері

Ті, з даних способів, яким можна запобігти технічними засобами (огляд карти третіми особами, «накрутка» рейтингу), запобігаються або робляться неефективними. Інші ж, можливо відстежувати і запобігати тільки позаігровими засобами — наприклад, за допомогою адміністраторів сервера (для цього в гру вбудовують можливість «збору доказів» — скріншотів, логів, демо-роликів) або просто вибором надійних партнерів.

3.3 Технічні засоби захисту авторських прав

Для захисту авторських і суміжних прав в електронному середовищі правовласники все частіше використовують різні технічні засоби. Технічні засоби захисту можуть бути класифіковані як технічні авторські права і пов'язані технічні права.

Захист авторських прав включає будь-які технології, технічні пристрої або їх компоненти, які контролюють доступ до додатка, що запобігає або обмежує такі дії, які не дозволені автором або іншим правовласником програми.

Існують обмежені випадки, коли використовувалася програма дозволено без згоди автора або іншого правовласника. Однак загальне правило вказує, що дії з додатком заборонені без дозволу автора або іншого правовласника, включаючи дії, спрямовані на:

а) здійснювати без дозволу автора або іншого правовласника дії, спрямовані на усунення обмежень на використання додатка, встановлених шляхом застосування методів захисту авторських прав;

б) виробництво, розповсюдження, оренда, тимчасове безоплатне користування, імпорт, реклама будь-якої технології, будь-якого технічного пристрою або їх компонентів, використання таких технічних засобів для

отримання прибутку або надання відповідних послуг, якщо будь-які дії унеможливають використання методи захисту авторських прав або ці методи не зможуть адекватно захистити ці права.

До технічних засобів захисту авторського права відносяться засоби:

а) що збільшують накладні витрати при копіюванні;

б) контролюючі розповсюдження копій (наприклад, шість географічних зон DVD-дисків);

в) що захищають від несанкціонованого доступу.

Інформацією про авторське право, яку заборонено змінювати законом, є:

а) інформація, що ідентифікує твір, автора або іншого правовласника;

б) інформація про умови використання твору, що міститься на оригіналі або примірнику твору, прикладена до нього або з'являється у зв'язку з повідомленням в ефір або по кабелю або доведенням такого твору до загального відома;

в) будь-які цифри і коди, в яких міститься така інформація.

Стосовно будь-яких творів не допускаються такі дії з інформацією про авторське право без дозволу автора чи іншого правовласника:

а) видалення або зміна інформації про авторське право;

б) відтворення, розповсюдження, імпорт з метою розповсюдження, публічне виконання, повідомлення в ефір або по кабелю, доведення до загального відома творів, щодо яких була видалена або змінена інформація про авторське право.

Наприклад, сучасні безпечні формати в більшості випадків містять інформацію про назву ліцензії, умови ліцензії на використання програми, таких як термін дії ліцензій, кількість можливих запусків, копіювання. Ця інформація також захищена законом про авторське право і авторське право [15].

3.4 Сучасні методи захисту від несанкціонованого доступу і мета його застосування

Сучасні компанії, окрім власних розробок, використовують технічний засіб захисту авторських прав під назвою Denuvo. Тільки він ще вважається досить важким для злому.

Denuvo Anti-Tamper, або Denuvo – технологія захисту від несанкціонованого злому, розроблена австрійською компанією Denuvo Software Solutions GmbH.

Ранні повідомлення про те, що Denuvo Anti-Tamper «безперервно шифрує і розшифровує себе так, що неможливо зламати», з'явилися в листопаді 2014 року. Однак Denuvo Software Solutions заявила, що технологія «не постійно зашифровує і розшифровує будь-які дані на носії. Для цього не було б ніякої користі з точки зору безпеки або продуктивності». Компанія не розкриває, як працює Denuvo Anti-Tamper. Китайська вarez-група 3DM стверджувала, що вони зламали Denuvo Anti-Tamper 1 грудня 2014 року. Група стверджує, що ця технологія включає в себе «64-бітну шифрувальну машину», яка вимагає криптографічні ключі, унікальні для конкретного обладнання кожної встановленої системи.

Пізніше, на початку грудня, та ж група випустила кряк для гри Dragon Age: Inquisition, яка використовує Denuvo Anti-Tamper для захисту Origin Online Access DRM корпорації Electronic Arts. Але це зайняло майже місяць, що незвично довго для комп'ютерних ігор. Компанія Denuvo визнала, що «кожна захищена гра в кінцевому підсумку може бути зламана». Ars Technica зазначає, що більшість законних продажів для великих ігор відбулися протягом 30 днів з моменту випуску, і тому видавці можуть розглянути Denuvo як успішну, маючи на увазі, що знадобиться значно більше часу для зламування ігор.

Ігри, захищені Denuvo, вимагають повторну онлайн-активацію для кожного оновлення апаратної частини. Denuvo обмежує кількість активацій на 4 оновлення обладнання протягом 24 годин

Деякі споживачі стверджують, що Denuvo Anti-Tamper скорочує термін служби твердотільних накопичувачів (SSD), записуючи надмірна кількість даних на диск. Denuvo Software Solutions у відповідь стверджує, що «Denuvo Anti-Tamper не часто зчитує або записує дані на носій» .

У грудні 2017 року стало відомо, що проблеми на серверах Denuvo завадили гравцям використовувати набуті в Steam ліцензійні копії ігор Batman: Arkham Knight, Middle-Earth: Shadow of War і Mad Max. При спробі запуску виникали повідомлення про помилки, тому доводилося багато разів повторювати процедуру підключення. Неполадки стосувалися тільки видань від Warner Bros., але це дало нові аргументи прихильникам за повну відмову від Denuvo, яка заважає грати в законно куплені ліцензії.

У 2018 році режисер і продюсер Кацухиро Харада повідомив про проблеми в PC-версії Tekken 7: падіння частоти кадрів через механізми антипіратського захисту, які постійно оновлюються.

Тому багато розробників видаляють захист Denuvo через деякий час або коли гру зламують і вона попаде у «вільний доступ».

У сучасний час технічні засоби захисту авторських прав в ігрових додатків використовують більш не для того, щоб зовсім захистити свою гру від «піратів», а тільки щоб вона не попала в вільний доступ у так зване «релізне вікно».

Релізне вікно – це проміжок часу з моменту випуску гри і наступні 1-3 місяці. Саме в цей час відбувається найбільша кількість покупок.

4 РЕАЛІЗАЦІЯ ІГРОВОГО ЗАСТОСУВАННЯ

4.1 Загальні відомості

Ігрове застосування під назвою «FirstPerson» призначено для демонстрації можливостей атак на багатокористувацьке онлайн ігрове застосування. Ігрове застосування реалізовано на рушії Unity з використанням скриптів на C#. Він може бути запущений на операційних системах сімейства Windows.

4.2 Функціональне призначення

Ігрове застосування реалізовано для демонстрації атак на багатокористувацьке онлайн ігрове застосування. За допомогою цього застосування можна рухатися в віртуальному просторі та стріляти вогнем.

4.3 Опис логічної структури

У ігровому застосуванні використовується плагін Forge, який реалізує мережевий зв'язок між клієнтами

а) При запуску ігрового застосування на екрані з'являється лобі, де можна або стати хостом або підключитися до хоста.

б) Після створення кімнати або підключення до вже створеної на екрані є два тексти, про бали двох гравців

в) При вході в гру створюється гравець.

г) При знищенні куба – набираються бали гравцю.

Більшість операцій виконується у Unity GameLoop за допомогою функцій Awake, Start та Update.

Гра містить 4 класи:

PlayerMovement:

void Update() – зчитується натиск клавіши гравцем для руху ігрової моделі.

MouseLook:

`void Update()` – зчитується рух маніпулятора миша для руху камери у ігровому додатку.

Gun:

`void Update()` – зчитується натиск на клавішу вистрілу.

`void Shoot()` – розрахунок цілі попадання та створення ефектів вистрілу.

Enemy:

`void Update()` – проводиться пошук гравця да проводиться рух до гравця

`void TakeDamage(float amount)` – проводиться нанесення шкоду при влученні.

4.4 Використовувані технічні засоби

Ігрове застосування не вимагає великих обчислювальних потужностей, але у різних конфігурацій комп'ютера може займати різний час запуску. Для реалізації поставленого завдання і тестування був використаний комп'ютер наступної конфігурації: CPU Intel Core i7 2.5 GHZ, Memory: 8GB, SSD: 256 GB.

4.5 Виклик і завантаження

Запуск застосування здійснюється через відкриття файлу «FirstPerson.exe».

4.6 Вхідні дані

Вхідні дані відсутні, ігрове застосування функціонує як звичайна гра.

4.7 Вихідні дані

Вихідні дані відсутні, тому що даний ігрове застосування було створено, щоб провести дослідження атак на багатокористувацький онлайн ігрове застосування.

5 РЕАЛІЗАЦІЯ АТАК ТА МЕХАНІЗМІВ ЗАХИСТУ ДЛЯ БАГАТОКОРИСТУВАЦЬКОГО ОНЛАЙН ІГРОВОГО ЗАСТОСУВАННЯ

5.1 Теоретичні основи

Ігровий рушій Unity коли створює ігри на PC генерує наступні файли:

ProjectName.exe - Запускатор проекту. У ньому міститься точка входу програми, яка запускається на рушії Unity.

UnityPlayer.dll - це бібліотека DLL містить весь код рушія Unity. Він також підписаний за допомогою сертифіката Unity Technologies, що дозволяє легко перевірити, що двигун не піддавався змінам.

* .pdb файли - Символьні файли для налагодження. Unity копіює їх до каталогу збирання, якщо увімкнути копіювання файлів PDB у вікні "Параметри збирання".

WinPixEventRuntime.dll - Ця бібліотека DLL дозволяє підтримувати профілі Windows PIX. Unity створює цей файл, лише якщо ви перевіряєте розробку розробки у вікні "Параметри збирання".

Папка ProjectName_Data - ця папка містить усі дані, необхідні для запуску вашого проекту.

Нас цікавить папка ProjectName_Data. У цій папці є інша папка Managed у якій є файл Assembly-CSharp.dll у якому зберігається увесь код, який створював розробник.

Саме цей файл будемо атакувати. А саме змінювати початковий код, щоб отримати вигоду у багатокористувацькому онлайн ігровому застосуванні[5].

5.2 Підготовка програмного та апаратного обладнання

Нам знадобиться 2 комп'ютери. Перший – звичайний користувач. Другий – користувач-зловмисник, який буде змінювати своє ПО.

На усіх комп'ютерах повинно бути встановлено застосування «FirstPerson». На другому комп'ютері додатково треба встановити ПО під назвою dnSpy та Macro Recorder.

5.3 Блокування можливості для створення декількох вікон додатку водночас

Мультибоксинг означає грати за кількох окремих персонажів одночасно в MMORPG. Цього можна досягти, використовуючи кілька окремих машин для запуску гри, або запустивши кілька окремих примірників гри. Мультибоксинг можна вважати формою обману.

У Unity заблокувати можливість для створення декількох вікон додатку можливо двома способами.

Перший, за допомогою правильною настройку проекту в Unity.

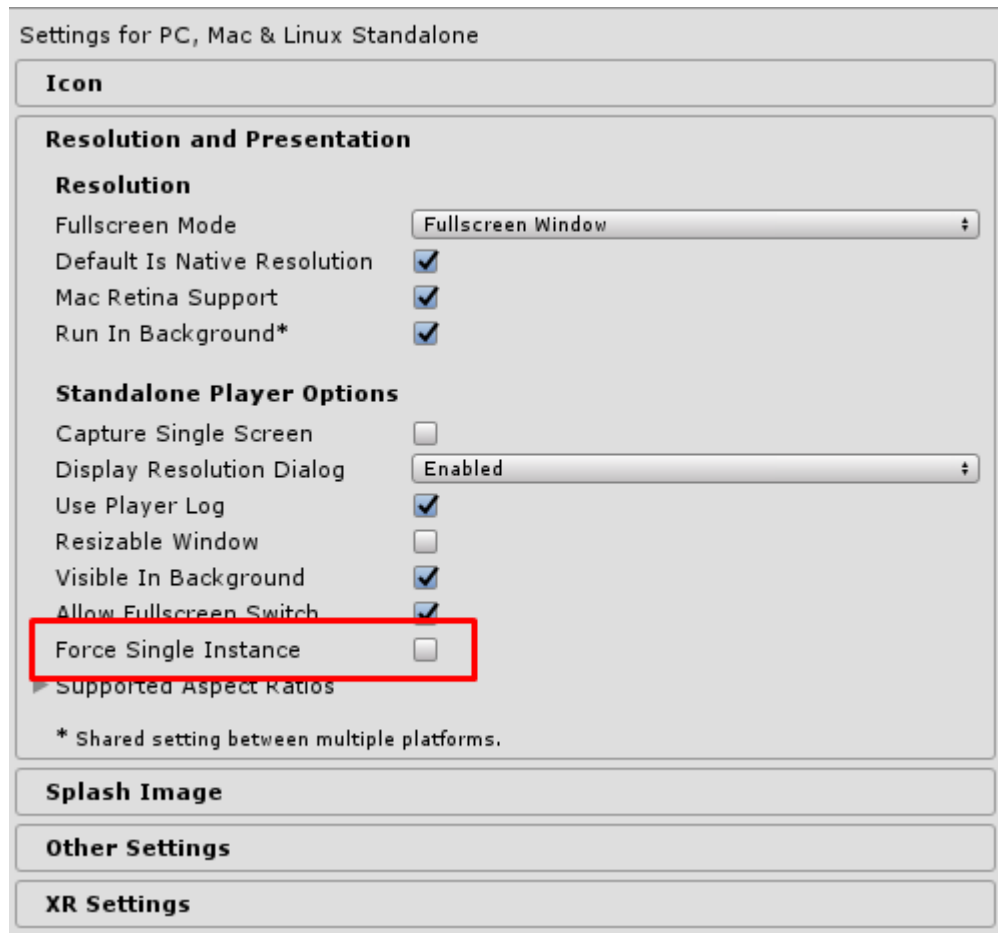


Рисунок 5.1 – Блокування запуску декількох вікон додатку

Другий, завдяки реалізації деякого коду.

Можливо дозволити користувачу створювати декілька вікон додатку, але грати він зможе лише в одному.

Для цього треба в модуль, який створює зв'язок с хостом додати код на перевірку вже запущеної гри. Якщо гра вже запущена, то не давати можливості підключитися до сесії.

5.4 Створення AimBot

Для створення AimBot треба буде модифікувати початковий код додатку. Для цього скористуємось програмним забезпеченням dnSpy.

Для модифікації початкового коду скористуємося ПО dnSpy.

1. Після запуску dnSpy треба вибрати асемблі, яка буди підвергатися модифікації.

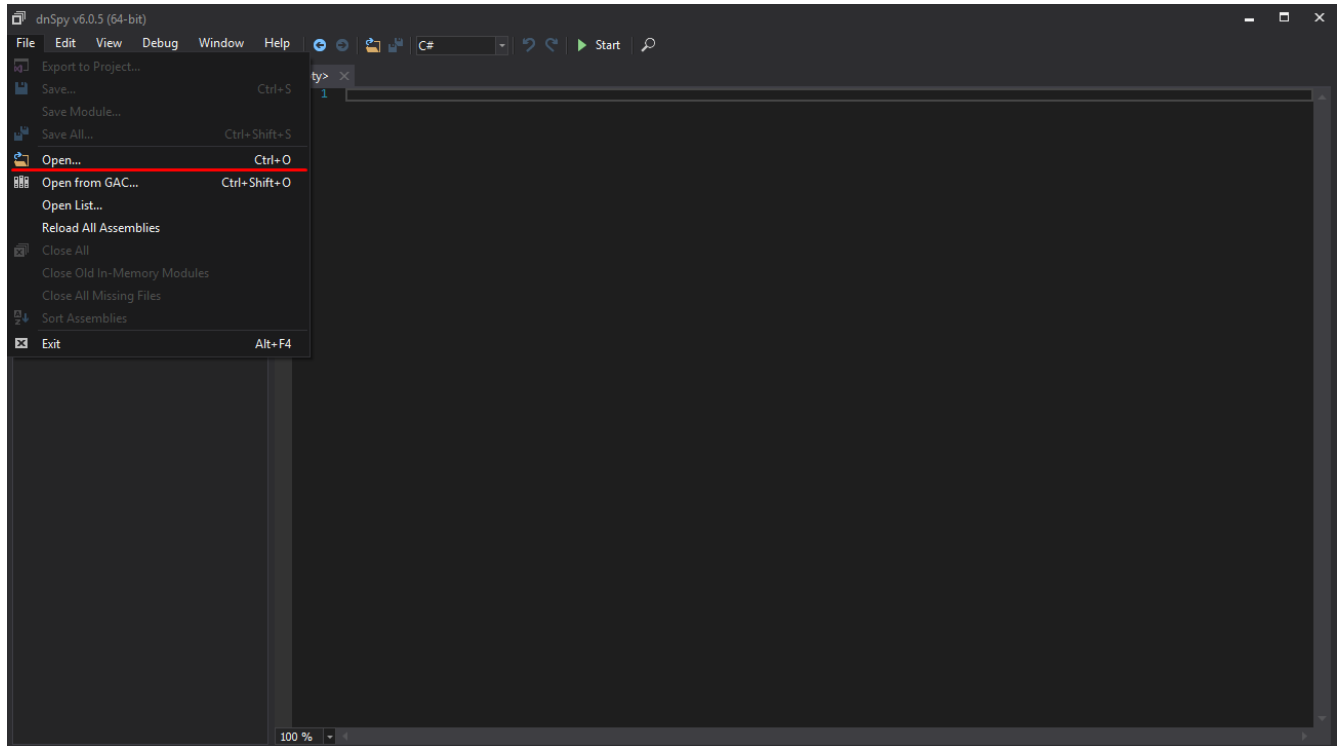


Рисунок 5.2 – Відкриття файлу для модифікації

2. Треба відкрити файл під назвою «Assembly-CSharp.dll» який знаходиться у FirstplayerShooter\ FirstPerson_Data\Managed\

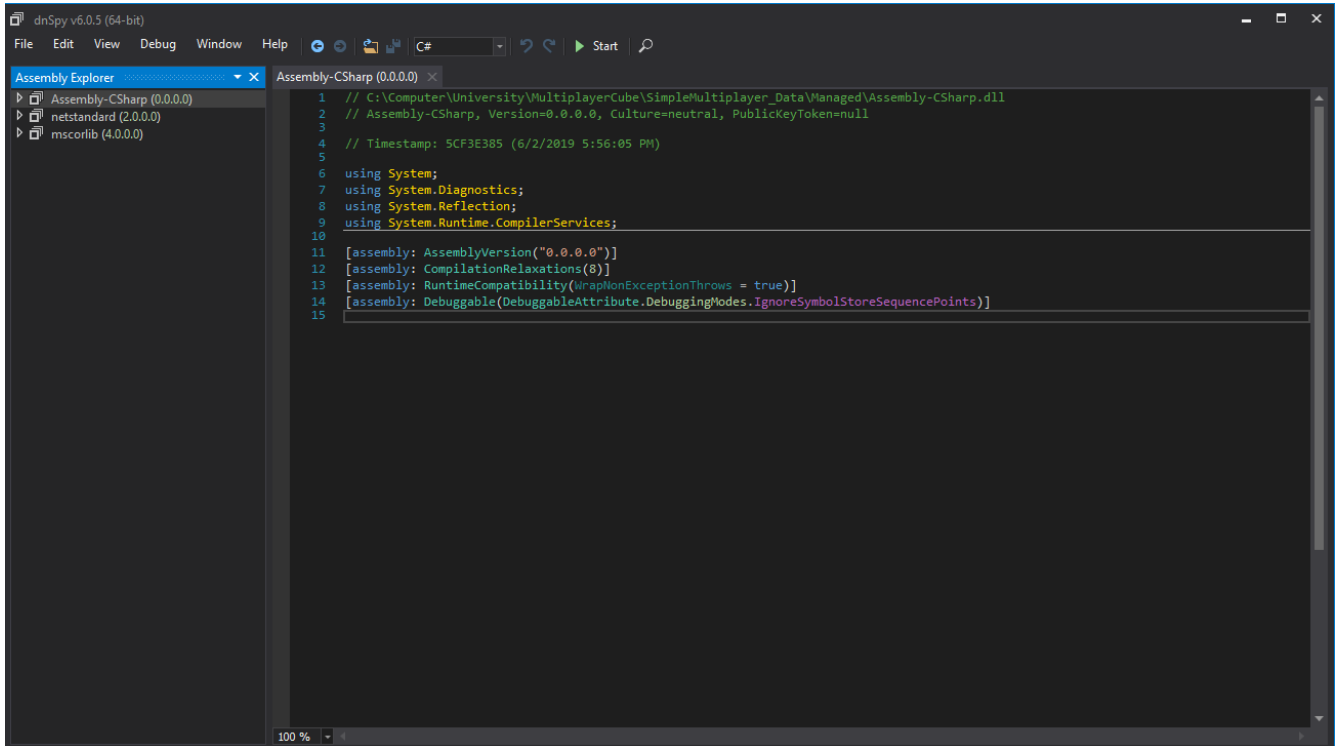


Рисунок 5.3 – Вигляд вікна з відкритим Assembly-CSharp.dll

3. Знаходимо клас PlayerMovement та MouseLook. Саме їх ми будемо модифікувати, щоб отримати створити AimBot.

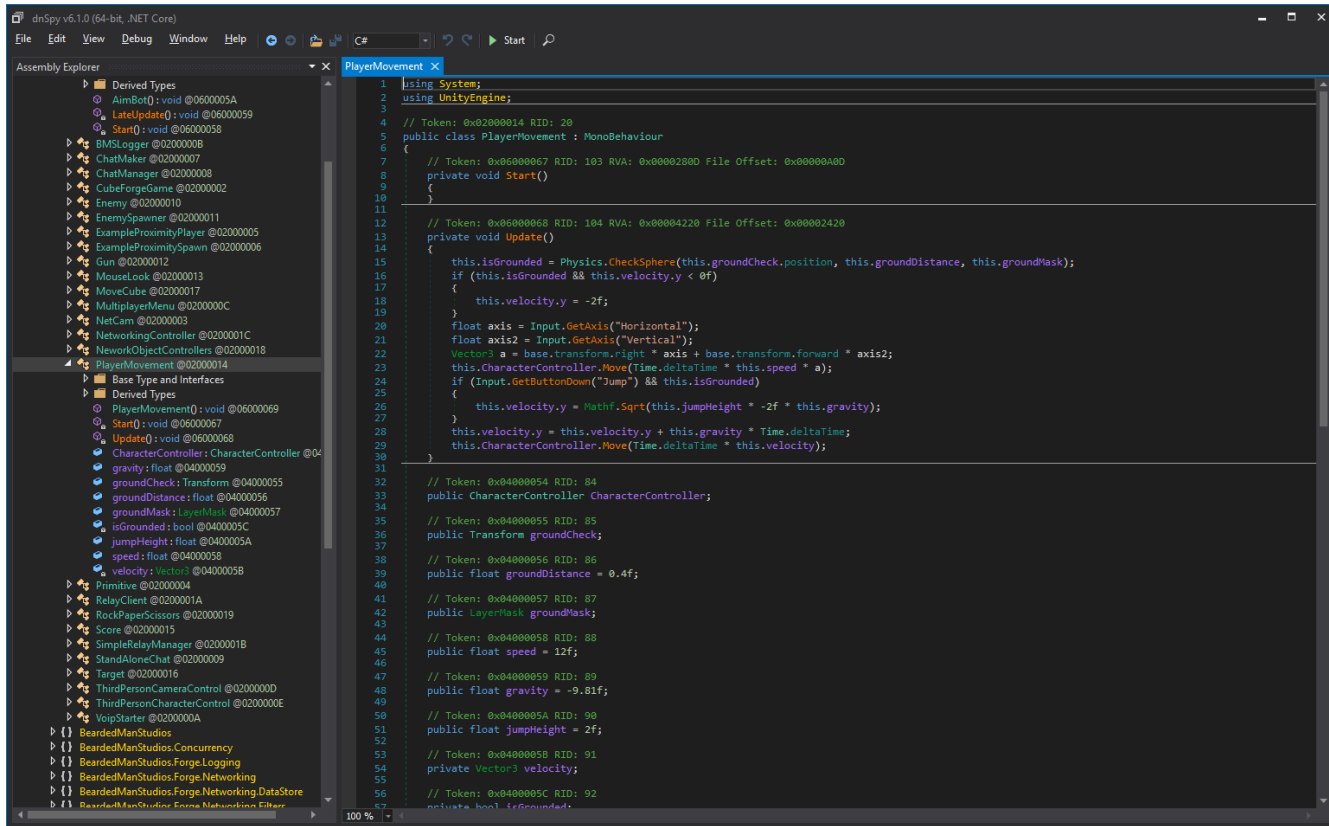


Рисунок 5.4 – Вигляд PlayerMovement у dnSpy

4. Знаходимо функцію яку ми хочемо змінити та вибираємо Edit Class (C#)

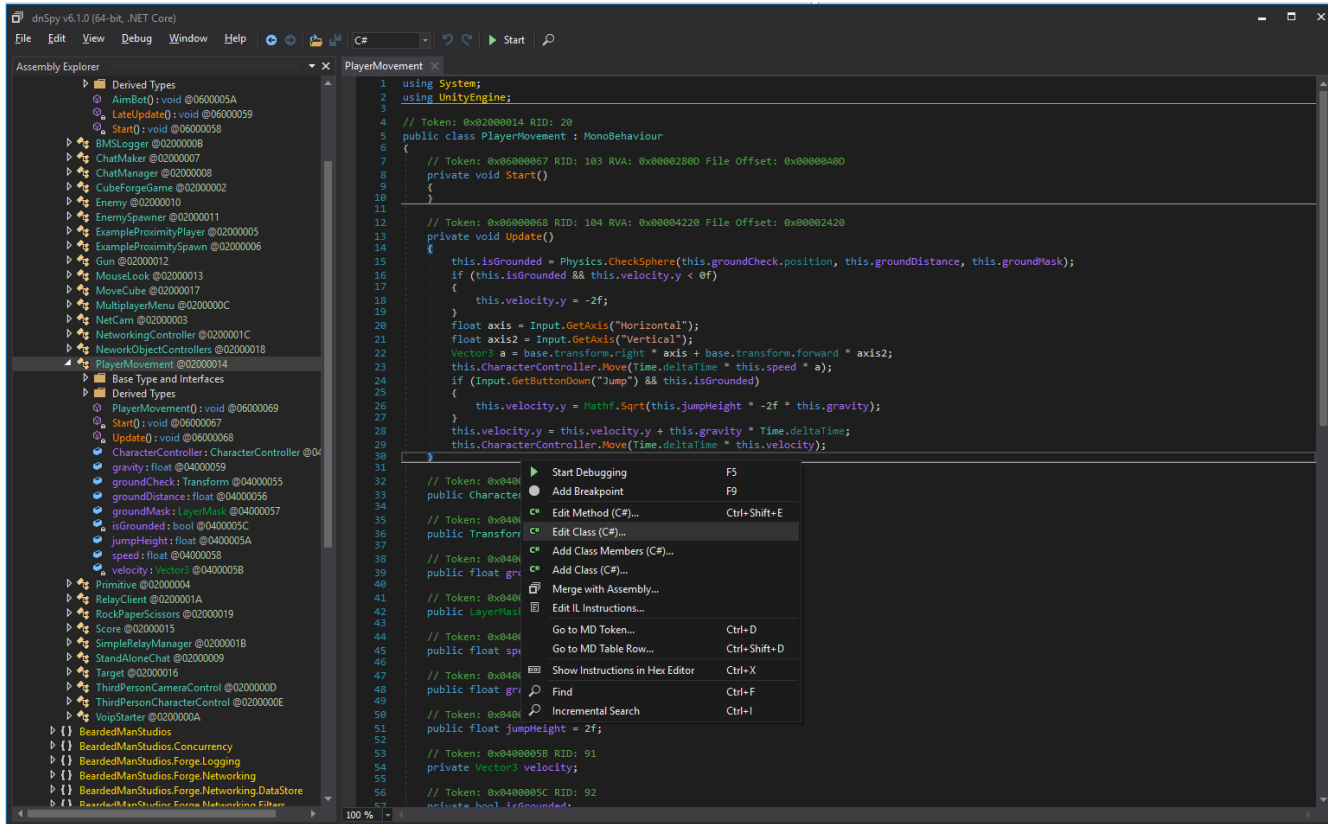


Рисунок 5.5 – Опція модифікація класу у dnSpy

5. Додаємо код як на рисунку 5.6 у клас PlayerMovement та MouseLook.

```

1 using System;
2 using UnityEngine;
3
4 // Token: 0x02000014 RID: 20
5 public class PlayerMovement : MonoBehaviour
6 {
7     // Token: 0x00000067 RID: 103
8     private void Start()
9     {
10    }
11
12
13     // Token: 0x00000068 RID: 104
14     private void Update()
15     {
16         this.isGrounded = Physics.CheckSphere(this.groundCheck.position, this.groundDistance, this.groundMask);
17         if (this.isGrounded && this.velocity.y < 0f)
18         {
19             this.velocity.y = -2f;
20         }
21         float axis = Input.GetAxis("Horizontal");
22         float axis2 = Input.GetAxis("Vertical");
23         Vector3 a = base.transform.right * axis + base.transform.forward * axis2;
24         this.CharacterController.Move((Time.deltaTime * this.speed * a));
25         if (Input.GetButtonDown("Jump") && this.isGrounded)
26         {
27             this.velocity.y = Mathf.Sqrt(this.jumpHeight * -2f * this.gravity);
28         }
29         this.velocity.y = this.velocity.y + this.gravity * Time.deltaTime;
30         this.CharacterController.Move(Time.deltaTime * this.velocity);
31
32     }
33
34     // Token: 0x00000074 RID: 118
35     private void LateUpdate()
36     {
37         GameObject go = GameObject.Find("Target");
38         if (go != null)
39         {
40             base.transform.LookAt(go.transform);
41         }
42     }
43
44     // Token: 0x04000054 RID: 84
45     public CharacterController CharacterController;
46
47     // Token: 0x04000055 RID: 85
48     public Transform groundCheck;
49
50     // Token: 0x04000056 RID: 86
51     public float groundDistance = 0.4f;
52
53     // Token: 0x04000057 RID: 87
54     public LayerMask groundMask;
55
56     // Token: 0x04000058 RID: 88
57     public float speed = 12f;
58
59     // Token: 0x04000059 RID: 89
60
61 }

```

Рисунок 5.6 – Зміна початкового коду класу PlayerMovement

6. Після модифікації натискаємо на клавішу Compile. Та зберігаємо наш модифікований файл

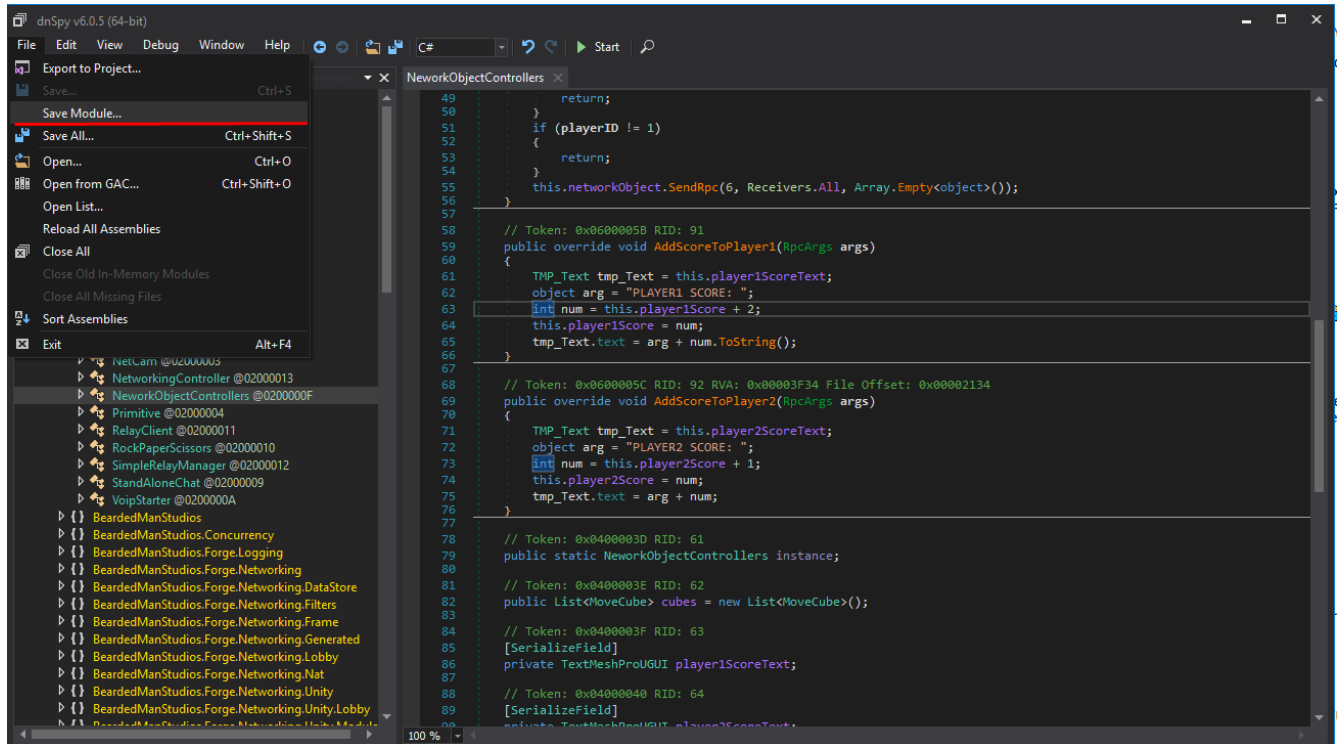


Рисунок 5.7 – Вікно зберігання модулю

7. Запускаємо застосування FirstPerson та дивимось на результат наших дій. Тепер ми завжди дивимось на противника і нам залишається лише натискати ліву клавішу миші.

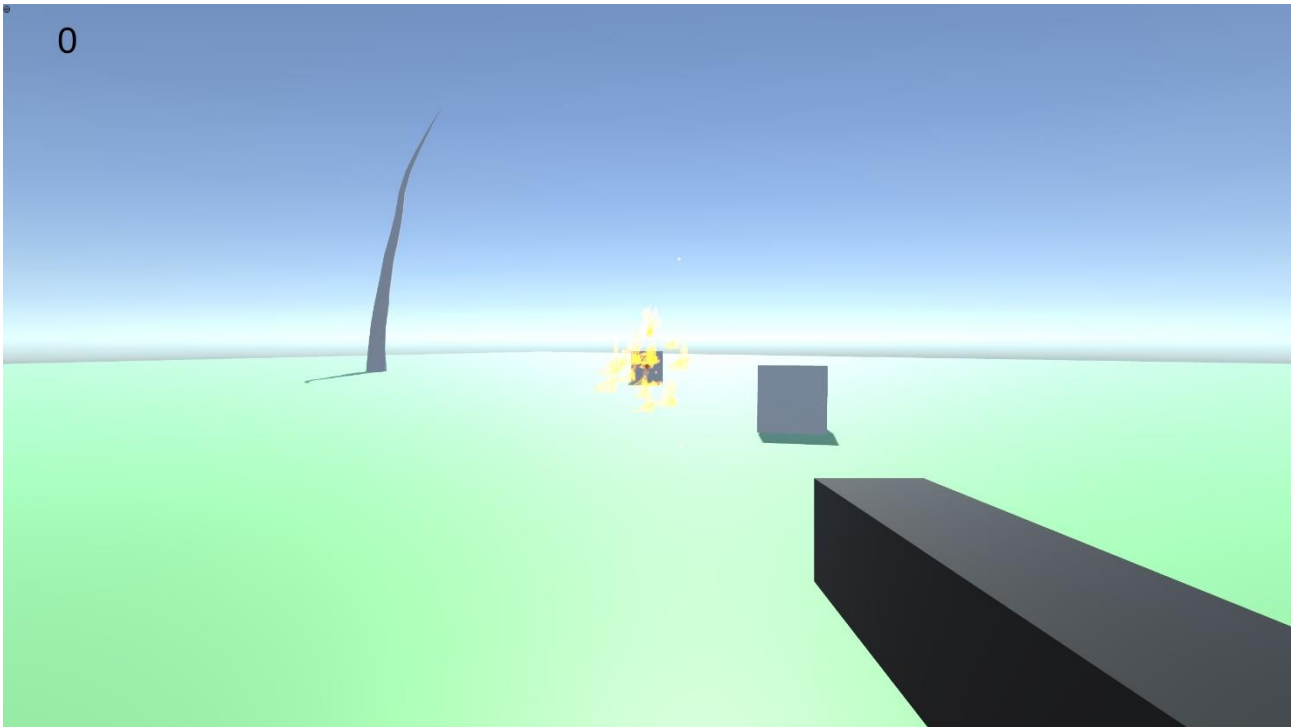


Рисунок 5.8 – Результат модифікації початкового коду

5.5 Створення бота, з записом дій користувача

Для запису дій користувача скористуємося програмою Macro Recorder

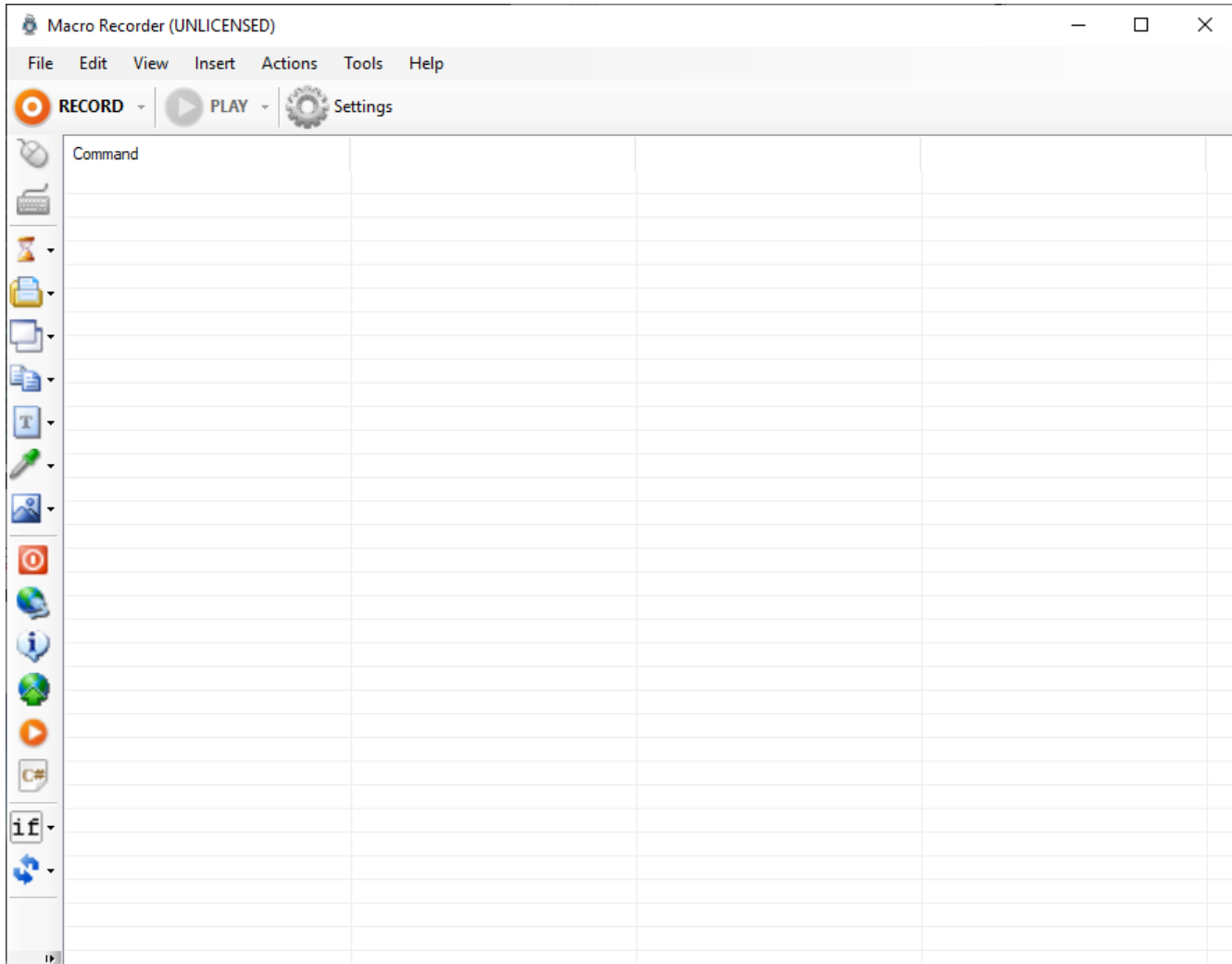


Рисунок 5.9 – Вигляд вікна Macro Recorder

При натиску на клавішу «RECORD» починається запис дій користувача.

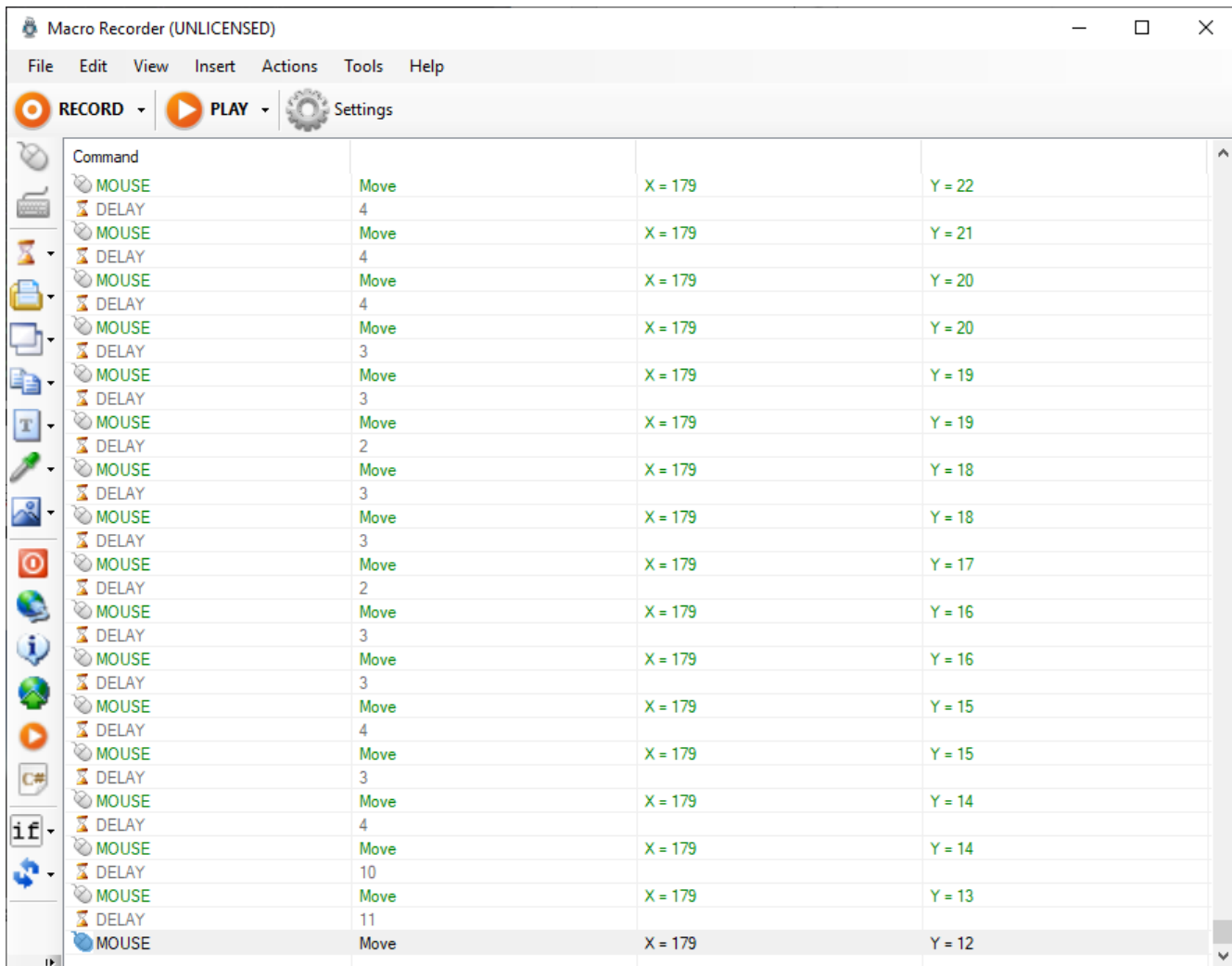


Рисунок 5.10 – Вигляд записаного макроси для гри FirstPerson

Щоб програти макрос натискаємо на клавішу «PLAY».

5.6 Висновки проведених атак

Атака зі створенням Aim боту – найпоширеніша атака у жанрі ігор с камерою від першої особи. Особливо якщо такі ігри мають кіберспортивні змагання. Захист від такої атаки можливо досягнути з використанням Obfuscator[16], або ускладненням кодової бази та використанням мов програмування, які неможливо декомпілювати.

Атака по створенню боту являється найважчою для виявлення. З покращенням методів виявлення – покращуються і методи поведінки ботів. Щоб однозначно сказати, що дії які проходять у клієнта виконує машина або людина, треба визначати ті дії, які може створювати лише людина.

Сучасна ігрова індустрія розвивається у напрямку хмарних технологій, тому атаку зі зміною початкового коду та запуск програм, які слідкують за пам'яттю додатка буде не можливо провести. А боти будуть працювати і якщо гра працює на сервері. Створення ботів та боротьба с ними будуть найбільш перспективними напрямками безпеки ігрових додатків.

ВИСНОВКИ

Дослідження атак та методів захисту від них показало, що від деяких атак захиститися неможливо, це такі атаки, як зчитування моделей або текстур з відео пам'яті, коли у неї завантажуються моделі з ігрового додатку. Від інших захист буде сповільнювати роботу гри, ускладнювати програмну реалізацію. Тож треба заздалегідь визначити які ресурси найважливіші для правильного функціонування ігрового додатку та захищати саме їх.

В даний час більшість розробників відмовляються від захисту своїх ігрових застосувань, тому що це не тільки коштовний процес, та й ще не йде на користь ігровому застосуванню. Деякі онлайн магазини погоджуються продавати ігровий додаток з умови, що у нього нема DRM (наприклад GoG.com).

Великі компанії використовують захист, але тільки на певний час, поки його не зламують. А потім видаляють захист з гри, щоб додаток функціонував як і було задумано.

У багатокористувацькому онлайн ігровому застосуванні без захисту зловмисник наносе шкоду не тільки своєму клієнту та розробнику, але і іншим користувачам даної відеогри, тому захист онлайн застосувань необхідне усім розробникам.

У майбутньому нас чекає хмарний геймінг. І тому створення ботів буде найбільш актуальною справою у зловмисників ігрової індустрії, а захист від них – для розробників ігрових додатків.

ПЕРЕЛІК ПОСИЛАНЬ

1. National Endowment for the Arts [електронний ресурс]. – Режим доступу: <https://www.arts.gov/>
2. The Designer's Notebook: Sorting Out the Genre Muddle. [електронний ресурс]. – Режим доступу: www.gamasutra.com.
3. GameIsArt [електронний ресурс]. – Режим доступу: http://gamesisart.ru/game_dev_create.html
4. Bethke, Erik — Game development and production — Texas: Wordware Publishing, Inc, 2003. — 405 с.
5. Unity3D [електронний ресурс]. – Режим доступу: <https://unity3d.com> (20.02.2018)
6. Unreal Engine [електронний ресурс]. – Режим доступу: <https://www.unrealengine.com> (25.02.2018)
7. Major piracy group warns games may be crack-proof in two years. [електронний ресурс]. – Режим доступу: <https://arstechnica.com>. (18.03.2018)
8. Fullerton, Tracy; Swain, Christopher; Hoffman, Steven — Indie Game Jam: An outlet for Innovation and Experimental Game Design — Morgan Kaufmann, 2008. — 491 с.
9. Бабкин С.А. Интеллектуальная собственность в Интернет. — М.: АО "Центр ЮрИнфоР", 2006. — 512 с
10. wiseGEEK What is an MMORPG [електронний ресурс]. – Режим доступу: <https://www.wisegeek.com/what-is-a-mmorpg.htm> (дата звернення: 07.06.2019)
11. Йоган Гейзинга — Досвід визначення ігрового елемента культури — Київ: Основи, 1994. — 250 с.

12. Tobold What is an MMORPG actually? – [электронный ресурс]. – Режим доступа: <http://tobolds.blogspot.com/2003/07/what-is-mmorpg-actually.html> (дата звернения: 07.06.2019)
13. Luo L— Study on Security Protection of User's Personal Information in Social Networks. *Journal of Library and Information Sciences* — 2012. — № 14. — С. 36-40.
14. Зима В.М., Молдовян А.А., Молдовян Н.А. Защита компьютерных ресурсов от несанкционированных действий пользователей. – Учеб пособие. – СПб: Издательство ВИКА им. А.Ф. Можайского, 1997.
15. Афанасьева К.О. Авторське право: Практичний посібник. – К.: Атака, 2006. – 224 с.
16. Unity Objuscator [электронный ресурс]. – Режим доступа: <http://devxdevelopment.com/> (05.03.2018)