



Софийски университет „Св. Кл. Охридски“

Факултет по математика и информатика

Катедра „Софтуерни технологии“



ДИПЛОМНА РАБОТА

на тема

Изследване на проблемите при
проектирането на ММО видео игри

Дипломант: **Иван Асенов Асенов**

Специалност: **Софтуерни технологии**

Факултетен номер: **25925**

Научен ръководител:
проф. д-р Боян Бончев

София, 2022 г.

Съдържание

1 Увод.....	5
1.1 Представяне на проблема.....	5
1.2 Цели и структура на дипломната работа.....	6
1.3 Очаквани ползи от реализацията на дипломната работа	7
2 Представяне на ММО жанра	8
2.1 Дефиниция на ММО	8
2.2 Виртуален свят	9
2.3 Разлики между ММО и виртуален свят	10
2.4 История на ММО.....	11
2.5 ММО разновидности.....	14
2.6 Примери за ММО видео игри	15
3 Технически проблеми при проектирането на ММО игри	20
3.1 Проблем 1: Обмяна на информация между игровия софтуер.....	20
3.2 Проблем 2: Взаимодействие между играчите	21
3.3 Проблем 3: Технически трудности при внедряването	22
3.4 Проблеми при използването на разпределени системи за създаване на ММО игри.....	23
3.5 Проблеми с физическото изграждане на сървърна архитектура	24
3.6 Възможни решения на проблем 1.....	25
3.7 Възможни решения на проблем 2	27
3.8 Възможни решения на проблем 3	28
3.9 Обобщение на технически проблеми при проектирането на ММО игри	30
4 Социални проблеми при проектирането на ММО видео игри	32
4.1 Проблем 1: Изграждане на игровата среда	32
4.2 Проблем 2: Икономиката в играта.....	34
4.3 Проблем 3: Взаимодействие между нови и развити играчи	36

4.4	Проблем 4: Недоброжелатели.....	37
4.5	Възможни решения на проблем 1.....	39
4.6	Възможни решения на проблем 2	41
4.7	Възможни решения на проблем 3	46
4.8	Възможни решения на проблем 4	48
4.9	Обобщение на социалните проблеми при проектирането на ММО видео игри.....	50
5	Избор на софтуерни технологии за реализация на ММО видео игри	51
5.1	Обобщен поглед върху процеса на разработка на видео игри	51
5.2	Игрови двигател.....	53
5.3	Двигателят Hero (HeroEngine)	54
5.4	Двигателя Unity.....	55
5.5	Двигателя Unreal (UnrealEngine)	58
5.6	Други технологии за реализация на ММО игри.....	59
5.7	Играта Minecraft.....	59
5.7.1	Кратка история на играта	60
5.7.2	Как се играе играта	62
5.7.3	Разрешени проблеми в играта Minecraft	63
5.7.4	Неразрешени проблеми в играта Minecraft.....	64
5.7.5	Задачи по проектиране и експериментално тестване на сървър на играта Minecraft	65
6	Проектиране на решението - игрови сървър на играта Minecraft	
	66	
6.1	Изграждане на Minecraft сървър	66
6.1.1	Стартиране на „Vanilla“ сървър	66
6.1.2	Библиотеката Bukkit (Bukkit API)	67
6.1.3	Стартиране на Paper сървър.....	68
6.1.4	Работа с приставки	69
6.2	Добавяне на важни приставки	70

6.2.1	Приставки за икономика, комуникация и правомощия.....	70
6.2.2	Приставки за допълнителна сигурност и управление	75
6.3	Имплементиране на решения на проблема за изграждане на игровата среда	76
6.4	Имплементиране на решения на проблеми с икономиката	77
6.5	Имплементиране на решения на проблеми с взаимодействието между нови и стари играчи	81
6.6	Имплементиране на решения на проблеми с недоброжелатели ...	83
6.7	Техническа оптимизация на сървъра	85
6.8	Подготовка за начало на експеримента с реални играчи	87
7	Практически експеримент с игрови сървър на играта Minecraft.	
	89	
7.1	Кратък обзор на експеримента. Валидиране на получените резултати	
	89	
7.2	Профил на участниците в експеримента	89
7.3	Анализ на икономическо състояние на сървъра	90
7.4	Анализ на работата на екипа и на поведението на недоброжелателите.....	93
7.5	Анализ на играчите и решенията пряко свързани с тях.....	95
7.6	Обобщение на анализа	97
8	Заключение.....	99
8.1	Обобщение на постигнатите резултати	99
8.2	Насоки за бъдеща работа.....	99
	Използвана литература	101
	Речник на използвани термини и абревиатури	106

1 Увод

1.1 Представяне на проблема

Последните години са едни от най-бурните в света на компютърните игри. Още от самото си начало, тази индустрия заема своето място като важен икономически сектор, който включва всичко свързано с видео игрите – от разработката до разпространението им. С навлизането на света в модерната епоха и развитието на глобалната интернет мрежа, размерите на тази индустрия нарастват многократно. Това се дължи в голяма част и на появата на редица платформи за дигитално разпространение на заглавия, като се започне от платформи за търгуване като *Amazon* и се стигне до специализирани за видео игри пазари - [Steam¹](https://store.steampowered.com/), [Epic Games²](https://www.epicgames.com/store/en-US/), [GoG³](https://www.gog.com/) и още много други.

Благодарение на такива платформи е по-лесно от всякога да се разпростири дадена игра, без да има нужда от допълнителни разходи за изработка на физически копия и дистрибуцията им. Това прави възможно дори самостоятелни разработчици да създават игри и да печелят от тях – нещо, което преди само две десетилетия би било почти невъзможно. Ето защо всяка година излизат все повече и повече нови видео игри като до края на 2020г. само в платформата *Steam* се очаква да излязат около 8500 нови заглавия⁴.

В тази бурна обстановка се забелязва и една тенденция спрямо вида на новите игри. Става въпрос за това, че все по-рядко се появяват игрови заглавия, които са изцяло фокусирани върху игра в самостоятелен режим. Вече се обръща доста повече внимание на режими на игра с други хора, като значително нараства и броят на изцяло фокусираните върху онлайн игра заглавия. Причините за това са доста. Първата е, че днес това е много по-лесно. В зората си онлайн игрите са били ограничени физически до затворени университетски и други локални мрежи и с ниски скорости на връзката. Дори първите модерни онлайн игри, които са имали достъп до интернет, е трябвало сами да изградят и поддържат някаква сървърна инфраструктура. А днес всичко това може да се получи на готово. Понякога това става възможно дори само с натискането на няколко бутона.

Друга причина е, че този тип игри отварят възможност за допълнителни методи за печалба дори след първоначалната продажба. Това включва магазини с предмети и бонуси за играта, абонаменти, козметични промени, допълнително съдържание и дори нашумелите последно време „кутии с плячка“⁵.

Вероятно най-голямата причина обаче е, че груповите игри позволяват на играчите да взаимодействат един с друг, което дава възможност на разработчиците да осигурят стотици часове игрово време не като внимателно изграждат всяка минута от някаква история, а като предоставят среда, в която всяко човешко решение дава практически уникално преживяване.

¹ <https://store.steampowered.com/>

² <https://www.epicgames.com/store/en-US/>

³ <https://www.gog.com/>

⁴ По статистически данни на [SteamSpy \[1\]](#).

⁵ „Кутии с плячка“ (от англ. *Loot Box*) представляват виртуални предмети, които не се използват директно в играта, а се „отварят“, при което играчът получава на случаен принцип предмет или друг вид награда.

Това е особено силно изразено в груповите игри с огромен брой хора (стотици и хиляди), което представлява жанрът на MMO (Massively Multiplayer Online), затова и той е основното лице на тази модерна тенденция.

Многото играчи на едно място обаче води и до редица проблеми. Част от тях са от чисто техническа гледна точка като например как да се обработи взаимодействието на различни играчи с един и същ обект в общата среда без да възниква конфликт. Други проблеми идват от самите играчи, особено в игри с отворен свят, където има огромна свобода на действие. Но с тези проблеми се сблъскват всички - и големи студии за разработка, и многото независими разработчици. Именно затова въпросът за проектирането на MMO видео игри е може би един от най-важните въпроси в игровата индустрия днес.

1.2 Цели и структура на дипломната работа

Целта на тази дипломна работа е да се изследват проблемите при MMO видео игри, като се проектира, тества и внедри игрови сървър на играта *Minecraft*, на който да се проведат експерименти с реални играчи, имащи за цел анализ и валидиране на предложеното решение.

Задачи, произтичащи от целта:

1. Представяне на MMO жанра и сравнителен анализ на проблемите му. Разглеждане на играта *Minecraft*.
2. Преглед на използваните софтуерни технологии за реализация на MMO видео игри.
3. Изграждане на игрови сървър на играта *Minecraft*, което включва:
 - анализ
 - проектиране
 - тестване
 - внедряване
4. Практически експеримент с използване на игровия сървър на играта *Minecraft* с реални играчи.
5. Анализ на резултатите от експеримента.

Работата е структурирана по следния начин. Ще започна с представяне на MMO жанра, като дам по-конкретна дефиниция и я облека с примери. След това ще разгледам проблемите един по един разделени условно на две категории. Ще продължа анализа с представяне на възможни решения на всеки от разгледаните проблеми. След това ще направя преглед на използваните технологии за реализация на MMO видео игри.

На края ще разгледам играта *Minecraft*. Тя е типична MMO видео игра. Има и режим за самостоятелна игра, но това, което я прави популярна, е именно възможността да се играе заедно с други хора. В нея играчите имат свободата да избират как да играят в един на практика безкраен свят. Освен това се модифицира лесно и има голям набор от инструменти, с които могат да се създадат различни среди за играчите. Тези нейни характеристики я правят много удобна за провеждане на експерименти с реални играчи, които имат за цел да валидират някои от предложените решения за преодоляване на проблемите при проектирането на MMO видео игри.

1.3 Очаквани ползи от реализацията на дипломната работа

Качествената видео игра, особено ако е част от ММО жанра, има нужда от сериозно планиране. Решения на възможните проблеми трябва да са част от дизайна на играта, а не да им се обръща внимание по време на самата разработка. Следователно, разработчиците трябва да са предварително запознати с проблемите, които са специфични за ММО. Съдържанието на тази дипломна работа ще помогне в случаите, в които разработчиците нямат предварителен опит с жанра.

Литературата по темата е изключително осъкъдна, а голяма част от съществуващите разработки са писани преди повече от 10 години. Технологичният напредък през това време, особено от гледна точка на интернет (скорост, обем на трафик, сигурност), прави много от препоръките и изводите неподходящи, а част дори напълно безполезни.

Резултатите от тази дипломна работа се очаква да послужат за потвърждение или опровергаване на остатялата литература по темата.

Тезата, особено практическата част, ще бъде и изключително полезна за хората, които искат да проектират собствен сървър в играта *Minecraft*. Ще могат да се запознаят с най-съвременните практики и използван софтуер. Ще научат всичко необходимо за създаването на една стабилна виртуална среда, която вече да могат да надграждат спрямо собствената си визия.

Дипломната работа ще бъде полезна и за хората, който просто искат да се запознаят подробно с ММО жанра. Тази ползва произлиза най-вече от първата поставена задача, която има за цел да представи ММО жанра. Това се случва основно в следващата глава. В нея ще изведа конкретна дефиниция за ММО, ще разгледам подробно важни концепции, ще направя исторически обзор на жанра и ще дам примери.

2 Представяне на ММО жанра

2.1 Дефиниция на ММО

MMOG е абревиатура и идва от *Massively Multiplayer Online Game*, като почти винаги последната дума се пропуска и остава само ММО. Няма общоприет български аналог на термина, но и няма особена нужда от това, защото директният превод води до същата абревиатура. Както много други неща, свързани с интернет културата, трудно е да се даде конкретна дефиниция. Различни научни публикации по теми, засягащи ММО, дават кратки и често субективни дефиниции на термина, а практиката всеки разработчик да самоопределя жанра на играта си води до още по-голямо объркване. За да дефинирам по-точно жанра, ще разгледам описанията в няколко популярни онлайн речника.

1. **[Dictionary.com](#)**: Всяка онлайн видео игра, в която всеки играч взаимодейства с голям брой други играчи. [2]
2. **[Techopedia](#)**: Видео игра, която позволява голям брой играчи да участват едновременно чрез интернет връзка. [3]
3. **[Your Dictionary](#)**: Компютърна игра, играна от много хора. Различава се от стандартна мултплеър игра по това, че средата на действие е вечна. [4]
4. **[Lexico](#)**: Онлайн видео игра, която може да се играе от много голям брой хора едновременно. [5]

Всички дефиниции имат две основни общи части – голям брой хора, които си взаимодействват онлайн. На база тези дефиниции и характеристики на игри, които всеки приема като част от жанра, ще формулирам следната крайна дефиниция за ММО:

ММО е такава онлайн видео игра, в която технологията на играта позволява всеки играч да взаимодейства с огромен брой хора.

Тази дефиниция покрива най-важните моменти от предните – играта се играе онлайн от много хора. Добавих и словосъчетанието „технологията на играта позволява“, защото не трябва крайната популярност или лоша инфраструктура (например сървъри с ниска производителност) на дадена игра да влияят на нейното класифициране. Важното тук е по дизайн играта да позволява много играчи на едно място.

Все пак и с тази формулировка остават въпроси поради факта, че трудно може да се дефинира конкретно число, което да се счита за „огромен брой хора“. В различни специализирани на тема ММО форуми могат да се срещнат всякакви стойности, като за долна граница (която е въсьщност важната за дефиницията) се срещат от 500 до няколко хиляди. Заради всичките неясноти по дефиницията на ММО, често се среща използването на характеристики за ММО черти, за да се определи дали дадена игра спада към този жанр или не. Тези характеристики се срещат в повечето ММО игри, но не са задължителни за да бъде една игра представител на този жанр.

Игрите обикновено се разиграват в някакъв общ виртуален свят като не е необходимо всеки да играе постоянно (тоест света продължава да съществува и да се развива дори когато играчът не е на линия). Това значи, че една такава игра може да продължи месеци дори

години. В зависимост от играта, играчите трябва да работят заедно, да се надпреварват или имат свободата да избират между едното или другото. Характерни са наличието на някакъв тип виртуална икономика и система за комуникация.

На практика обаче могат да се срещнат всякакви характеристики в игри от този жанр. Причината е, че MMO спада към игрови жанрове, но в основата си е по-скоро концепция (както се вижда от дефиницията) и не може да съществува самостоятелно. Това означава, че друг игрови жанр възприема неговата основна характеристика, при което се създава нов под-жанр. Най-популярният пример за това е MMORPG (от RPG – *Role Playing Game*, тоест ролева игра). Този тип игри са най-срещаните примери за MMO, защото в основата на RPG е именно взаимодействието с други хора и с игровия свят. Това е и причината MMO и MMORPG често, но погрешно да се считат за едно и също нещо. Всяка MMORPG е MMO, но не всяко MMO е MMORPG.

Други популярни под-жанрове включват MMO варианти на стратегии в реално време, FPS (*First Person Shooter* – стрелец от първо лице), състезателни и браузър игри. Практически всеки жанр има потенциал да бъде превърнат в MMO, но при някои това е по-трудно поради специфичните конфликти възникващи между природата на MMO и на конкретния жанр, затова се срещат по-рядко.

При почти всички MMO игри обаче взаимодействието става в някакъв виртуален свят. Сега ще се спра по-подробно на него и ще се опитам да дам отговор защо това е така.

2.2 Виртуален свят

Както с MMO, и концепцията „виртуален свят“ дълго време няма точна дефиниция, въпреки че се използва още от края на миналия век. Това се променя със статия публикувана в *Journal of Virtual Worlds Research* през 2008. Статията, озаглавена „Към дефиниция за Виртуален Свят“, има за цел именно представяне на точна дефиниция на термина. Обединявайки три популярни до този момент дефиниции, авторът достигат до следната нова формулировка:

Синхронизирана и постояннона мрежа от хора, изобразени чрез аватари, поддържана чрез система от компютри. [6]

За да няма грешни тълкувания на представената дефиниция, авторът допълнително изясняват значението. Общите действия на участниците във виртуалния свят трябва да са синхронизирани – всеки да може да вижда действията на друг. Само така ще са възможни общи, масови групови дейности. Допълнително според статията, синхронна дава възможност да се изграждат концепции като близко и далеч както и някакво усещане за география или терен.

Света също така трябва да бъде постоянен, което означава, че ще продължи да съществува дори след напускане на участниците в него. Той не трябва да може да се поставя на пауза. По този начин никой участник не е автоматично център на виртуалния свят, а всеки е част от някакво динамично общество, което би съществувало независимо от негово участие в него. Освен това, постоянството изисква наличието на определени правила (като физичните закони в реалния свят), които трябва да са еднакви за всички.

Участниците пък трябва да могат да взаимодействат един с друг било то пряко или посредством света, като това взаимодействие се случва чрез аватари⁶, а това да се случва чрез някаква система от компютри. Последно уточнение е направено за да се изключат игри, които са на живо (например популярната игра „Дракони и Тъмници“, която реално отговаря на всички останали точки в дефиницията, но е вид настолна игра).

В същия брой на *Journal of Virtual Worlds Research* излиза и друга статия „Дефиниране на Виртуални светове и Виртуални среди, в която независимо от W. Bell, се подчертава необходимостта от дефиниция и се дава сравнително близка до предходната:

Постоянна виртуална среда, в която хора възприемат присъствието на други хора и където могат да си взаимодействат. [7]

Въпреки тези две формулировки обаче следващите години в различните публикации отново започват да се появяват разминаващи се дефиниции (и като цяло различно използване на термина „виртуален свят“). Има противоречия дори в различни статии на едни и същи автори. Проблемът допълнително се усложнява с появата на нови термини като „виртуална реалност“, „симилирана реалност“ както и подтипове на виртуален свят като „сериозен виртуален свят“, „потапящ виртуален свят“ (от *immersive* - потапяне), „социален виртуален свят“ и други. Стига се до там, че статии от последните години относно термина приличат по-скоро на статистически обзор, отколкото на логически изградена дефиниция. Затова ще спра до тук и ще използвам дефиницията на Bell за виртуален свят в следващите глави, защото тя най-добре описва общоприетото от най-много хора разбиране.

2.3 Разлики между ММО и виртуален свят

На пръв поглед е лесно ММО и „виртуален свят“ да се приемат за едно и също нещо, защото и при двете имаме онлайн взаимодействие между играчи. Разликите обаче стават лесно очевидни ако се направи по-дълбок сравнителен анализ. Най-основната е броят на играчите. При ММО има дефиниран „голям брой хора“. Не обръщайки внимание на трудностите при определяне на точно число за „голям брой“, факт е, че във виртуален свят такова изискване няма. Според дефиницията и петима могат да са мрежа от хора и стига да са спазени останалите критерии, те ще бъдат част от виртуален свят. Освен това очевидно условие за ММО е да бъде игра. Ако липсват очевидни игрови елементи (цели, задачи, правила, надпревара), то няма как даден виртуален свят да се счита за игра.

Следователно не всеки виртуален свят е ММО. Но възможно ли е обратното – всяко ММО да е виртуален свят? Отговорът е не, и това е така заради необходимостта на виртуалния свят да бъде постоянен. Той трябва да може да съществува и без хора в него. Такава необходимост при ММО няма – средата, в която са взаимодействвали играчите, може да спре да съществува (например в края на играта).

Но каква е причината тези две концепции да са толкова свързани една с друга? Ще припомня, че в дефиницията за ММО има две ключови части – **взаимодейства и огромен брой хора**. Те са в основата на проблемите при проектиране на ММО игри, които условно

⁶ Аватор е всяко дигитално представяне на реална личност, което притежава свойството да извършва действия и се контролира от човек в реално време. Аватора буквально функционира като кукла, управлявана от играта. [6]

се разделят на технически (поради големия брой хора и техническото им взаимодействие) и социални (поради големия брой хора и социалното им взаимодействие). Проблемите на коя от тези две групи ще са по-трудни за преодоляване зависи силно от останалите характеристики на играта. При MMORPG например по-голямо внимание трябва да се обърне на социалния аспект, а при MMOFPS е много по-трудна техническата част.

Независимо от конкретните характеристики на играта, в повечето случаи първото нещо, с което се сблъскват разработчиците при проектиране на MMO игри е именно проблемът с огромния брой хора. От въпроси как да се синхронизират играчите за да няма проблеми с възприятията на всеки играч (един и същи елемент да се вижда по еднакъв начин от всички) до това как ресурсно да се поема и обработва големият интернет трафик. Но най-вече стои въпросът как логически да си взаимодействват всички едновременно. Ето защо виртуалният свят е почти винаги в основата на MMO игрите. Самата му концепция гарантира взаимодействието и освен това тя предоставя редица инструменти, с които разработчиците да се справят с техническите проблеми при изграждането на MMO.

2.4 История на MMO

Въпреки, че не са напълно еднакви, MMO и виртуален свят притежават много общи неща, като едно от тях е тяхната история. Началото на тази история поставя появата на виртуален свят. Тази концепция се развива независимо поне 6 пъти, но повечето от тези светове не оставят трайна следа върху бъдещото и развитие.^[8] Затова ще обърна внимание само на две. За хронологично начало се смята разработената през 1973-74 година игра *Maze* (от англ. Лабиринт), чието действие се развива именно във виртуален свят ^[9]. В тази игра, целта на играчите е да си проправят път през лабиринт и да стрелят по останалите играчи, с които се срещнат. Налице са всички точки от дефиницията за виртуален свят - има постоянна мрежа от хора, които са изобразени чрез аватари, но играта не е MMO, защото дизайна и не позволява взаимодействието на голям брой хора. Въпреки това е доста близко за времето си особено когато играта е подобрена да може да се играе през APRAnet⁷.

За родоначалник на MMO жанра обаче се смята друга игра, която отново е и виртуален свят. Това е играта MUD (*Multi User Dungeon*, Тъмница за много потребители).

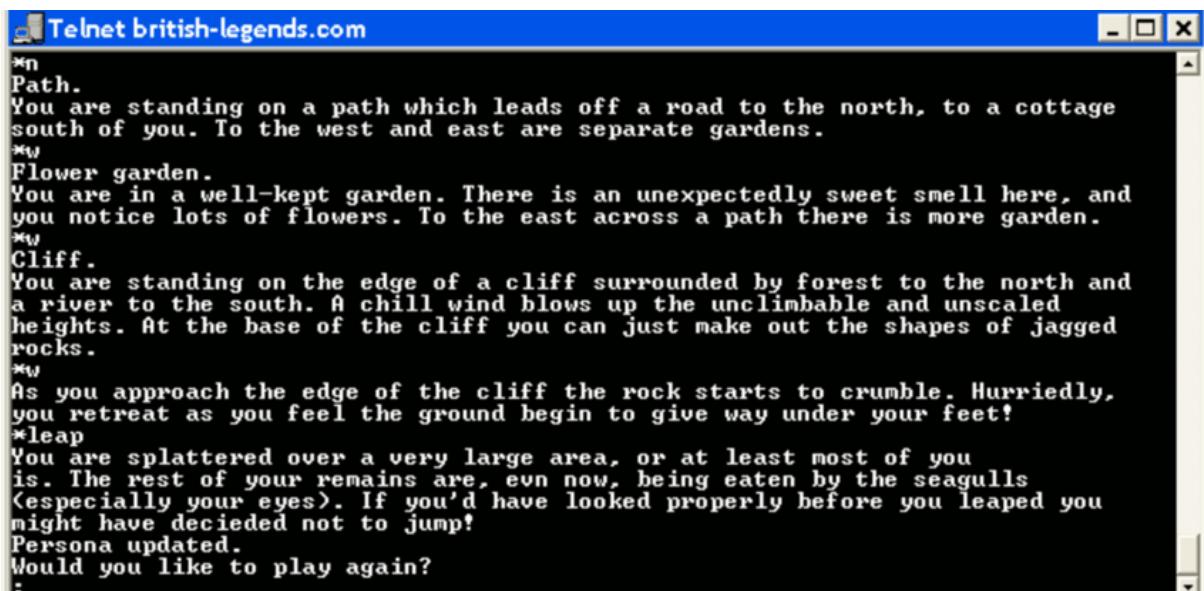
Разработена е през есента на 1978 година в университета Есекс от Рой Т. и Ричърд Бартъл и е изцяло текстова. Това значи, че всичко, което играчите виждат, чуват, усещат и правят е представено визуално с думи.^[8] Това разбира се не противоречи на дефиницията за виртуален свят, защото там няма изискване за графика. Най-важното в тази игра и причина да се смята за основоположник е факта, че тя е с отворен край. Играчите имат свобода, а светът няма предварително начертан път и край. Разбира се играта не може да се счита за



Фигура 1 - Част от лабиринта в играта *Maze* както се вижда от играчите. Изображението е от видео в YouTube на Bruce Damer качено през 2011, снимано 2004.

⁷ APRAnet е предшественик на Интернет и е първата в света компютърна мрежа, която използва протокола за комуникация TCP/IP. [\[10\]](#)

представител на ММО, защото липсва условието за масовост, но историята на целия жанр може да се проследи директно до MUD. Започват да се появяват много вдъхновени от нея игри като *Shades* (Newell, 1985), *Gods* (Laurie, 1985), *AMP* (Blandford, 1985), *MirrorWorld* (Cordrey, 1985) и други. Самата игра минава през няколко итерации през следващото десетилетие. В края на 80-те играта е пренаписана за последен път и получава обозначението MUD2⁸. Въпреки че са налични и финансови успехи, все още игрите си остават лукс поради високата цена на телефонните обаждания тогава и се играят предимно в университетски мрежи. [8]



```
Path.
You are standing on a path which leads off a road to the north, to a cottage
south of you. To the west and east are separate gardens.
*W
Flower garden.
You are in a well-kept garden. There is an unexpectedly sweet smell here, and
you notice lots of flowers. To the east across a path there is more garden.
*W
Cliff.
You are standing on the edge of a cliff surrounded by forest to the north and
a river to the south. A chill wind blows up the unclimbable and unscaled
heights. At the base of the cliff you can just make out the shapes of jagged
rocks.
*W
As you approach the edge of the cliff the rock starts to crumble. Hurriedly,
you retreat as you feel the ground begin to give way under your feet!
*leap
You are splattered over a very large area, or at least most of you
is. The rest of your remains are, even now, being eaten by the seagulls
(especially your eyes). If you'd have looked properly before you leaped you
might have decided not to jump!
Persona updated.
Would you like to play again?
```

Фигура 2 - Снимка на момент в играта MUD1 (Wilks, неизвестна дата)

Затова за да видим първата ММО игра трябва да погледнем след средата на 90-те, когато Интернет навлиза в живота на огромна част от западния свят. По това време има няколко заглавия, които се надпреварват с ясната цел да са първите в един нов тип игри, които да бъдат играны от истински масивен брой хора. Водейки се от същата логика като с първите виртуални светове, от значение са две заглавия.

Meridian 59 е играта, която хронологически излиза първа, през 1996 година. Тя представлява фентъзи игра от първо лице. Света е изграден от три града, околностите им и подземен свят, в който играчите попадат, ако героят им умре. Дизайнът на играта позволява множество инстанции, които могат да са дори на отделни сървъри, да работят паралелно, като всяка се използва от няколко стотин души едновременно. По този начин хиляди могат да играят играта едновременно. [11]

Игра е много обещаваща, но не претърпява голям успех. Сред споменаваните за това причини са недостатъчно съдържание, сравнително слаба графика спрямо популярни за времето игри в самостоятелен режим, както и липса на сериозна реклама от разработчика.[8]

⁸ Оригиналната версия на играта вдъхновява много нови игри и MUD започва да се използва като общо наименование за всички тях, затова я преименуват на MUD1. [8]

До юни 1999 година са продадени едва 14,359 копия⁹, което определено е достатъчно, за да е представител на MMO, но показва и липсата на успех.

Meridian 59



The 3DO Company takes a stab at creating the ultimate multiplayer online experience. Can it succeed?



Meridian 59 is truly a self-contained world. It has its hazards (above), its own economy (shop keeper, right), and even its own version of an online newsgroup (*The Obsidian Globe*, far right)

Фигура 3 - Част от статията за *Meridian 59*. Горният текст гласи „Компанията 3DO прави опит да създаде най-върховното преживяване онлайн. Може ли да успее?“. Долният текст гласи „*Meridian 59* е напълно самостоятелен свят. Има опасности (снимката от ляво), собствена икономика (снимката в центъра, изобразен е търговец) и дори собствен онлайн новинарски център (снимката от дясно, изобразена е Обсидиановата сфера)“

Така се стига до *Ultima Online*, излязла през 1997 година. Действието на играта се развива във вселената на играта *Ultima* (Garriott, 1980) и е създадена да носи възможно най-доброто преживяване във виртуален свят. Този свят има собствена икономика, екологична система, очаква се да бъде населена от хиляди и винаги да е достъпен. Освен това е игра – със собствена история, сюжет и игрови задачи, които са навързани и принуждават играчите да си участват в тях заедно за да се стигне до развръзка (като е възможно играчите да трябва и да си помогнат, и да са изправени един срещу друг).

И е характерно голямото ниво на свобода. Всеки може да стане известен като герой, който е доблестен и защитава слабите или като бандит, който граби и унищожава. [13]

За по-малко от година играта отчита 100,000 играчи. Така тя вече е MMO не само по дефиниция, но и на практика. За съжаление става жертва на собствения си успех. Огромният брой хора предизвиква редица технически проблеми, които пречат на бъдещото и развитие. Въпреки това години по-късно в периода 2001-2003 играта все още държи около 250,000 потребители и е активна и до днес със завидно количество играчи. [8]

От тогава до днес не спират да излизат MMO игри, като се чупят рекорди отново и отново. Модерните технологии позволяват все по-мащабно MMO изживяване, но дали сега се намираме в златния век на жанра или още в началото на неговата история тепърва ще се разбере.

⁹ Тази информация идва от уебсайта *GameDaily*, който пък цитира PC Data – американска асоциация за пазарни статистики. И двете вече не съществуват, но могат да се намерят архивирани страници на статии. [12]



Фигура 4 – Екранна снимка от играта *Ultima Online* в съвременния и вариант. Изображението е озаглавено "Победител M" от пролетно състезание през 2016 [14]

2.5 MMO разновидности

В началото на тази глава споменах, че всяка MMO игра на практика е представител и на някой друг жанр. Тук вече става дума за класификация, която е основана на конкретни игрови характеристики. В зависимост коя характеристика се използва, има различни популярни категории. Класификацията по жанр може да се прави спрямо игровата обстановка (устърн, фантастика, фентъзи), спрямо игровата платформа (конзолни игри, PC, браузър), спрямо историята (екшън, приключенска) и дори спрямо начина, по който играчите виждат и взаимодействват с играта (от първо лице, от трето лице, стрелба, състезателна).

На теория почти всеки жанр може да се обедини с MMO, ако е спазена дефиницията. Някои дори са особено съвместими, защото основните им елементи работят много по-добре в обстановка с много играчи (докато при други е много по-трудно, защото важи обратното). Такива жанрове толкова добре обединяват MMO концепцията със своята основна характеристика, че на практика създават нов под-жанр. Сега ще направя кратък обзор на най-популярните MMO под-жанрове.

MMORPG (от RPG – *Role Playing Game*, ролева игра) жанрът комбинира характеристики на ролевия жанр с тези на ММО. В този жанр, всеки играч поема контрол над измислен персонаж, който е потопен в някакъв добре изграден, постоянен виртуален свят. Основна причина светът да изглежда „жив“ е именно огромния брой хора. Характерни са системи за комуникация, за взаимодействие (като играчите могат да си помагат, но могат и да си пречат), система за развитие на героя, изследване на игровата среда (която често е с големи размери) и наличието на история, в която играчите често имат високо ниво на свобода на действията. Повечето игрови елементи са така проектирани, че да стимулират игра с други хора. Почти винаги потребителите се обединяват в някаква социална структура като съюз, гилдия или клан, които възникват дори без да са заложени в играта. Голям елемент са и взаимоотношенията между големи групи от хора, като често те водят до масови социални събития като битки или войни, състезания и други.

MMORTS (от RTS – *Real-Time Strategy*, стратегия в реално време) обединява характеристиките на жанра „стратегия в реално време“ с тези на ММО. В стратегическите игри, играчите обикновено поемат контрол над множество единици, сгради, градове дори държави. Действието се развива върху някакъв предварително дефиниран регион и е характерно събиране на ресурси, строене на бази и трениране на войски, като всичко се управлява директно и в реално време. На игровата карта се намират други подобни на управляваните от играча структури, с които той може да се съюзява или да влиза в конфликт. MMORTS жанрът е на практика абсолютно същия, но се отличава с огромен брой играчи на една и съща карта, състоянието на която не спира да се променя, дори когато играчите не са на линия (например ресурси продължават да се генерираят, единици продължават да се движат и т.н.). Докато в повечето стратегически игри, враговете са най-много дузина, то тук могат да са хиляди.

Много близък до MMORTS жанра са **браузър ММО** игрите. Представителите им са всички игри, които могат да се玩а изцяло чрез браузъра, и в които играчите могат да взаимодействат с огромен брой хора. Много често тези игри имат всички характеристики на стратегия, затова двата жанра се бъркат. Срещат се обаче и други браузър ММО-та, които имат само елементи от ролевият жанр например, затова не е правилно да се смята, че браузър ММО и MMORTS са едно и също нещо. Ключовото в този жанр е, че от техническа гледна точка е много по-лесно да се изгради масово населен свят, защото те са на практика сайтове. Играчите извършват действия чрез натискате на бутони, а решенията се взимат на база предоставена статистическа информация за света и управляваните от потребителя елементи.

Срещат се и други под-жанрове като **MMOFPS** (от FPS – *First Person Shooter*, стрелец от първо лице), **MMOTBS** (от TBS – *Turn-based Strategy*, походова стратегия), **Casual MMO** и други. Няма да се спират подробно на тях, защото за разлика от MMORPG например, където огромния брой хора добавя нови игрови елементи, то те са абсолютно същите като не-масовия си вариант, просто със силно увеличен брой на участниците. Затова за да завърши представянето на ММО жанра, ще премина към конкретни примери.

2.6 Примери за ММО видео игри

Някои от най-известните по света игри са представители на ММО. Може би най-споменаваната е *World of Warcraft* – WoW¹⁰ (играта е представител и на MMORPG жанра).

¹⁰ <https://worldofwarcraft.com/en-us/>

В нея всеки играч може да създава и развива герои, с които взаимодейства с останалите във виртуален онлайн свят. С близо четири милиона активни играчи към днешна дата¹¹, играта определено отговаря на условието за огромен брой, въпреки факта, че е необходим месечен абонамент. Играта е и виртуален свят. Поради голямата си популярност и роля на лице на жанра, много от нейните характеристики се срещат и в много други ММО игри. Но следващият пример добре илюстрира как само наличието на такива елементи не е достатъчно условие дадена игра да е ММО.

*League of Legends*¹² е безплатна фентъзи игра, която на пръв поглед наподобява WoW. И при нея всеки играч поема контрол над герой, наречен шампион, с който се бие и който развива. Играта обаче не е представител на ММО жанра, защото не отговаря на изискването всеки да може да взаимодейства с огромен брой хора. Това е така, защото самата игра се свежда до 30-40 минутни мачове, в които всеки шампион започва от ниво нула независимо от предишни мачове. Въпреки, че самата игра има над сто милиона активни играчи¹³ и определено отговаря на идеята за масовост, тези играчи не могат да си взаимодействват извън обхвата на мачовете, които имат не повече от десет души едновременно. Следователно играта е по-скоро представител на MO – Онлайн Мултиплейър.

За да избегна стереотипа с ролевия жанр, ще разгледам играта *Imperia Online*¹⁴. Разработена от българска компания през 2005 [17], тя все още има огромно присъствие на сцената на браузър стратегиите и до днес. Действието на играта се развива в епоха наподобяваща средновековието, като всеки играч поема контрол над своя собствена империя. Въпреки че една част от играта представлява самостоятелно развитие на провинции в империята, същинската игра представлява именно взаимодействието с останалите владетели в игровия свят. Дизайнът на играта позволява хиляди империи да съществуват едновременно в един сървър следователно играта отговаря на изискванията и е представител на ММО жанра.

Не е задължително една игра да има само един режим. Например игрите от серията *Age of Empires*¹⁵. Те са част от жанра на стратегии в реално време и в тях играчът поема контрол над цивилизация, с която строи, тренира войски и води битки. При *Age* игрите еднакво са на фокус и режима за самостоятелна игра, и мултиплейъра. Налице са както внимателно изградени кампании с история и задачи, така и система за генериране на терени, в които играчите да се изправят срещу други. Играйте обаче не са представител на ММО жанра, защото не отговарят на условието за масово взаимодействие – всяка игра в мрежа поддържа не повече от дузина играчи. Въсъщност въпреки първоначалното ми твърдение почти не съществуват игри, които да са едновременно ММО и да имат самостоятелна кампания.

Едно от малкото изключения е играта *Age of Conan*¹⁶. Тя е типичен представител на MMORPG жанра и много наподобява WoW по характеристики. Отново имаме сложна система за създаване на герой (аватар), който вдига нива, събира предмети и екипировка и взаимодейства с огромен брой хора. Интересното и различното тук е възможността да се мине в режим на самостоятелна игра, в който всеки може да си развие героя до максимално ниво в локално копие на играта. Така развитият герой след това може да се върне при останалите и отново да е част от ММО света на *Age of Conan*.

¹¹ По изчисления на [MMO Populations](#) [15]

¹² <https://na.leagueoflegends.com/en-us/>

¹³ По изчисления на [LeagueFeed](#) [16]

¹⁴ <https://www.imperiaonline.org/>

¹⁵ <https://www.ageofempires.com/>

¹⁶ <https://www.ageofconan.com/>

Не трябва да се забравят и модификациите, които обществото от играчи на всяка игра правят. Особено последното десетилетие, практиката да се добавя или променя съдържание на игри не от техните разработчици, а от индивидуални лица прави възможно една и съща игра да се изживее отново и отново по съвсем различен от първоначалния начин. Това включва и създаването на MMO режим в игри, които първоначално са били само за самостоятелна игра или нормален мултиплейър.

Пример за това е играта *Freelancer*¹⁷ разпространявана от Microsoft. Тя представлява космически симулатор, който включва елементи на търгуване и космически битки в пилотирани от играчите кораби. Мрежовата игра дори е представител на виртуален свят, но не и на MMO поради ограничения брой играчи. Това обаче се променя с модификацията *Freelancer Discovery*¹⁸. Тя отваря възможността голям брой играчи да съществуват едновременно, като могат да строят бази, да се бият, да търгуват и така нататък. Хората зад проекта дори поддържат официален сървър, но всеки може да си създаде частен такъв.



Фигура 5 - Масова битка в официалния сървър на модификацията *Freelancer Discovery*. Изображението е от сайта на проекта. [\[18\]](#)

¹⁷ https://www.eurogamer.net/articles/r_freelancer_pc

¹⁸ <https://discoverycgc.com/>

Друг пример за модификация е *Skyrim Together*¹⁹. Тя е модификация на игра от поредицата *Elder Scrolls*, в която е пета поред – *Skyrim*. Тази игра е една от най-представителните в жанра на ролевите игри и притежава редица от елементите срещани и в много MMO игри.



Фигура 6 - Въпреки че светът в играта *Skyrim* е огромен, той не е представител на виртуален свят, защото играта е в самостоятелен режим и светът спира да съществува, когато играчът го напусне.

Изображението е от ревю на играта в *The Telegraph* (Sam White, 2016)

Но има единствено режим за самостоятелна игра. Модификацията *Skyrim Together* добавя възможността за мрежова игра, но това е предназначено за малки групи. Следователно играта не е представител на MMO.

Последния пример, който ще разгледам, е *Second Life*²⁰. Той представлява виртуален свят, в който всеки потребител приема ролята на жител и така взаимодейства с огромен брой други хора. Това взаимодействие включва комуникиране, участие в групови дейности, търговия с виртуални предмети и услуги и т.н. *Second Life* определено е представител на виртуален свят – средата е постоянна, взаимодействието става мрежово и чрез аватари, но не е MMO, просто защото не е игра. Няма поставени цели, няма задачи или история. Никакви ограничения освен правилата на виртуалния свят.

В следващата таблица ще обобщя всички примери. В първата колона е заглавието на разглеждания пример. Втората и третата са необходимите за да бъде MMO игра критерии, а в последната дали е представител на този жанр. Критериите са само два, защото дефиницията на MMO изискава само две неща:

Критерий 1 – Разглежданият пример игра ли е?

Критерий 2 – В разглеждания пример потребителите взаимодействват ли с огромен брой хора?

¹⁹ <https://skyrim-together.com/>

²⁰ <https://secondlife.com/>

Заглавие	Критерий 1	Критерий 2	ММО ли е?
World of Warcraft	Да	Да	Да
League of Legends	Да	Не	Не
Imperia Online	Да	Да	Да
Age of Empires	Да	Не	Не
Age of Conan	Да	Да, но само в онлайн режим	Да
Freelancer	Да	Не	Не
Freelancer Discovery	Да	Да	Да
Skyrim	Да	Не	Не
Skyrim Together	Да	Не	Не
Second Life	Не	Да	Не

Фигура 7 - Сравнителна таблица по критерии.

Критерий 1. Заглавието игра ли е?

Критерий 2. Има ли взаимодействие с огромен брой хора?

Така дефиниран, ММО жанрът изглежда тривиален за проектиране – само две изисквания. Трябва обаче да се прави разлика между това една игра просто да отговаря на дефиницията и това да бъде качествен представител на жанра. За да бъде успешна една ММО игра, нейните разработчици трябва да се справят с редица проблеми. Ако създадат своята игра без да направят това, то тя може би ще бъде представител на жанра, но няма да бъде интересна за играчите, следователно дори технологията и да позволява взаимодействието на огромен брой хора, на практика няма да има такова взаимодействие, просто защото никой не играе.

Все пак, преди да се обърне внимание на качеството, си остава основната задача как технически да се случва онлайн взаимодействието. В следващата глава ще разгледам подробно този въпрос, като представя основните технически проблеми при проектирането на ММО игри и предложа възможни решения.

3 Технически проблеми при проектирането на ММО игри

Проектирането на ММО игри изправя разработчиците пред най-много предизвикателства от вероятно всички останали жанрове. Това е така, защото освен проблемите, идващи от големия брой хора, остават и всички трудности свързани с разработването на самата игра. Още повече, тук са на лице толкова много елементи, че и най-малкото решение за някой от тях може да повлияе по невъобразими начини на цялата игра. Затова правилното и изключително детайлно планиране на цялата софтуерна архитектура е първата задължителна стъпка при разработката на ММО. И още в началото на планирането възниква първият основен проблем – въпросът за обмяна на информация между игровия софтуер на всеки играч.

3.1 Проблем 1: Обмяна на информация между игровия софтуер

Този проблем присъства при всяка ММО игра, защото по дефиниция взаимодействието се случва онлайн, следователно всеки играе играта от някаква собствена инстанция на игровия софтуер. Веднага става ясно, че ММО игрите са представител на някаква разпределена система. Разпределена система е такава, в която всеки компонент в мрежова система от компютри комуникира и координира своите действия само чрез обмяна на съобщения. [\[19\]](#)

Природата на ММО прави тази задача още по-трудна дори в контекста на съвременните комуникационни мрежи. Причината за това са няколко специфични изисквания относно тази обмяна на информация. На първо място е необходимостта това да става за възможно най-кратко време. Високата скорост е важна от една страна играчите да получат гладко изживяване в играта, а от друга да се намали максимално възможността от грешно представяне на случващото се поради забавяне на информацията за конкретно събитие.

Второ изискване е да няма загуби по време на обмяната на важни съобщения. Това е очевидно, защото изживяването на играчите трябва да бъде едно и също. Загубата на дори малка част от информацията за ключови действия на други означава, че всеки ще възприема едно и също нещо по различен начин, а това е недопустимо. Разбира се има случаи, в които не е толкова важно да се получат абсолютно всички пакети с информация като например при действия, които се извършват няколко пъти в секунда.

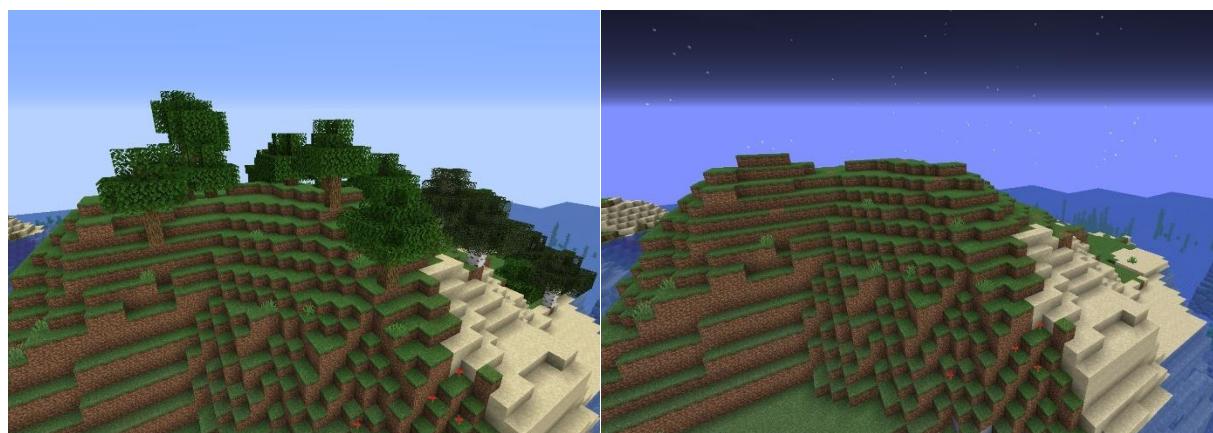
Трето изискване е сигурност на връзката. Съдържанието на съобщенията трябва да бъде защитено и да не може да се чете или променя. В противен случай, този обмен може да стане обект на атаки с цел научаване на информация в по-ранен момент от колкото трябва или промяната и. Такива слабости могат да се използват за разработката на зловреден софтуер – копие на оригинална, с който някои играчи да имат нечестно предимство над останалите или дори напълно да поемат контрол над игровата среда и да я манипулират в тяхна полза.

И на края от дефиницията за ММО идва последното изискване, което ще спомена - необходимо е всеки играч да има достъп до информация за действията на всички останали играчи. На пръв поглед това е най-голямата пречка именно поради огромния брой хора. Ако по никакъв начин, дали пряко или непряко, не може да се усети ефект от действията на друг играч, то взаимодействие реално няма. Ненапразно обаче използвах и думата „непряко“. Това е така, защото не е нужно даден потребител да знае за действията или състоянието на всеки останал във всеки един момент. Достатъчно е да има възможност тази информация да се получи, ако има нужда от нея.

3.2 Проблем 2: Взаимодействие между играчите

Краят на миналия проблем е началото на следващия. А именно въпросът какво точно представлява взаимодействието между играчите. То се разделя на два вида. На първо място е прякото взаимодействие. Такова има, когато двама или повече играчи се намират в една и съща ситуация в контекста на играта и всеки получава информация за действията на абсолютно всеки от останалите. Основен проблем тук е как да се синхронизират действията на всички играчи за да се получи ясна поредица от събития. Най-важното е полученият низ от събития да е един и същ за всеки от участниците в тази локална среда. Ако това не е така, възникват редица усложнения. От простия въпрос кое е реалното състояние на дадената ситуация до възможността за пълен срив в играта.

Второто взаимодействие е непрякото. Това е нещото, което обединява всички играчи и е в основата на повечето ММО игри. Основният проблем при него е въпросът как да се съхранят текущото състояние на цялата игровая среда, за да може даден играч да получи информация за събития, на които не е присъствал. Сложността на тази задача зависи от относителния размер на игровата среда и нивото и на абстрактност. По-детайлните игри имат нужда да следят повече обекти, с които играчите могат да взаимодействат.



Фигура 8 - Дървесината е важен консуматив в играта Minecraft и се получава като се секат дървета. С отсичането на цялата гора на снимката от ляво, даден играч принуждава всеки следващ да търси друга гора. Въпреки че тези играчи може никога да не се срещнат, те имат взаимна връзка помежду си, която регулира техните действия. Възможно е дори да не разберат за това.

Сложен момент представляват и случаите, в които двете концепции се преплитат. Например когато играчи едновременно попаднат в локална среда, за която нямат информация, те трябва да я получат по едно и също време и по един и същ начин. В противен случай има рисък един от играчите да извърши действие в отговор на ситуация, за

която игровият софтуер на останалите не знае, и така да се получат две паралелни състояния на играта, които си противоречат.

3.3 Проблем 3: Технически трудности при внедряването

Неслучайно набледнах на това колко е важно да се планира цялата игра преди да се започне с каквото и да е писане на код. Това включва и взимането под внимание на обстоятелства, които реално ще настъпят чак в края на процеса, по време на внедряване на играта. Става дума за въпроси от типа на как е разпределена географски целевата аудитория и какъв е приблизителния брой на реално очакваните играчи, което включва не само някакво число, но и каква е очакваната активност през денонощето например. Тези и други обстоятелства трябва да се преценят добре, защото от тях може да зависи какви решения ще се използват и по другите въпроси.

От значение са две по-конкретни стойности. На първо място е пълния брой играчи, които ще играят играта през цялото и съществуване. За да има непряко взаимодействие между хората в дадено ММО, то е необходимо прогресът на всеки играч да се съхранява някъде и колкото повече потребители има, толкова повече памет трябва да се задели за тази цел. Ще отбележа, че количеството данни свързани с всеки отделен играч може да бъде доста голямо, защото тук не става дума за обикновен сайт. Този проблем става още по-сериозен, ако се съхранява и информация за цели обекти (ако играта позволява потребителите да създават собствени аватари например) или се съхранява състоянието на цялата игрова среда.

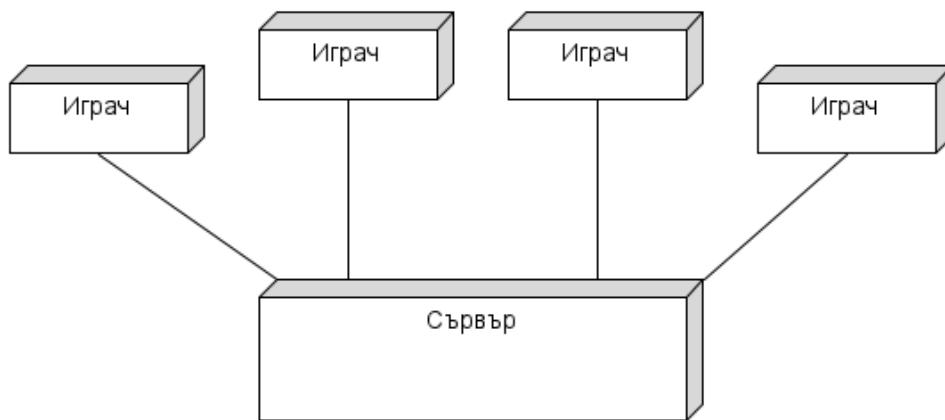
Другата стойност, която е още по-важна, е очакваният брой активни едновременно играчи. Игри по начало са представител на софтуер с високи технически изисквания. Това е така, защото е необходимо те да се изпълняват гладко, без проблеми, без да се забавят и като цяло без да се нарушава по никакъв начин възприятието на играча. По-голям брой играчи естествено водят до нуждата от още по-големи изисквания и са необходими компютри и сървъри по-добра производителност. Колко точно по-добри и кои параметри по-конкретно, зависи силно от вида на играта. Най-вече броят хора влияе на изискванията за изчислителната сила (брой на ядра в процесорите и тяхната тактова честота), оперативна памет (размер и поколение), вътрешна памет за съхранение (HDD или SSD), графична памет (размер и поколение) и скорост на връзката.

Лесно е това да се види. Например в игра с компютърно управлявани същества (чудовища, животни, персонажи) повече активни играчи означава повече същества, особено ако играчите са отделени един от друг. Това би се отразило най-вече на изчислителната сила и на оперативната памет заради допълнителните операции, които са необходими при управлението на изкуствения интелект на всичките същества. Друг пример би бил игра, в която се взаимодейства силно с игровия свят. В този случай повече играчи означава повече промени върху игровата среда едновременно. Това може да се отрази на вътрешната памет, която ще се забави от огромния брой I/O операции.

3.4 Проблеми при използването на разпределени системи за създаване на ММО игри

Преди да започна с анализ на първия проблем, ще разгледам по-подробно разпределените системи. Могат да се използват всякакви архитектурни модели, но най-често те спадат към една от две основни категории:

1. Клиент-Сървър разпределена система (от англ. *Client-Server model*): Това е най-споменаваната, когато се говори за разпределени системи и се счита за най-използваната днес. [19] Такава структура разпределя задачите между сървъри, които предоставят някакъв ресурс и клиенти, които правят заявки за тези ресурси. Това може да се случва и на повече от едно ниво и даден участник да бъде сървър за един, но клиент за друг. Типичен пример е система с база данни. Мястото, на което се съхранява информацията (може и да се обработва) е сървъра. А клиентите са тези, които поискват или създават тази информация. Идеята зад тази архитектура е да има възможност различни потребители да работят с едни и същи данни. [20] Веднага се вижда, че това би било изключително полезно за едно ММО. Тази разпределена система си има и своите минуси, които са силно изявени при игри от този жанр. Големият брой хора означава огромен трафик, който трябва да се поеме и обработи от сървъра. Ако той няма необходимия ресурсен капацитет за това, то цялата система ще започне да се забавя (в най-добрия случай), което е пагубно за една онлайн игра. Скалирането в общия случай е трудно и изиска много точна оценка на очаквания брой играчи, което е много трудна задача.



Фигура 9 - Най-често срецираната архитектура в ММО игрите (Клиент-Сървър). На сървъра се намира игрови софтуер, който най-често има за основна цел да управлява взаимодействията между играчите и да съхранява текущото състояние на игровата среда. При играчите се изпълнява друга форма на игровия софтуер, който има за цел да достъпи информацията предоставена от сървъра, да я визуализира и да предаде обратно към сървъра реакциите на играта спрямо тази визуализация.

2. *Peer to peer* разпределена система: Терминът няма общоприет български аналог, среща се в преведена литература като *p2p*. Идеята зад този тип архитектура се съдържа в името (*Peer* – равен, равнопоставен). В него всички участници са равни и не могат да се разделят на клиенти и сървъри – участват еднакво в изпълнението на

процесите. За разлика от предната система, където най-често се изпълнява различен софтуер в зависимост от ролята, то тук на практика при всеки участник работи една и съща програма. [19] Този тип разпределена система разрешава един от основните проблеми на предходната. Поради факта, че всеки участник играе ролята и на сървър, то скалирането е изключително лесно и няма значение какъв брой играчи ще има. Освен това лесно се постига висока скорост на връзката между тях, защото са свързани директно. Въпреки това почти няма MMO игри, които работят на този принцип. Една от основните причини за това е сигурността. Играта с модифициран софтуер е много по-лесна в една *p2p* система, защото липсват много от стандартните методи за контрол, които са налице в една централизирана система. Това води до лесна манипулация от страна на недоброжелатели. Освен това много по-трудно е да се гарантира последователност в игровата среда, често се губи информация и лесно възникват конфликти между участниците. Има интерес към използването на този тип архитектура в MMO игри и се разработват техники и методи за преодоляване на проблемите като например система за репутация, посредници, методи за потвърждение на резултати от изчисления и други. Но на този етап всичко това се случва в академичните среди и този тип архитектура не се използва на практика. [19] [21]

Ако основната софтуерна архитектура на дадена MMO игра е основана на системата Клиент-Сървър, то голяма част от техническите проблеми вече имат възможни решения. Затова от този момент ще разглеждам само нея. Тук обаче възниква въпросът за **хостинг**.

3.5 Проблеми с физическото изграждане на сървърна архитектура

Както вече споменах, MMO игрите често изискват доста повече от обикновен сайт. Необходими са машини с доста добра производителност, а такива машини идват със съответните изисквания за поддръжка. Основните варианти за хостинг са три.

1. Самостоятелен хостинг – Този вариант е най-труден от организационна гледна точка. Трябва да се обърне внимание на абсолютно всеки аспект свързан с поддръжката и експлоатацията на сървъри. Това включва осигуряване на достатъчно бърза интернет връзка, закупуване на подходящи сървъри и съответните поддържащи елементи, осигуряване на подходяща среда за сървърите, изграждане на защити срещу физически атаки и инциденти и допълнителни разходи (за ток, застраховка, DDOS защита и т.н.). Този вид хостинг изиска най-голяма начална инвестиция и често са необходими години работа преди тя да се покрие. Най-подходящ е, ако е сигурно, че ще се използва дълго време.
2. Колокация – Този вариант представлява предоставяне на собствени машини в специално пригодени за това информационни центрове (*Data centers*). Такива центрове имат необходимата инфраструктура за експлоатация и поддръжка на сървъри и могат да гарантират определени скорости и обем на трафика. Първоначалната инвестиция е много по-малка от тази на самостоятелен хостинг, но пък този вариант е обвързан с месечна или годишна такса. Най-подходящ е за организации и лица, които имат нужда от сървъри с висока производителност,

но не искат да се занимават с поддръжката. В зависимост от информационния център, този вариант може да се окаже и най-ефективния откъм разходи.

3. Нает хостинг – Тук спадат много услуги, но основната идея е, че клиентът използва „под наем“ чужд сървър. Най-популярните вида нает хостинг са VPS²¹, *Cloud hosting*²² и *Dedicated hosting*²³.[\[22\]](#) Този вариант се възползва от всички плюсове на колокация, защото почти винаги наетите сървъри се намират в информационен център, и освен това не изисква никаква начална инвестиция. За сметка на това обаче това е най-скъпия хостинг в дългосочен план. Месечните такси са по-високи от тези при колокация, защото в тях се включва абсолютно всеки възможен разход, включително и инвестициите за сървърите, която не се прави от клиента. Този вариант е най-подходящ, когато няма наличен сериозен начален бюджет и има възможност приходи всеки месец да покриват разхода за хостинг.

Изборът на подходящ хостинг е много важен етап за всяко интернет начинание, но особено при ММО игрите. И е важно да се направи правилна оценка на разходите. Например да се поддържа подходяща система за създаване, съхраняване и управление на резервни копия на информация за играта е свързано с допълнителен разход, но ако този разход се спести има възможност да бъдат загубени месеци труд. Всеки компромис с едно или друго изискване се отразява пряко на играчите и може да доведе до провал на играта.

3.6 Възможни решения на проблем 1

Първият проблем се занимава с обмяната на информация между игровия софтуер на всеки играч. С използването на Клиент-Сървър софтуерна архитектура, този проблем се свежда до обмяна на информация между играчите и даден сървър. Представих няколко основни изисквания за това – висока скорост на обмен на информация и тази информация да бъде без загуби, защитена и достъпна от всички играчи.

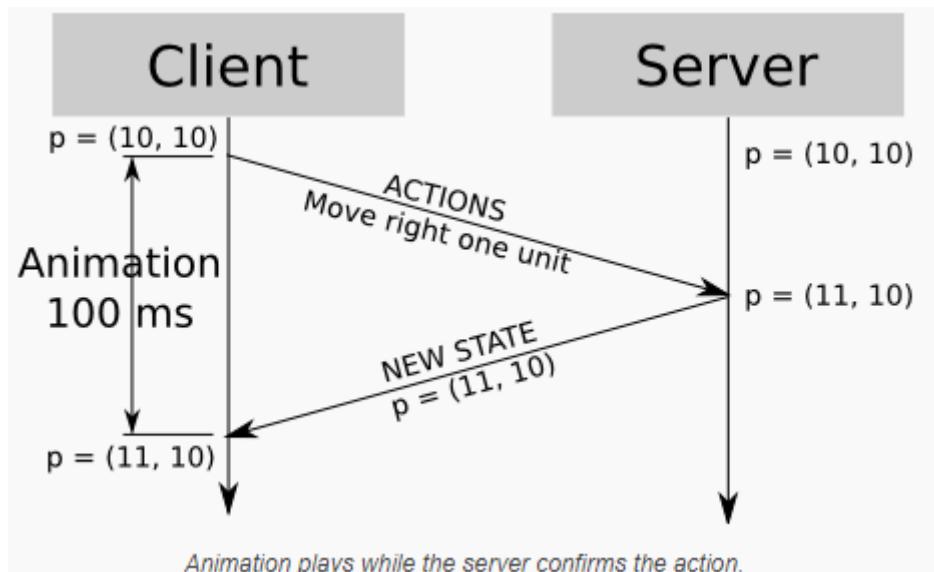
За да се гарантира висока скорост не е достатъчно само да се осигури хостинг, чиято инфраструктура да позволява високи скорости. Това е така, защото за връзката са от голямо значение и възможностите на играчите. В повечето случаи, потребителите имат сравнително ниски скорости, които допълнително се забавят в определени пикови часове. Поради факта, че разработчиците нямат контрол над това, остава само едно решение – да се намали количеството информация, което трябва да се обмени. Тук силно зависи какъв тип ММО игра се разработва. При някои просто няма такова количество данни за обмен, че по-ниските скорости да оказват влияние. При други, особено тези с много ограничено пряко взаимодействие между играчи, игровите действия се извършват през по-голям период от време, следователно по-рядко се комуникира с игровите сървъри и има достатъчно време за тази комуникация.

²¹ *Virtual Private Server* – Виртуален сървър. Представлява изолирана виртуална среда, която имитира физическа машина. Тази инстанция се намира на някаква реална физическа машина, заедно с други подобни.

²² Облачен хостинг. Представлява изолирана виртуална среда, която имитира физическа машина, но е изградена върху система от много на брой реални физически машини.

²³ Нает сървър. Представлява физическа машина, която не се споделя с други. Получава се достъп до всичките и ресурси.

Затова проблемът се среща основно при ММО игри с голямо пряко взаимодействие между играчи, с много действия и информация за тях. Най-често срещаният метод за намаляване на необходимите за обмен данни е да се остави на игровия софтуер на всеки играч да извърши част от работата на сървъра, въпреки че същата се извърша и от самия сървър. В този случай е достатъчно да се провери само крайният резултат за да се избегнат разминавания и евентуално да се коригира. Допълнително може да се имплементира система за предсказване на действия, което значително ще ускори комуникацията.



Фигура 10 - Основна имплементация на система за предсказване. Даден персонаж се намира на позиция $p = (10, 10)$. Играч извърши движение с една позиция, информация за което ще пристигне след 50мс до сървъра на играта и още 50мс за обратно потвърждение. Вместо да чака това потвърждение, клиента допуска, че ще го получи и стартира анимация, която отнема 100мс. Така цялата комуникация изглежда мигновена. Схемата е от статия за Клиент-Сървър комуникация в игри. [23]

Проблемът за загуба на информация има стандартни решения и зависи от това какви мрежови протоколи ще се използват. Конкретно за обмяната на пакети най-често се използват TCP и UDP²⁴. В случай, че дадено събитие задължително трябва да се обработи от сървъра, се използва по-сигурният протокол TCP. Но за действия, които се случват много пъти и постоянно идва информация за следващата итерация на дадено събитие (например вървене на игрови персонаж) е за предпочитане да се използва UDP, защото пропуснат пакет се компенсира почти веднага от следващия в поредицата. Често обаче е важно да се предприемат допълнителни мерки за запазване на последователността на тези пакети, която не е гарантирана при UDP. Използване на контролни суми и защитена връзка също е важно за да се осигури защита на пренасяната информация.

Наличието на сървър дава възможност да се съхранява на едно място състоянието на игровата среда и така на теория може да се споделя с всички участници. Това е достатъчно за покриване на последното изискване.

²⁴ Transmission Control Protocol и User Datagram Protocol са едни от основните в семейството на интернет протоколите. TCP и UDP са част от „Транспортен слой“ и се използват именно за обмяната на пакети с информация.

3.7 Възможни решения на проблем 2

Този проблем разглежда взаимодействието между играчите и трудностите около него. Представих два основни типа взаимодействие – пряко и непряко. Отново, ако основната сървърна архитектура е Клиент-Сървър имаме очевидно решение и на двета вида. Това се вижда по-лесно при непрякото взаимодействие между играчи, където връзката между тях се осъществява посредством състоянието на някакъв междинен обект. Достатъчно е да се проектира играта така, че всички тези обекти да бъдат съхранявани на сървъра и да бъдат достъпни по един и същи начин от всички.



Фигура 11 - В MMO играта *Stronghold Kingdoms*²⁵ всеки играч може да основе ново село само на определени места (оградено в червено). Ако състоянието на всяко от тези полета не се съхранява на сървър, то двама или дори повече играчи могат да заселят едно и също място, което ще доведе до две противоречащи си състояния на игровата среда.

Доста по-предизвикателно е прякото взаимодействие. При него трябва да се обърне внимание на синхронизация, последователност, как се скалира при огромен брой хора и т.н. Важно е обаче, то да е съвместимо с непрякото взаимодействие, защото те често се преплитат. На Фиг. 11 дадох пример за действие – основаване на село. На дадено поле в играта *Stronghold Kingdoms* даден играч може да основе село много преди друг, нов играч, да започне игра, защото информацията за това се съхранява на сървър. Това действие на пръв поглед е непряко, но реално не е само такова. Двама души, които играят едновременно, могат да решат да основат село на едно и също поле и приблизително по едно и също време. Следователно трябва да се вземат под внимание и двете ситуации при дизайна на това действие.

При прякото взаимодействие е много по-трудно да се предложи универсално решение, защото то силно зависи от вида на играта. В *Stronghold Kingdoms* и много от стратегическите MMO игри, има време на пътуване между обекти в играта, което зависи от разстоянието между тях. Този елемент в тези игри може да се използва за по-лесно подреждане на събития и то по повече от един начин. Достатъчно е да са ясно дефинирани всички действия, които могат да бъдат преки, и всеки игрови софтуер да изпраща и получава възможно най-бързо информация за тях до сървъра.

²⁵ <https://www.strongholdkingdoms.com/>

В игрите от първо лице обаче е по-трудно. Прякото взаимодействие се преплита донякъде с проблем 1, но тук въпростът е не как да се обменя информация, а как да се обработи²⁶. Най-използваното в практиката решение е изграждане на играта като един голям програмен цикъл. На всяка итерация от този цикъл всяко нещо в игровата среда се променя по малко – обекти се придвижват с някаква миниатюрна стъпка, атрибути се обновяват според действията на играчите и т.н. Всяко действие в играта е синхронизирано с този цикъл или наречен още вътрешен часовник²⁷. Обикновено това се случва много пъти в секунда, но отново зависи от вида на играта и колко динамично е прякото взаимодействие. В игри със стрелба например, където снарядите се движат с огромна скорост, са характерни 64, 128 и дори повече итерации на игровия часовник в секунда (*ticks per second*, или съкратено **TPS**). В играта *Minecraft*, където битките са с мечове, брадви и лъкове, е налице 20 тиков цикъл – 20 итерации в секунда. Важно е всичко това да се обработва от сървъра за да се гарантират еднакви резултати и еднакво възприятие от всички играчи.

Очевидно е, че при по-голям брой хора ще е нужно повече време на сървъра да обработи всеки игрови тик, поради повечето действия. Следователно ще бъдат необходими или машини с много висока производителност, или повече време за всеки тик. Затова в ММО игрите, висок **TPS** е трудно достижим. Възможно решение е разделяне на игровата среда на зони, всяка с отделен вътрешен часовник. Това се среща в MMORPG игри с огромни светове като *World of Warcraft*. В тази игра се среща още едно интересно решение на проблема – отделен часовник за различни действия. Например продължителността на временни ефекти от заклинания могат да се следят от вътрешен часовник с 1-2 **TPS** и така да не се „задръстват“ по-динамични действия, които се обработват с по-висок **TPS**.

Прякото взаимодействие представлява най-голямото предизвикателство в проектирането на ММО. Много наподобява разработката на първите игри преди десетилетия, когато компютрите са с много ограничени ресурси и всяко неефективно решение оказва лошо влияние над цялата игра. Изиска много внимание, но ако се направи правилно, то шансовете за успех на цялата игра се увеличават много, защото най-важното в едно ММО и това, което носи най-голямо удоволствие на играчите е именно възможността да взаимодействаш с други реални хора директно.

3.8 Възможни решения на проблем 3

Този проблем реално възниква по време на интеграцията на игровите сървъри, но е директно обвързан с взетите до този момент решения и при недостатъчно планиране в началото може да наложи промени. Той е като обобщение на работата до този момент. Докато предишните проблеми се занимаваха със самата концепция „огромен брой хора“, то тук вече са от интерес конкретни стойности.

За да има някакъв реален смисъл величината **TPS** трябва да се знае точната производителност на сървъра. Например в играта *Minecraft*, където е налице 20 тиков часовник, е необходим процесор, способен да обработва всеки тик за не повече от 50 милисекунди. В противен случай, възприятието на играта от играчите няма да бъде такова, каквото трябва да бъде. Това в най-добрая случай означава **лаг**, а в по-лоши нарушаване на последователността на важни действия. Тук от значение е очаквания брой хора играещи

²⁶ Този проблем не е уникатен за ММО игрите, но е допълнително усложнен от огромния брой хора.

²⁷ От тук идва и използвания термин – тик.

едновременно. При по-голям брой ще бъдат необходими повече действия извършвани всеки тик за да се достигне до следващото състояние на играта. На практика на този етап не трябва изобщо да възниква такава трудност и играта да е проектирана така, че да могат съвременните процесори да извършат необходимата работа. В краен случай обаче могат да се предприемат допълнителни мерки. Едно решение е да се постави лимит на играчите, които са на линия по едно и също време. Важно е да се знае, че това ще се отрази негативно на играта в дългосрочен план и силно ще ограничи нейното развитие. Негативното влияние може да се омекоти с техники, като поставяне на опашка за влизане в играта или изграждане на приоритетна система, която дава предимство на определени играчи.

Друг вариант е изграждане на паралелни версии на игровия свят с помощта на повече сървърни инстанции. Това не е същото като разделяне на света на зони, защото тук едно и също място се „копира“ на няколко инстанции. Този вариант е възможен само при игри, в които средата не може да бъде сериозно модифицирана от играчите и се създава илюзията, че всички са едно място (въпреки че виждат само локален набор от играчи, които са попаднали в същата инстанция).



Фигура 12 – „Легендарния остров“ е част от света на WynnCraft²⁸, която е напълно автономна MMORPG игра изградена върху Minecraft. Въпреки че островът е един, той съществува паралелно в огромен брой сървъри, в които играчите могат да ходят по избор (инстанция 34 е показана на снимката в горния десен ъгъл). Всеки сървър има лимит на играчите, но едновременно играят хиляди. Въпреки, че прякото взаимодействие е ограничено до 40 души, всяка инстанция споделя информация с останалите и при смяна на инстанциите играчът на практика не усетва разлика. Всичките му предмети, прогрес, локация и т.н. съществуват едновременно навсякъде и така се запазва условието да може да се взаимодейства с всички играчи (в този случай непряко) за да бъде играта MMO.

Както споменах, точната оценка на изискванията за хостинг зависи от взетите решения до този момент, но още повече от вида на играта. При игри, в които играчите могат да модифицират игровия свят са характерни огромен брой I/O операции. В такива случаи е необходимо да се използва SSD²⁹ тип памет вместо традиционните твърди дискове. В

²⁸ <https://wynncraft.com/>

²⁹ Solid State Drive - Полупроводниково дисково устройство. Без механични части, то е способно да пише и чете записаната информация в пъти по-бързо от твърдите дискове.

противен случай, огромния брой хора може да доведе до забавяне на играта, и колкото и да е мощен даден процесор, играта ще започне да засича.

Друг проблем, който споменах, е този за съхраняване на огромната информация. От една страна трябва да се направи оценка за необходимото пространство, което трябва да се задели. Но от значение е и как ще се съхранява тази информация. Силно е препоръчително да се използват специализирани за много информация бази данни като *HBase*³⁰, *MongoDB*³¹, *Cassandra*³². Ако създаваното ММО е с *относително* малък брой хора са подходящи *MySQL*³³ и *SQLite*³⁴. Поради естеството на ММО, не е добра идея да се съхранява информация директно във файлове, особено ако е отделен файл за всеки играч.

Необходимо е да се обърне внимание и на игровия софтуер при играчите (клиента). Много игрови аватари на едно място може да затрудни по-старите видео карти и процесори. Масовият потребител не използва компютри с висока производителност, като в зависимост от географския регион това може да е сериозен проблем. Решението е предоставянето на различни настройки, с които всеки потребител може да ограничава визуалното представяне на получаваната информация до абсолютен минимум. Важно е обаче това да се направи така, че да не наруши баланса на играта и да не води до нечестно предимство.

3.9 Обобщение на технически проблеми при проектирането на ММО игри

В тази глава разглежда основните чисто технически проблеми, които възникват по време на разработката на една ММО игра. Важно е да се отбележи, че тук не говоря за **всички** възможни проблеми, преди които се изправят разработчиците, а само тези, които възникват поради природата на ММО – основно огромния брой хора. Като всеки друг софтуерен проект, налице са редица препятствия. Това са проблеми свързани с техническата работа като нереалистични очаквания, изпуснати срокове, лоша документация, надвишаване на бюджет, недостатъчно тестване, проблеми с инструментите за разработка и много други.

Освен това възникват проблеми, които са типични за видео игри, но не зависят от броя играчи като трудностите при създаване на изкуствен интелект за играта, оптимизирано използване на изчислителни ресурси, логиката зад взаимодействието с игровата среда и т.н. Всеки от тези проблеми заслужава подробно разглеждане, но това излиза извън обхвата на тази дипломна работа.

³⁰ <https://hbase.apache.org/>

³¹ <https://www.mongodb.com/>

³² <https://cassandra.apache.org/>

³³ <https://www.mysql.com/>

³⁴ <https://www.sqlite.org/index.html>

Най-важните изводи за конкретните проблеми, възникващи от природата на ММО са:

1. Усложняване на по-тривиални проблеми свързани с обмен и съхранение на информация чрез интернет.
2. Силно зависят от конкретната ММО игра.
3. За преодоляването им е необходимо подробно планиране на цялата архитектура още преди да се започне каквото и да е писане на код както и добра оценка на очаквания брой хора.

Техническите проблеми са предизвикателство за всеки тип софтуер. С разрешаването им обаче, трудностите в голяма част са приключили. Това не важи при проектирането на ММО видео игри, поради допълнителните проблеми породени от самите играчи. Тези проблеми не са просто някакво неудобство като лош потребителски интерфейс например, а са също толкова сериозни, колкото техническите, и ако не им се обърне внимание цялата игра може да претърпи провал. Затова в следващата глава ще разгледам социалните проблеми при проектирането на ММО видео игри.

4 Социални проблеми при проектирането на ММО видео игри

Често срещан проблем при проектирането на какъвто и да е софтуер, в който има някакво взаимодействие с реален потребител, е че хората са непредвидими и могат да предизвикат със своите действия непредвидени резултати дори всичко чисто софтуерно да работи. Както с техническите проблеми, които разгледах в предходната глава, отново естеството на ММО усложнява в пъти този проблем и поражда редица трудности, които аз ще нарека най-общо социални проблеми при проектирането на ММО игри.

Тези проблеми не са явни по време на разработката на дадено ММО, а се проявяват известно време след старта на самата игра. Възможно е, докато ефектите от даден проблем вече ясно се виждат, да е прекалено късно да има възможни решения и играта претърпява провал. Затова тук е дори още по-важно да се анализират подробно всички възможни основни взаимодействия между играчи и да се търсят слабости, което не е лесна задача. В тази глава ще покажа едни от най-често срещаните проблеми, които през годините са водели до проблеми в популярни ММО заглавия, и на които задължително трябва да се обърне внимание.

4.1 Проблем 1: Изграждане на игровата среда

Една от първите задачи при разработката на каквото и да е видео игри, но особено при ММО е да се изгради концепцията за игрова среда. Това включва не само как ще я виждат играчите, но и как ще функционира тя, какви правила ще определят съществуването на всеки един елемент в тази среда и най-вече какви действия могат да извършват играчите помежду си и със света. И тук възниква въпросът каква степен на свобода на действията да имат играчите в този свят. Колкото по-свободни са те, толкова по-интересна ще бъде ММО играта, защото съдържанието идва директно от изобретателността на хората. Но тази свобода може да се окаже нож с две остриета, защото отваря възможност за така наречените **експлойти**.

Думата идва от английската „*exploit*“ и има няколко значения, но най-често означава да извличаш полза от нещо³⁵. В софтуерните среди обаче придобива още едно значение – възползване от слабост в даден софтуер, обикновено с лоша умисъл и за лична изгода. Тези слабости невинаги са причинени от някакъв бъг или грешка в кода, което ги прави изключително трудни за предотвратяване или откриване. Известен пример са **CVE-2017-11882**³⁶ и **CVE-2017-0199**³⁷. Открити през 2017 тези слабости в офис пакета на *Microsoft* отварят задна врата към компютрите на милиони потребители. Такива слабости са от национален интерес и се следят от правителства по цял свят.[\[24\]](#)

Експлойтите са особено голям проблем в ММО игрите. Част от тях, предимно технически слабости, се използват за създаването на зловреден софтуер, за който споменах в предходната глава. Но тези, които са още по-опасни, защото от тях може да се възползва

³⁵ <https://dictionary.cambridge.org/dictionary/english/exploit>

³⁶ <https://nvd.nist.gov/vuln/detail/CVE-2017-11882>

³⁷ <https://nvd.nist.gov/vuln/detail/CVE-2017-0199>

всеки, са експлойти в контекста на игровата среда. Това са слабости, които възникват или поради прекалено голяма свобода на действия на играчите, или пропуски в логиката при изграждане на игровия свят (често и комбинация от двете).

Обикновено игровата средата се проектира така, че да имитира реалния свят и неговите закони. Този подход обаче е една от основните причини за слабости в едно ММО, защото нещо, което работи по един начин в реалността, по една или друга причина работи напълно различно в контекста на играта. Може би най-типичният пример за това са *AFK*³⁸ експлойти. Те представляват действия в играта, които даден играч може да изпълнява без дори да е пред компютъра³⁹, но докато неговият аватар е вътре в играта. Това са най-често монотонни, повтарящи се действия, които обаче носят някаква награда или полза. Това веднага нарушава баланса в игровата среда, защото на практика позволява денонощна игра и хора, които не използват такъв експлойт, започват да изостават в своето игрово развитие. Такива експлойти са в популярни ММО игри и до днес като например експлойт в играта *World of Warcraft*, който дава възможност на играчите да развиват свой герой без реално да играят, само със стоеще на определени места. [25]

Друга често срещана слабост в изграждането на игровия свят е правенето на едни действия в играта много по-ефективни от останалите. Този тип експлойти са толкова много и толкова различни, че е трудно дори да се говори едновременно за всичките. Затова ще ги разделя на два вида. Първият вид са такива, от които всеки играч може да започне да се възползва по всяко време и без нищо необходимо. Другият вид са тези, за които трябва известно време да се играе и изискват някаква подготовка. Но независимо от типа, наличието на такива експлойти принуждава играчите да играят по един и същ начин, дори да има възможност за разнообразие, защото често тяхна основна цел е да достигнат възможно най-далеч в своето развитие, за възможно най-кратко време. Това не само наруши баланса в играта, но е причина тя по-бързо да доскуча на потребителите.

Дори ако всичко е проектирано и създадено както трябва, често проблеми носят и бъдещи **ъпдейти** на играта. Игровата среда, след начало на играта, постоянно се променя и това е предпоставка за пропуски при въвеждане на нови елементи. Такива пропуски на теория не трябва да съществуват, ако се гледа началното състояние на игровия свят и затова лесно се пропускат. Такъв е например експлойт с щети от дронове⁴⁰ в играта *Eve Online*⁴¹. При една от актуализациите на играта, един елемент от нея – дрон, е значително подобрен, но е поставен лимит на броят, който може да се използва. Игровият свят обаче вече е пълен с изоставени дронове, които получават подобренията, но не се влияят от лимита. Това води до изключително неравностойни битки и дори се стига до наказания на хванатите да се възползват от това играчи. [26]

Последният основен елемент, на който трябва да се обърне внимание при изграждането на игровия свят, е икономиката в играта. Там много лесно могат да възникнат слабости, от които играчите да се възползват и затова сега ще се спра по-подробно на нея.

³⁸ Away From Keyboard, Далеч от клавиатурата. Използва се за да се каже, че играчът в момента не е пред экрана.

³⁹ Достатъчна е обикновена програма за автоматизация или дори просто поставяне на тежест върху клавиатурата или мишката.

⁴⁰ <https://wiki.eveuniversity.org/Drones>

⁴¹ <https://www.eveonline.com/>

4.2 Проблем 2: Икономиката в играта

Дефиницията за икономика в ММО игра не се различава от стандартната дефиниция. В общия случай тя представлява някакво обобщение на състоянието на даден регион, държава или в този случай игрова среда от гледна точка на производството и потреблението на сировини, продукти и услуги⁴².

Икономически отношения между играчите могат да възникнат без дори това да е заложено в софтуера на играта. Те могат да бъдат съвсем елементарни под формата на бартер на предмети или услуги, но могат да бъдат и по-сложни отношения с участието на някаква валута, която играчите сами са определи за ценна. Да се остави това в ръцете на потребителите обаче не е добра идея, защото означава липса на контрол и в повечето случаи води до пълен колапс. **Срыв в икономиката на ММО игра може да доведе до провал на играта.**

За щастие, всяка икономическа система в дадена ММО игра би имала някакъв реален аналог. Това е от огромна полза, защото дава възможност да се черпи информация от огромната литература по темата за да се проектира в самия софтуер на играта някаква система, която да регулира и стабилизира икономиката. Има обаче някои особености, които възникват от природата на видео игрите като цяло, но както с повечето разгледани проблеми са значително по-сериозни в контекста на ММО. Това е изключително обширна и сложна тема и заслужава самостоятелно разглеждане, но за целите на тази дипломна работа ще се спра само на най-основните моменти.

Първият проблем при проектирането на някаква икономическа система идва още в началото и е свързан с това как дадена валута достига до играчите. В реалния свят, парите се контролират от банки и правителства. Те управляват количеството навлезли в обръщение пари, като така контролират до някаква степен възможните проблеми, най-вече инфлация, която представлява трайно увеличение на цените на продукти и съответния спад в покупателната способност на парите⁴³. [\[27\]](#)

Във видео игрите, навлизането на пари в обръщение е обвързано с игрови механики, например с убиването на чудовища [\[28\]](#). Следователно тези игрови механики ефективно се превръщат в машини за принтиране на пари, които са без контрол. Това е основната причина почти всяка ММО игра да страда в някакъв етап от живота си от инфлация и дори хиперинфлация. Из многото примери се откроява играта *Asheron's Call*, в която основната валута до толкова се обезценява, че играчите са принудени да я изоставят напълно и да използват създадени от тях валути, базирани на предмети от играта. Друг пример е играта *Gaia Online*, която е ударена от такова ниво на хиперинфлация, че разработчиците предлагат да дарят по 250 долара за благотворителност за всеки играч, който доброволно унищожи огромен брой (15 трилиона) злато. [\[29\]](#)

Тези проблеми може и да не се видят веднага. Това е така, защото важността на някаква валута в общия случай е много по-малка в една игра от колкото в реалния свят. В едно ММО липсват тези ежедневни разходи, които са част от модерния свят като харчене на пари за храна например. Обикновено такива ключови нужди в една игра са лесно достъпни в контекста на играта и всеки може да си ги набави сам с игра, без да е нужно да харчи пари

⁴² Дефиниция от Lexico.com - <https://www.lexico.com/definition/economy>

⁴³ Дефиниция от Dictionary.com - <https://www.dictionary.com/browse/inflation>

за тях. Още повече, че играчите не стоят в играта денонощно, което допълнително намалява възможните разходи. Тези обстоятелства ограничават циркулацията на пари в играта, което е причина негативните ефекти от инфлацията да не са видими веднага.

Инфлацията в едно ММО води до редица вторични проблеми. Затруднява се търговията между играчи, защото трудно може да се оцени даден предмет или услуга. Нови играчи могат много по-лесно да се сдобият със скъпи предмети или функционалности, а стари играчи, които не са играли известно време, се завръщат практически разорени. Освен това, играта може да се сведе само до малък набор от действия, които носят максимално количество от валутата, защото всяко друго действие би било загуба на време (и играта се превръща в еднообразно „фармене“). На такива проблеми няма смисъл да се търси решение, а е по-важно да се изгради стабилна икономическа система.

Следващ проблем са всичките евентуални експлойти, които директно влияят на икономиката. За разлика от реалния свят, където е предизвикателство да се фалшифицират пари, то в една ММО игра това може да се окаже възможно поради пропуски в логиката на играта или бъгове, а понякога и с помощта на допълнителен софтуер. Това например се случва в играта *Everquest II* през 2005. Разработчиците забелязват, че в рамките на едно денонощие, количеството циркулираща валута се увеличава с 20%, поради бъг с който играчите копират една от валутите в играта – платина. За щастие, успяват да реагират навреме и дори да възстановят състоянието на икономиката преди атаката. [\[30\]](#)

До този момент говорих единствено за пари и валути. Това е така, защото техните качества като измерител на стойността на продукти и услуги дават възможност да се избегне занимаване с по-фундаментални икономически проблеми, които сякаш ще се разрешат сами, ако се изгради достатъчно стабилна игрова валута. Това обаче не винаги е така. Всяко общество от хора, включително и в една ММО игра, трябва да се сблъска и разреши три основни проблема в контекста на породената икономическа система. Те са най-общо въпросите: какви продукти и услуги се произвеждат (и в какви количества), как се произвеждат и за кого. [\[31\]](#)

Какви трудности за разработчиците пораждат тези проблеми отново зависи силно от вида на проектираната ММО игра. От значение са много неща като например дали играта е от първо лице, дали се играе в реално време, какъв е мащабът, какви са възможностите на играчите и много други. На практика икономиката е силно обвързана не само с основните, но и с второстепенните игрови механики и от решения свързани с нея ще зависи как играчите ще играят.

Ако играта е планирана добре и има ясно изградени цели, то няма да бъде особено трудно да се обърне внимание на първия въпрос – „какво се произвежда в играта и в какво количество“. За всяка игрова механика ще са нужни някакви предмети или сировини. Например в ММО играта *EverQuest II* има изградена система за готовене, чрез която се произвеждат различни храни. От това следва и въвеждането на различни съставки като месо, плодове, зеленчуци, подправки и други. Предизвикателство обаче е въпросът за произход и количество.

Тези две характеристики са тясно свързани. Откъм произход⁴⁴, всеки предмет може да съществува от началото на играта като част от игровата среда (да бъде добиван), да има

⁴⁴ В смисъл да се появи ново копие на предмета в игровата среда.

възможността да бъде произвеждан или и двете. Ако игровият свят е краен, то всички предмети, които се добиват, ще бъдат с ограничено количество през цялото съществуване на играта. Още повече, предмети, които използват добивни суровини за производство също ще бъдат с краен брой. Това е сериозен проблем за една ММО игра по редица причини, най-малката от които е фактът, че ако ключови за съществуването на играчите предмети свършат, то играта няма как да продължи.

Също толкова труден е и въпросът за производство на различните предмети (и евентуално услуги). По-конкретно, най-трудно тук е балансирането между различните начини за създаване на предмети в играта. В много от игрите, в които играчите се изправят един срещу друг, има някаква имплементация на оръжия породена именно от нуждата за по-интересни битки. Като често използван предмет, обикновено оръжията имат повече от един начин да бъдат получени. Един меч например може да бъде купен от търговец, може да бъде част от плячка при победени врагове, може да бъде произведен в съвсем прям смисъл от играчи с помощта на суровини и т.н.

Всеки от тези методи се различава коренно от останалите по необходима подготовка и изпълнение. Балансът е важен, защото ако играчите могат много по-лесно да си произведат сами оръжието, то не биха се занимавали да търсят плячка (действие, което е често обвързано с риск). Така може да се окаже, че цели игрови механики са игнорирани напълно от потребителите. Задачата става в пъти по-сложна колкото повече предмети има и особено колкото повече връзки има между тях.

Последният проблем идва от въпроса за кого се произвеждат нещата. Ясно е, че повечето предмети ще се използват от играчите, но този отговор не е достатъчно точен. Често има голяма разлика между предмет пред назначен за нови и такъв за развити играчи например. Някои предмети пък могат да бъдат създадени само за ползване от компютърно управлявани врагове. Такъв предмет е оръжието *Frostmourne* в играта *WorldOfWarcraft*. Играчите не могат да получат този меч по никакъв начин, което дава възможност на разработчиците да приложат висока щета на меча без да се притесняват, че би повлияло по някакъв начин на битки между играчи например. Това обаче прави още по-важно да се обърне внимание на въпроса за кого е предназначен даден предмет. Ако тази задача се пренебрегне, то могат ще се усложнят други проблеми при проектирането на играта.

Икономиката в ММО игрите е изключително трудна за проектиране и управление. Това е причината почти всички ММО заглавия и до днес да страдат от проблеми с нея. Дори проблемите да не причинят пълен срив в икономиката, то създадените вторични такива влошават игровото преживяване на играчите и затова трябва да се обръща огромно внимание не само при проектиране, но и по време на самото съществуване на играта.

4.3 Проблем 3: Взаимодействие между нови и развити играчи

Още от дефиницията на ММО става ясно, че взаимодействието на играчи е задължителна част от всяка игра в този жанр. Колкото повече свобода на действията имат играчите обаче, толкова по невъзможно става да се предвидят достатъчно техните действия, особено спрямо други играчи. Това може да доведе до неприятно игрово преживяване, което да отблъсне текущи или бъдещи възможни играчи. Затова сега ще се спра по-подробно на

основен проблем, който е характерен при взаимодействие между играчи в едно ММО – взаимодействието между нови и развити играчи.

Този проблем е явен най-често по време на конфликти. Обикновено основните конфликти в игровата среда са два – **PvE** и **PvP**.

1. **Player vs. Environment** (играч срещу средата). Това са конфликтите между играча и игровата среда. Това най-често е битка с **NPC**⁴⁵, но може да представлява и справяне с трудностите на средата – например контролиране на глад или жажда, оцеляване в трудни условия и други.
2. **Player vs. Player** (играч срещу играч). Това е тип конфликт между реални играчи, най-често битка или състезание.

При ММО игрите е характерно, че и при двата типа конфликт имаме взаимодействие между играчи, като при **PvE** те си взаимодействват, като работят заедно срещу средата например⁴⁶.

Основен проблем при тези взаимодействия е нивото на развитие на всеки от участниците. Ако един играч, който е играл играта месец се срещне с друг, който е едва започнал, то конфликта несъмнено би завършил с победа за по-развития. Това напълно обезсмисля **PvP** аспекта в едно ММО. В зависимост от това какво ниво на умения и игрово знание е нужно в една битка между играчи, този проблем може да се задълбочи още повече. За да бъде интересно и изобщо смислено едно **PvP**, то всички участници трябва да имат *реален шанс*⁴⁷ за успех.

Всъщност, дори да няма никакъв конфликт между играчи, нивото на тяхното развитие води до трудности още при самото изграждане на игровият свят, защото ако се проектира за играчи в начален етап на развитие би бил прекалено тривиален за тези в по-късен. Разбира се, важи и обратното – проектирана за развити играчи среда би била прекалено трудна за игра от нови играчи, което на практика би убило играта, защото няма да има приток от нови хора.

Новите играчи са живота на всяко ММО. Те помагат на играта да се разраства и развива. Поредната голяма трудност пред всеки разработчик на ММО игри е именно как да предоставят реален шанс за игра на нови играчи в средата на развити такива и как това игрово преживяване да е възможно най-интересно.

4.4 Проблем 4: Недоброжелатели

В главата с технически трудности споменах за проблем, при който играчи, с помощта на зловреден софтуер, който се възползва от слабости в игровата логика или бъгове, манипулират игровата среда в тяхна полза. В тази глава пък разглеждах и проблема с експloitите. Реално тези неща не вредят на една игра до момента, в който някой започне да се възползва от тях. Във всяка игра съществува такава група от играчи, които ще нарека

⁴⁵ NPC – Non-Player Character, персонаж управляем от играта (не от играч).

⁴⁶ А могат и да си пречат! Например, ако даден играч, чрез своите действия лиши друг играч от храна. Те реално може да не се състезават и да не са в пряк конфликт, но чрез действията си първият играч вреди на втория.

⁴⁷ Разбира се това не означава равностоен, защото тогава пък се обезсмислят останалата част от играта, подготовката и опита.

най-общо недоброжелатели, които извършват вредни за едно ММО действия. Тези действия са от всякакво естество, не само гореспоменатите. Част от недоброжелателите дори нямат за цел да играят играта, а единствено да пречат на нейното съществуване и развитие. За други анонимността, характерна за онлайн игрите, им носи чувство за безнаказаност (или по-точно казано липса на неприятни последици) и са склонни на държание, което не биха си позволили в реалния свят. Сега ще разгледам някои от най-честите действия, които такива хора предприемат, в допълнение на вече разгледаните по-рано. Важно е да се отбележи, че не всеки играч, който извършва тези действия, осъзнава, че вреди⁴⁸, но задължително трябва да им се обърне внимание.



Фигура 13 – Глобален онлайн чат в ММО игра The-West.net

разпространяват фалшиви новини, обиждат и псуват, говорят неща неподходящи за непълнолетни. Всяко от тези действия има потенциала да създаде силно токсична среда в играта, което да доведе до много лошо игрово преживяване и да отблъсква играчи. Допълнителен проблем имат и играчи от малцинства, както и от женски пол. Те често са обект на полова или расова дискриминация, нападки и нежелано нарушаване на личното пространство. Ако не се вземат адекватни мерки, то е много вероятно един от най-основните методи за взаимодействие между играчи – комуникацията, да бъде най-лошият елемент в играта.

Следващият проблем идва от възможността за контакт на потребителите извън игровата среда. В зората на ММО, а и игрите като цяло, този контакт се е случвал във външни чат стаи, интернет форуми и дори в първите социални мрежи. Днес, този контакт е толкова разпространен, че дори е причина различни предоставени в играта комуникационни канали да станат излишни. Най-използвани днес са платформи като *Discord*⁵⁰, или приложения за комуникация като *Mumble*⁵¹ и *TeamSpeak*⁵², но все още се използват и много от старите методи. Проблемът е, че много често разработчиците на едно ММО нямат никакъв контрол върху тази комуникация. Това отваря врати на множество лоши практики.

Всяка популярна ММО игра задължително привлича хора, които не се интересуват от играта по никакъв начин, но намират начини да „печелят“ реални пари или някаква изгода извън контекста на играта. Това може да бъде предложение за търговия на игрови предмети срещу реални пари например. Измамите задължително имат елемент извън играта, затова

Всяко ММО има някаква система за комуникация. Най-често това представлява система за изпращане на текстови съобщения в реално време⁴⁹, но може да бъде и виртуална поща, гласова комуникация и други. Основният проблем е, че в такива системи за комуникация, всеки играч може да достигне до огромен брой хора независимо от това дали те искат да го „чуят“ или не. Недоброжелатели могат да се възползват от това, като

⁴⁸ Например част от хората, които се възползват от експлойти, го правят защото търсят най-лесния начин за успех в играта, не за да причиняват вреди.

⁴⁹ Наречена онлайн чат или само чат.

⁵⁰ <https://discord.com/>

⁵¹ <https://www.mumble.info/>

⁵² <https://www.teamspeak.com/en/>

е изключително трудно да се засекат, особено ако играчите изпратят първи пари на измамника. В по-редки случаи са възможни и по-сериозни проблеми като опити за изнудване, разпространение на лична информация на играчи и дори заплахи. Някои от използваните за тази външна комуникация платформи се опитват да се борят срещу това, като предоставят някаква система за докладване, но често докато играч се усети, че е измамен, е прекалено късно.

Последният проблем, който ще спомена, е на пръв поглед технически, но го включвам в тази глава, защото той не би бил от значение, ако не бяха недоброжелателите - **DDOS** атаки. *Distributed denial-of-service attack* е огромен проблем за всичко, което използва интернет пространството. Целта на такъв тип атака е да се наруши или предотврати напълно достъпа до дадена услуга, като най-често се изпращат огромен брой заявки към уеб сървърите на услугата, което изразходва целия наличен ресурс. Поради естеството на ММО, всички игри от този жанр са изключително уязвими към този тип атака, но най-вече са тези със силно развито пряко взаимодействие между играчи, където има нужда от непрестанна комуникация. По-подробно разглеждане на този проблем излиза извън обхвата на тази дипломна работа, затова ще спра до тук. [\[32\]](#)

Недоброжелателите са причина много от редовните и сериозни играчи на една ММО игра да спрат да я играят. Дали поради игра с непозволени средства като модифициран софтуер, поради влошаване на игровата среда чрез възползване от някакъв експлойт, или просто поради токсичната атмосфера, която такива потребители създават – няма особено значение. Но това несъмнено ще доведе до провал на играта. А с такива проблеми ще се сблъска задължително всяка ММО игра.

4.5 Възможни решения на проблем 1

Този проблем разглежда експлойти предимно в контекста на игровата среда. Както вече споменах, тези експлойти винаги ще съществуват. ММО игрите често са с такова ниво на свобода на действията, че това е просто неизбежно. Но е важно да бъдат открити и коригирани колкото се може повече експлойти преди играта официално да бъде открита. Това е причината при толкова много игри, но особено при ММО, да са характерни по-дълги **бета версии**. Слабостите, които представих във проблем 1, са толкова характерни за този жанр, че задължително трябва да се провери дали присъстват в разработваната ММО игра. Това е най-добре да се случи именно в бета етапа от цикъла на разработка.

АФК експлойтите могат да бъдат контролирани по няколко начина. Най-очевидното е всяко възможно действие в играта да се проектира така, че да изиска взимане на някакво логическо решение или да е нужно постоянно внимание на играча. На практика действията да не могат да се автоматизират. Това обаче е доста сложно да се направи, защото е важно действията да не бъдат прекалено усложнени, което ще направи играта по-малко привлекателна за играчите. Като цяло това е още едно усложнение при проектирането на игри, защото за решението на всеки проблем трябва да се мисли допълнително дали няма това решение да повлияе негативно на игровото преживяване на потребителите. Ако за едно често извършвано действие трябва да се минава през набор от допълнителни стъпки всеки път, то играта много по-бързо ще омръзне на играчите.

Друг начин за предотвратяване на АФК експлойти е разработването на специална система против тях. Най-простата такава система представлява проверка дали в определен кратък

период от време играчът извършва каквото е да е действия. Такава проверка би предотвратила основната форма на АФК експлойт, за която е достатъчно само играчът да бъде в играта без да е нужно да извършва действия. Проблемът е, че е изключително лесно да се заобиколи тази проверка. Достатъчен е някакъв софтуер за симулиране на натиснат бутон на мишката например. Освен това повечето АФК експлойти изискват извършване на монотонни действия. Затова е нужно системата против АФК да бъде по-сложна, а това зависи силно от вида на играта и по-конкретно как играчите взаимодействват със света.

За ММО игри от първо лице е стандартно да се използва позицията на аватара за да се определи дали играчът е АФК. Ако за определен период от време даден играч стои на едно място, то той е АФК независимо дали извършва други действия. Това например би предотвратило разгледания в проблем 1 АФК експлойт в играта *World of Warcraft*. Тогава е използвана само основната система за засичане дали даден играч е АФК и играчите я заобикалят, като на определен период от време автоматично подскачат на едно място. [\[25\]](#)

Такава система обаче има възможност да засече и играчи, които не са АФК, но по някаква причина не се движат (защото пишат в чата например). Затова е важно и какво се случва, когато играч бъде засечен. Обикновено това е прекъсване на връзката между играча и сървъра, но за да се избегнат редките случаи на грешно засичане може преди това да се изиска някакво действие от играча като например **Captcha**. Именно защото се очаква да се случва рядко на играчи, които не са АФК, то вече не би повлияло негативно на игровото преживяване.

Браузър ММО игрите са особен случай от гледна точка на АФК експлойтите. При тях самата игра не изиска постоянно извършване на действия – играчите взимат решения, които отнемат часове да се изпълнят. Проблем обаче си остава автоматизирането на действия. В такива игри е задължително да се използва някаква система, която да гарантира, че действията се извършват от човек. Често това е **Captcha**, но може да бъде и конкретна за самата игра защита – например следене на скоростта на извършване на заявки, позицията на мишката и т.н.

Още по-трудни за намиране и преодоляване обаче са експлойтите, които са свързани с действия в игровата среда. За тях няма много какво да се направи за да бъдат предотвратени. Важното е да се знае как да бъдат открити навреме, което задължително започва с много тестване на играта. Разделих ги на два вида – такива, които нямат нужда от подготовка и такива, които имат. Първата основна разлика между тях е, че вторите е много по-трудно да бъдат открити по време на тестване. Това е така, защото за тяхното откриване е необходимо по-дълбоко разбиране на играта, което **бета-тестърите** все още нямат. Освен това, времето за подготовка за експлойта често е по-дълго от целия период на тестване. За да бъдат открити на време е необходимо постоянно да се следи какво се говори за играта в социалните мрежи, форумите и всички възможни публични канали за комуникация след нейния старт. Всяко ММО има много такива слабости, но те стават опасни за играта единствено когато станат популярни.

Другият вид експлойт са всички тези, за които не е нужна подготовка. Такива слабости се откриват по-лесно по време на тестване на играта, но при тях има и допълнителни методи. Един от тях е да се следи развитието на нови играчи. При тях силно си личи, когато се извършват такива действия, защото колкото и голямо ниво на свобода на действията да има едно ММО, началото винаги е една идея по-предсказуемо и контролирамо. Може да се направи оценка за какво време или по-конкретно за колко време игра може да се стигне до

определенi моменти. Всяко сериозно отклонение е добре да се проучи. Освен игрово време, за оценка на развитието могат да се използват и други характеристики – богатство, предмети дори резултати от PvP конфликти.

Друг вариант за откриване на слабости е всяко възможно действие да получи количествена оценка. Тя може да бъде игрово време, спечелена валута, спечелен опит или други (в зависимост от играта). Това може да бъде трудно за изпълнение, но ако се постигне то ще се види директно, ако дадено действие е много повече „доходоносно“ от останалите. Важно е да се отбележи, че в някои MMO игри е характерно да има такива действия, но те се компенсират най-често от високо ниво на риск. Следователно в такива игри този риск трябва да се вземе под внимание при поставяне на оценка на дадено действие, защото иначе резултатите ще са подвеждащи.

Последният вид слабости, които разглеждах, са такива, които възникват поради бъдещ ъпдейт на играта. Тук тестването в някаква pre-prod среда не е достатъчно, защото играта вече е претърпяла много промени, предизвикани от самите потребители⁵³. Възможно решение на този проблем е промените да се приложат само на малка група играчи и да се следи тяхното развитие. Това обаче не може да се приложи за ъпдейти, които правят основни промени по игровата среда. Още повече, в зависимост от тестваната промяна, тази група играчи има вероятност да получи нечестно предимство или да бъде ощетена не поради експлойт, а само защото играе по „нови правила“, което би повлияло на резултатите от тестването.

Друго възможно решение е създаването на чисто нов тестов игрови свят още в началото. Този свят не е копие на някой друг, а е напълно независима инстанция, на която могат да се тестват всякакви промени, като играчите са предварително уведомени за това. Ако е нужно, може да се постави лимит на броя играчи или някакви други ограничения с цел по-малко разходи по поддръжка на такъв свят, но е важно преживяването да е възможно най-близко до това на основния игрови свят. Тази специална тестова среда се отличава от стандартните с това, че също еволюира с времето и предоставя реални възможности за откриване на експлойти породени от ъпдейти на играта.



Фигура 14 – Допълнителен надпис в тестови свят на играта The-West.net, който напомня на играчите, че са в такъв свят. Играчите са предварително уведомени по време на регистрация, че игровата логика се променя постоянно и дори е възможно пълно рестартиране на свeta без предупреждение.

4.6 Възможни решения на проблем 2

Този проблем се занимава с трудностите при изграждане и управление на икономиката в дадена MMO игра. Той може да се раздели на проблеми с парите и проблеми с икономическата организация. Най-добре е първо да се обърне малко внимание на икономическата организация преди да се започне работа върху игрови валути. Това включва прототипно дефиниране на предмети, суровини, материали и техните възможни ползи. След това да се изгради парична система и на края отново да се обърне внимание на

⁵³ Въпреки това не трябва да се пропуска!

организацията вече в контекста на готовите пари. Този подход на разработка драстично ограничава възможните експлойти с цените в различните аспекти на играта – NPC магазини, ъпгрейди, услуги, награди, плячка и прави много по-лесно разглеждането на класическите три проблема⁵⁴.

Първоначалното проектиране на предмети в повечето случаи е сравнително лесно, защото е доста абстрактно. Могат да се пренебрегнат част от свойствата и характеристиките на предметите, защото така или иначе те ще бъдат разгледани отново в по-късен момент от изграждането на икономическата система. Дори могат да се пренебрегнат фундаментални въпроси от вида на „как играчите ще добиват даден предмет“. Важно е обаче да бъдат ясни неговите възможни ползи. Именно тази информация за функцията на всеки предмет е в основата на редица методи за справяне с проблеми в игровата икономика и по-конкретно при проблеми с парите.

Преди да разгледам основния проблем на игровите валути обаче ще кажа няколко думи за самите пари. Както в реалния живот, основната им функция е да бъдат средство за размяна. Дори без да участва директно в търговия, е характерно всеки предмет в играта да има цена, като тя може да бъде фиксирана от разработчиците, определена от играчите на пазарен принцип или комбинация от двете. Парите могат да бъдат напълно виртуални⁵⁵, но и да представляват предмет в играта. Напълно виртуалните валути са често предпочитани от разработчиците, защото са по-лесни за създаване, по-лесни за следене и по-трудно се намират експлойти, които да се използват за тяхното копиране или манипулация.

Независимо от вида валута обаче, основния проблем на парите в MMO игрите е инфлацията. По-конкретно, този проблем може да се опише като „По-голямо количество пари в циркулация от планираното в началото“. Ще представя две коренно различни решения, като всяко си има своите предизвикателства. Идеята зад първото е да се създаде баланс между навлизащи и излизащи от циркулация между играчите пари. На теория, ако достатъчно пари се премахват от играта напълно, то ще се компенсира това, че първоначално се появяват от нищото и тяхната покупателна стойност няма да се променя драстично. Допълнително улеснение е, че за разлика от реалния живот, в MMO игрите не е особен проблем да има моменти, в които повече пари напускат игровата среда, защото парите не са необходими за съществуване на играчите. Въпреки това си остава предизвикателство да се намери необходимия баланс.

Друг подход е предварително създаване на цялото необходимо количество пари, като играчите не могат да създават повече, а единствено обменят вече създадената валута помежду си. Така със сигурност използваната валута никога няма да се обезценя. Въпреки че това решение на проблема с инфлацията изглежда по-сигурно, то е и много по-трудно за реализация. Най-вече стои въпросът какво точно да бъде това начално количество пари. Ако е базирано на количеството очаквани играчи, то грешна оценка на това би довело до недостиг на парите сред играчите или обратното до инфлация още от началото, ако много по-малко играчи играят играта. Още повече, играчи, които спират да играят играта, на практика изолират притежаваната от тях валута от останалите играчи, което води до постепенно изчезване на парите. Тези проблеми разбира се си имат своите възможни решения, но първият подход е много по-гъвкав и е много по-подходящ за повечето MMO игри. Затова повече няма да разглеждам варианта с предварително създаване на пари, а ще

⁵⁴ Трите проблема представени в книгата Economics на Samuelson – „какво“, „как“ и „за кого“ [31]

⁵⁵ Тоест играчите само имат информация за своето богатство, без то да има някакъв предметен аналог като банкноти или монети.

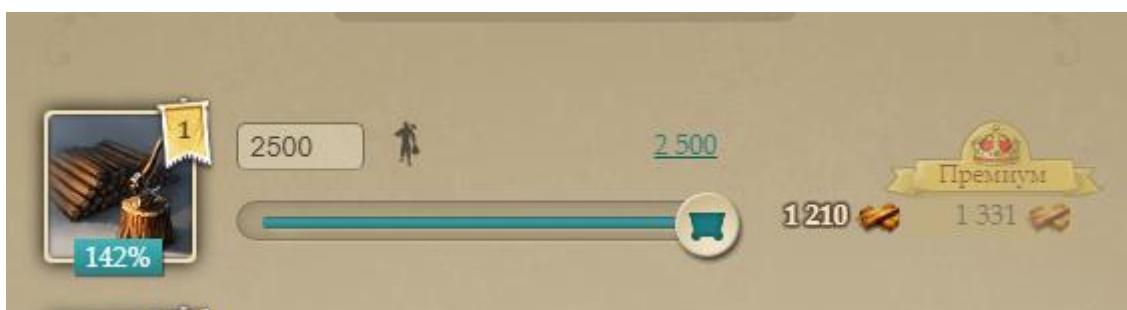
се спра по-подробно върху подхода с баланс между създаване и премахване по време на играта.

Измежду основните причини за инфлация се открява прекалено лесното създаване на пари и влизането им в играта. Има няколко възможни действия, които да предотвратят това. На първо място е директно ограничение на печалби. Това означава, че за един час игра например не може да се спечели повече от ясно дефинирано количество валута. Самите играчи могат да достигнат лимита и за няколко минути, но общото количество навлезли пари никога няма да бъде повече от предварително зададеното. Това може да се постигне или като се сведе до нула прихода от дадено действие (независимо колко пъти се извършва), или да се забрани даденото действие след достигане на лимита. Така може теоретично да се знае колко най-много пари ще имат играчите и по това да се изграждат цени в други аспекти на играта.

Друго възможно действие е печеленето на пари да се обвърже с игрови действия, за които се знае времето за извършване и следователно печалбата. Това време е или точно дефинирано, или с помощта на други ограничаващи елементи е невъзможно действията да се извършат по-бързо. По този начин отново се постига необходимият ефект за контрол на влизашите в циркулация пари, но без ограничения на игрови елементи.

Вариант за по-съревнователните MMO игри е парите да се печелят в състезания, които се случват определен брой пъти и имат ясно дефинирани награди. Така отново се постига контрол над парите, но с рисък да се ограничи развитието на по-слаби играчи.

С обвързването на пари с игрови елементи се обръща внимание и на друг разгледан проблем – дадено действие да бъде много по-доходносно от друго. Като се използва допълнителна величина като време ще бъде много по-лесно да се оцени възможният доход и да се избегнат ситуации, в които играчите вършат едно и също нещо отново и отново с цел най-много приходи за най-кратко време.



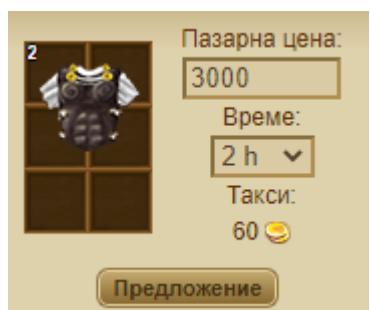
Фигура 15 – В браузър MMO игра Imperia Online, играчите могат да назначават работници за добив на дървесина, необходима за строене на сгради, което е основно действие в играта. Колкото повече работници има, толкова повече дървесина ще бъде произведена за единица време. Играчите обаче не могат да назначат колкото си искат работници, а са ограничени до 2500 на това ниво. Тук са налице три ограничаващи фактора – добивната сила за работник, максималният брой работници, но и общото население.

Последното възможно действие, което ще представя, е по-скоро насока при разработката на различни игрови елементи, които се занимават с обмен на пари. По-конкретно да се избягват „безкрайни извори на пари“. Най-често това са NPC търговци, които без никакви ограничения купуват каквото им се предложи от играчите и имат безкраен паричен баланс за това, но може и да представляват цели механики. Възможно най-добрият вариант в такива случаи е парите да идват от предварително дефинирана виртуална сметка и на практика такива елементи да са посредник между парите на играчите, а не да са част от

цикъла на вкарани и изкарани от циркулация средства. Така например наличните средства на **NPC** търговци ще се следят и колкото повече предмети купуват играчите, толкова повече други ще могат да продават.

Избягването на „безкрайни извори на пари“ е възможно решение и на проблема с експлойти в икономиката на играта, защото по дефиниция тези експлойти водят до навлизане на голямо количество пари за кратко време. Ако липсват безкрайни източници, то дори няколко души да открият слабост и да се възползват от нея, това става в никакви крайни граници и отнема време. Така има повече време нарушителите да бъдат хванати, а слабостите поправени преди икономиката да пострада сериозно.

Колкото и да се контролират навлизашите в циркулация пари обаче, те рано или късно започват да се натрупват и тяхната покупателна стойност намалява. Затова е необходимо въвеждане на игрови елементи или промяна на вече съществуващи игрови механики, с цел унищожаване на парични единици. Сред често срещаните имплементации на това в MMO игрите са такси и данъци. Те могат да бъдат фиксирана сума, която се плаща за поддръжка на нещо (например данък върху притежаван имот) или процент при обмен на пари между играчи (например при търгуване). Това е изключително мощен инструмент за изкарване на пари, защото може да се прилага практически всеки път, в който се използват пари за нещо. Трябва обаче да се внимава, защото прекалено високи данъци или проценти при таксите, могат да имат негативен ефект върху игровото преживяване и да отблъскват играчите.



Фигура 16 – Продажба на предмет на пазара за играчи в играта *Gladiatus*⁵⁶. Таксата от 60 жълтици е изчислена върху предложената от играча сума.

Друг използван игрови елемент за контролиране на инфлацията са консумативите. Това са специални предмети, които могат да се използват ограничен брой пъти, най-често само веднъж и носят никакво временно предимство. В MMORPG жанра например, това са отвари, стрели, храна, съставки и други. Важното е, тези консумативи да се закупуват

от **NPC** търговци срещу пари за да може непрекъснатото им използване и следователно купуването на нови да изкарва пари от циркулация.

Още един често срещан метод е поставяне на наказания. Това най-общо представлява отнемане на предмети, имущество или директно валути от играчи в резултат на някакъв техен неуспех. Всеки премахнат предмет от играта на практика повишава покупателната стойност на общата валута, защото играчът вече не може да го продаде и допълнително трябва да похарчи пари за да възстанови загубите. Почти всяка MMO игра притежава реализация на този елемент в никаква степен, дори да не е планирано конкретно като част от мерките против инфлация. В някои заглавия, наказанията са много тежки – например в космическото MMO *Eve Online*, където ако играч бъде победен, космическият му кораб се унищожава, заедно с голям процент от инвентара. В други MMO игри загубите зависят от начина на игра – например в играта *The-West*, където ако играчите не приберат спечелените пари в банка ги губят при евентуално умиране на героя.

На практика, разработката на игрови елемент за изваждане на пари от циркулация не е особено предизвикателна. Трудното е да се направи по такъв начин, че играчите да не се чувстват притиснати, ограничени или натоварени от постоянно губене на спечелени

⁵⁶ <https://lobby.gladiatus.gameforge.com/>

средства, затова всяко отнемане на пари трябва да се усеща естествено и логично в контекста на игровата среда. Най-добрият вариант са специфични за всяка ММО игра решения, породени от конкретни нейни характеристики, които едновременно да бъдат интересни за играчите и в същото време да държат инфляцията под контрол.

Наличието на сигурни игрови валути само по себе си помага за разрешаване на част от поставените проблеми⁵⁷, но както вече споменах, често това не е достатъчно. Ако не се обърне внимание на икономическите въпроси „как“ и „за кого“, то може дори стабилни игрови валути да се обезсмислят (смисълът от пари например би бил по-малък, ако всеки играч може сам да произведе всичко в играта). Ако разработката на икономическата организация следва или е подобна на предложения в тази точка подход, то би следвало да е решен в голяма степен първия въпрос от класическите три проблема представени от Samuelson. Знае се кои предмети са консумативи, кои сировини и т.н. На този етап вече трябва да се обърне внимание на произхода и количеството им.

Основния проблем, който разглеждах, е свързан с опасността ключови за играта предмети да свършат. Това най-вече се случва с предмети, които са част от неговото създаване и играчите добиват по някакъв начин. Има няколко подхода за да се избегне такава ситуация. Един от тях е светът да бъде проектиран като „безкраен“, което означава или да се генерират нови и нови територии с времето, или просто да бъде толкова голям, че на практика дори хиляди играчи да добиват непрекъснато предмети, те няма да свършат. Този метод не може да се приложи за всяка ММО игра, но е много подходящ за игри с отворен свят. Така например играта *Minecraft* използва алгоритъм за генериране на нови територии, когато играч навлезе в тях. За да се избегне обаче проблем от другата крайност – прекалено много предмети да навлязат в циркулация, то трябва тяхното количество да се контролира от други игрови механики, например да се добиват трудно, бавно, или просто да се срещат рядко. Друг подход, подходящ за ММО заглавия, в които няма огромен свят, е проектирането на механики за **регенериране** на източниците. Това може да представлява пълно рестартиране на игрови зони след определен период от време например.

Освен предварително създадени с игровата среда, предметите често могат да бъдат и произвеждани от играчите. Този процес може има различно значение и зависи от вида на предмета. Задачата да се проектира подходящ начин за навлизане на предмети в игровата среда, би била по-лесна, ако тези предмети имат сравнима характеристика, тоест получат някаква количествена оценка. Възможен подход е това да се направи два пъти. Един път върху цели категории от предмети, а след това за всеки предмет в тях поотделно. Това улеснява процеса, защото може да се използва игрова логика за определяне на ценови обхват на голям брой предмети наведнъж. По този начин, дори да се допусне грешка при конкретното оценяване на даден предмет, тя не би повлияла особено на цялостната икономическа организация, ако е в границите на ценовия обхват на категорията.

Използвайки такава оценка, и комбинирайки я с други фактори като колко полезен е предметът, дали е консуматив и т.н., много по-сигурно могат да се балансират различните методи за създаване на ново копие на предмета.

Специфично за ММО жанра решение е най-общо формулирано – не всеки играч да има достъп до всеки начин за добиване на предмет. Ако дадени методи са достъпни само за част от играчите, то останалите са принудени да придобиват предмети по достъпните за

⁵⁷ Свързани с икономическото състояние на игровата среда

тях начини, дори да са неефективни. Така се избягва проблема цели игрови механики да бъдат напълно пренебрегвани от играчите. Разделението може да бъде основано на много неща – игрови класове, предишни избори на играчите, прогрес в играта, опит и много други.

Такова разделение е много популярно сред ММО игрите, защото е основано на най-основния принцип на жанра – взаимодействие между играчи. То стимулира играчи от различни групи да работят заедно. Усеща се естествено, въпреки че на практика е ограничение върху играчите. И като допълнителен бонус, този метод за контролиране на производството може сериозно да стимулира търговията между играчи, което пък е идеално място за поставяне на такси за допълнителен контрол над инфлацията сред игровите валути.

В заключение на тази обширна тема за инфлацията ще спомена за последния представен проблем – за кого са предназначени предметите. Тук няма никакво общо решение, защото този проблем се свежда до внимателно дефиниране на характеристиките на предметите и по-конкретно не само от кого, но и в какви ситуации ще се използват предметите. Това е най-вече важно при прякото взаимодействие между играчи. Голяма част от трудностите при дефиниране на характеристики идват от природата на друг основен проблем в ММО жанра – разлики между нови и развити играчи, с което продължавам в следващата точка.

4.7 Възможни решения на проблем 3

Този проблем се занимава с трудностите възникващи от факта, че почти всички ММО игри се играят дълго време – седмици, месеци дори години, което означава, че по едно и също време играта ще играят играчи в различен етап на развитие. Проблемът има два основни момента – взаимодействие между такива играчи, и едновременното взаимодействие с една и съща игрова среда.

За да може изобщо да се разреши този проблем, трябва първо да бъде изградена система, по която да се оценява добре развитието на играчите. В почти всяка ММО игра, това винаги включва система за ниво на играчите. Такива системи обикновено присъждат точки (най-често наричани точки опит) за различни игрови действия, като най-често колкото по-трудно и времеемко е дадено действие, толкова повече принос има към общото ниво на развитие на играча. Системата може да бъде основана и на достигане на конкретни моменти в игровия сюжет.

В ММО игри със силно присъствие на прям конфликт между играчи могат да се използват и статистики на играчите за оценяване на тяхното развитие като максимална щета, точки живот, отключени специални умения, брой войници в армия и много други.



Наличието на такава система за оценка на нивото на развитие дава достъп до редица техники за справяне с проблема. Игрови елементи, които не се използват едновременно от играчите, могат да бъдат проектирани с динамични стойности на техните характеристики. Така например екипировка, оръжия, консумативи дори цели игрови зони ще бъдат приспособени към нивото на играча.

По-трудно е обаче, когато игрови елемент се използва едновременно от повече играчи. Най-често трудностите са при проектирането на такива игрови зони. Едно възможно решение е всяка зона с конкретна функция да има няколко реализации, всяко от които предназначена за конкретен етап от развитието на играчите. Например зона за битка между играч и NPC врагове ще има няколко имплементации, като силата на враговете ще зависи от категорията на зоната. Разбира се това означава, че всяка възможна награда също ще зависи от тази категоризация, защото ако наградите бяха еднакви, то развити играчи нямаше да играят в по-трудните зони.

Фигура 17 – Предмет от MMO играта The-West, който играчите могат да екипират. Предметът носи допълнителни характеристики, които зависят

от нивото на носещия шапката. Освен това, предмета има изискване за игрово ниво и не може да бъде екипирана от играчи, които не са го достигнали. Тези изисквания обаче са за екипиране на предмета. Всеки може да го притежава, продава или разменя.

Този подход е добър, защото допринася и за игровото преживяване – играчите усещат своето развитие по-добре, но носи и опасност от непредвидени ситуации. Възможно е например развит играч да помага на нови играчи в тяхната зона, което може драстично да промени пътя им на развитие (ако достигнат до даден етап без реално да са готови за него). Трябва да се прецени дали такава намеса ще да бъде позволена като игрова механика или бъдат взети мерки.

Разбира се, възможни са и не толкова крайни варианти. Срецано в MMO игрите е и нивото на враговете в дадена зона да бъде динамично и да е някаква комбинация от нивата на играчите (може да бъде спрямо най-ниското ниво, най-високото, някаква осреднена стойност и т.н.).

Най-предизвикателен за разрешаване остава прекият конфликт между играчи. Най-добре е такива конфликти да бъдат напълно невъзможни между играчи с огромна разлика в развитието. Това е факт в почти всяка MMO игра, защото за да има реален шанс за успех нов играч срещу силно развит, то трябва или битката да не зависи от развитието, което е силно непропорционално (освен ако не пасва на игровия сюжет), или развитието да носи толкова малко облаги, че да не си струва. Дефинирането на обхват от игрови нива, с които даден играч може да влиза в конфликт, дава възможност и двете страни да имат реален шанс за успех, дори ако по-развитият има по-голям такъв шанс.

Този обхват трябва да бъде внимателно подбран. Ако е прекалено обширен, то има риск да не постигне желания ефект. Ако пък бъде прекалено тесен, то има риск играчите да останат без възможни противници, особено при по-високите нива, защото е характерно за MMO игрите броят на играчите с високо ниво на развитие да бъде в пъти по-малък.

За да бъде успешно едно ММО, трябва да има постоянен приток от нови играчи, но в същото време да не спира да бъде интересно за по-развитите. Балансът е труден за постигане, но изключително важен. Играта трябва да бъде лесна в началото, за да не отблъска новите потребители, които още я разучават, и да разгръща постепенно своят потенциал пред тях. Високо ниво на трудност обаче не е единственото нещо, което отказва нови потребители от играта. Често дори да има изграден добър баланс, има играчи, които създават нови профили и герои и нарочно ги държат на по-ниско ниво. Използвайки опита си и всякакви ползи от развитите си акаунти, такива играчи предпочитат да влизат в конфликти с нови такива с цел лесна победа. Именно възможни решения на такива лоши практики ще разгледам в следващата част.

4.8 Възможни решения на проблем 4

Този проблем се занимава с недоброжелатели – група от хора, които по някакъв начин вредят на играта. Важно е да се отбележи, че действия, които са вредни в една игра, може и да не са в друга. Например в ролева ММО игра може да има играчи в ролята на разбойници, които ограбват игровото имущество на други потребители. Въпреки това има действия, които са универсално вредни и именно тях извършват недоброжелателите.

Първо ще разгледам вредни действия в комуникационните канали на играта. Поради естеството на ММО, лесно могат да се достигнат огромен брой хора с едно изпращане на съобщение. На първо място, това е желателно да се ограничи. Въпреки че играта се играе от много потребители, прякото взаимодействие (към което спада комуникацията) се случва само с ограничен брой хора едновременно. Следователно комуникационен канал, който достига всички играчи едновременно, носи много повече вреди, отколкото евентуални ползи. Често срещано решение в ММО игрите е играчите да имат достъп до по-ограничени, обвързани с игровата логика канали (например само с играчи намиращи се в същата зона). Друго срещано решение е да съществува един глобален канал, но играчите да виждат само съобщения от намиращи се в определен радиус от тях потребители (става дума за някакво разстояние в игровата среда между аватарите).

Полезно е като цяло да се ограничи достъпът до по-публични канали на нови потребители и те да получават достъп постепенно с времето. Това предпазва от недоброжелатели, които създават нови профили за да избягват евентуални наказания, както и затруднява използването на **ботове** за автоматично изпращане на масови съобщения.

Такива методи за ограничаване на достигането имат няколко функции. Освен че предпазват масовата аудитория, те разбиват цялата комуникация на играта на по-малки части, което прави възможно да се приложи контрол и вече върху самите съобщения. Преди всичко е желателно да има ясно дефинирани правила, които да определят кое е позволено и кое не. На база на тези правила има три основни начина за модериране на разговорите в играта.

Първият представлява разработка на изцяло автономна система за цензура. В най-елементарната си форма, такава система автоматично ще предпазва от често срещани думи и изрази, дефинирани в някакъв черен списък, без да има нужда от намеса на хора. Постсложнни системи използват логика да засичат не само конкретни думи, но и опити за заобикаляне на цензурата⁵⁸. Разбира се, най-добре би било изкуствен интелект да може да

⁵⁸ Като например заместване на букви с подобни визуални символи.

улавя всякакви неподходящи съобщения, независимо дали са прикрити. Това обаче е изключително трудна задача, поради различни езикови феномени като правописни грешки и вариации, фразеологизми, липса на контекст и т.н. Все още няма напълно работеща такава система, но активно се разработват технологии в тази насока [33].

Най-стандартният начин остава ръчно модериране от хора. Те могат да бъдат наети от собственика на играта, но може и да са доброволци. Важно е обаче разработчиците да предоставят на изпълняващите тази функция достатъчно инструменти за да вършат добре своята работа. Модераторите, както най-често се наричат хората, които поддържат чата, трябва да могат да скриват или директно изтриват съобщения от недоброжелатели. Трябва да имат възможността и да поставят наказания, които могат да бъдат от отнемане на правото за ползване на общи комуникационни канали, до пълна забрана за игра. Освен това е полезно да се разработи система за докладване, която може да се използва от всеки играч. Целта на такава система е да поставя под вниманието на модераторите конкретни моменти от хилядите възможни редове чат, които може да са пропуснали, и дава възможност да се модерира и лична кореспонденция (ако играта има такава функционалност)⁵⁹.

Последният често срещан начин за поддържане на чатове представлява предоставяне на някои потребители ограничени права за модериране в конкретни комуникационни канали. Това е възможно в игри, в които играчите могат да участват в затворени групи, които имат частни чатове. Това са например гилдии, кланове, съюзи и т.н. (в зависимост от вида на играта). Идеята е да се предостави възможност на играчи, които са на ръководни позиции в такива групи, да управляват всичките принадлежащи комуникационни канали. Практиката показва, че играчите в ММО игри комуникират много по-често именно в тези затворени чат групи, следователно такава система значително ще улесни модерирането на глобалните такива.

Малко по-труден е обаче контролът над комуникацията между играчи, която се случва извън самата игра. Като цяло няма много какво да се направи, особено по време на разработка на играта. Най-доброто решение е да се проектира достатъчно добра система, следователно играчите да нямат никаква причина да използват друго. Остават обаче всички места, които се използват за да се говори за самата игра, като форуми, социални мрежи и т.н. Добре е играчите да бъдат максимално информирани, че не трябва да споделят лична или игрова информация извън играта.

Все пак има някои допълнителни мерки, които могат да се вземат, за да се ограничат щетите от различни действия на недоброжелатели, дори да се случват извън самата игра. На първо място е добре да се подсигурят акаунтите на играчите, като се спазват всички съвременни препоръки, включително възможността за **двуфакторна автентикация**. Често именно „**хакнати**“ акаунти се използват от недоброжелатели за различни измами, защото изглеждат правдоподобно. Добре е всяко действие на играчите да бъде подробно записано, но още по-важно е да е удобна работата с тези записи, защото огромният брой играчи означава голям брой информация и стотици действия за кратки периоди от време. Бързото установяване на истината в различни ситуации е ключово за възстановяване на загуби на измамени играчи, ако е възможно. Често обаче са замесени реални пари, които няма как да бъдат възстановени. Това е една от причините много ММО игри изрично да

⁵⁹ Добре е и да има система за докладване на самите модератори, за да се избегне евентуална злоупотреба с правата от тяхна страна.

забраняват търговията на игрови активи срещу пари. Такава забрана предпазва много играчи, които не биха рискували да загубят акаунтите си, дори „офертата“ да е добра.

В заключение на проблема с недоброжелателите ще се спра на **DDOS** атаките. Както вече споменах, това е универсален проблем. Характерно е обаче едно усложнение породено от софтуерната природа на MMO игрите. За разлика от един уебсайт, където е необходим значителен трафик за да бъдат причинени щети, то нормалната работа на игрови сървъри може да бъде нарушена много по-лесно. Причината затова е, че включването на един играч към игровата среда често е доста сериозен процес и изисква много ресурс. Ако изпращаните като част от атака пакети симулират играч, то сървърът ще задейства цялата тази логика, като това може да включва не само зареждане на цялата информация за играча, но и зареждане в паметта на частта на света в която се намира този играч. Допълнително усложнение идва от факта, че много от системите за защита срещу стандартни **DDOS** атаки просто не работят, защото не могат да засекат по-малкия трафик.

Възможно решение на този проблем е процесът по свързване към играта да е проектиран така, че автентификацията на всеки играч да е напълно завършена преди каквото и да е задействане на игровая логика. Ако MMO играта е платена, то това може да се извърши предварително и да се използва самата покупка за доказателство, че е реален играч. Като цяло е най-добре целият процес по автентикация да се извърши на отделен сървър, който да играе ролята на **прокси сървър**. На такъв сървър вече могат да се приложат всякакви стандартни проверки и защити и дори да има достатъчно мощна атака, която да попречи на процеса и да предотврати играчи да се включват в играта, то вече играещите няма да усетят нищо, което дава ценно време за справяне с атаката с минимални щети.

Независимо от вида на проектираното MMO, то задължително ще има проблеми с играчите. Трудно може да се управлява такава игра без да има активен екип от хора, които постоянно да оказват съдействие на играчите. И въпреки че реално проблемите ще настъпят дълго след етапа на разработка, то трябва да се имат предвид още от началото, за да може максимално да се затрудни работата на недоброжелателите.

4.9 Обобщение на социалните проблеми при проектирането на MMO видео игри

В тази глава представих проблеми, които са най-вече породени от основната характеристика на MMO жанра – огромния брой хора. Някои от тях са актуални от самото начало, но други могат да се проявят години след като играта вече е пусната на пазара. Ако разработчиците не са наясно с възможните усложнения по време на разработка, то справянето с тези проблеми, когато възникнат, може да се окаже невъзможно без да се налагат огромни компромиси или загуби за играчите.

Освен конкретните предложени решения за всеки проблем, е изключително важно спазването на стандартни практики за поддръжката на каквото и да е софтуер, най-вече често създаване на резервни копия на цялото състояние на игровата среда. Подробното и хронологично записване на всяко събитие в играта помага за по-лесно откриване на слабости и начини за справяне с тях. А това е изключително важно, защото всяка слабост в игровата логика задължително ще се прояви рано или късно.

Без значение колко е тествана една игра преди да бъде пусната, винаги ще има някаква логическа вратичка, бъг или слабост, която ще остане незабелязана. Поне докато играта не попадне в ръцете на играчите. В момента, в който играта е сред хората и най-малката пукнатина в кода ще бъде моментално залята от рояци опитни играчи, които ще я използват за своя изгода. – Списание PC Gamer⁶⁰

5 Избор на софтуерни технологии за реализация на MMO видео игри

Изборът на подходяща софтуерна технология е изключително важен за реализациите на една MMO видео игра. От този избор зависи какви ресурси ще бъдат необходими, и колко време ще отнеме процесът на разработка. Като всеки друг тип софтуер, най-подходящи са технологии, които са специално разработени за конкретната цел, като днеско време има голям избор от такива, предназначени за разработка на видео игри. Възниква въпросът до колко тези технологии (предназначени за стандартни видео игри) са подходящи и за представители на MMO жанра. В тази глава ще да дам отговор на този въпрос, като първо представя как се реализират повечето видео игри и с какво MMO се откроява, а след това дам конкретни примери. На края ще представя играта Minecraft и ще аргументирам защо е подходяща за анализ на MMO жанра.

5.1 Обобщен поглед върху процеса на разработка на видео игри

Изработката на видео игри като цяло обхваща целия процес от първоначалната концепция до пускането на пазара. Същинската работа се извършва от разработчик, който може да бъде както едно лице, така и цяло студио с много програмисти, художници, писатели, музиканти и т.н. Целият процес обикновено се спонсорира от отделна фирма - разпространител, която отговаря и за самото достигане на готовата игра до хората. Играйте, които са с голям бюджет и имат големи фирми разпространител зад тях, се наричат неофициално „AAA игри“ (тройна-А игра). Контрастно на тях са игри, които са с по-малък бюджет и брой разработчици (като често студиото разработчик и разпространител съвпадат), и които се наричат „инди“ заглавия (съкратено от английската дума *Independent*, независим). Изработката често отнема години, особено при AAA игри.

Формалният процес е разделен на няколко основни фази (действия). Поради установената практика в игровата индустрия разработката да бъде отделена от внедряването, то често работата по дадено заглавие започва доста преди играта дори да е одобрена. Това обхваща най-вече първата фаза, която е фаза на дизайн. Тази дейност се фокусира върху основната идея на играта и по време на нея се изработват първоначалните дизайни⁶¹. Извършват се проучвания, извлечение и анализ на изисквания както и тяхното валидиране. Целта на тази фаза е да произведе ясна документация, която дефинира основните задачи и включва първоначална оценка на необходимите ресурси и времеви график на работата. В тази фаза могат и да се създадат груби софтуерни прототипи, които да показват основни игрови елементи. С помощта на произведените в тази фаза материали от студиото разработчик,

⁶⁰ <https://www.pcgamer.com/whats-the-biggest-game-exploit-youll-admit-to-using/>

⁶¹ Те биват документи описващи игрови елементи, рисунки или друг вид визуално представяне на играта, които няма да се използват в самата игра, а само служат за скици.

концепцията на играта се презентира пред фирмата, която ще я разпространява (ако двете съвпадат, то играта се представя на вътрешното ръководство). Ако играта получи одобрение, започва същинската фаза на разработка.

Видео игрите са специален тип софтуер, който включва не само проектираните игрови елементи, но и аудио елементи, визуален стил, анимации, сюжет. Поради тази причина, често разработчиците се отдалечават от стандартните методологии за разработка, като оправдават това с твърдението, че проектирането на игри е по-скоро изкуство, отколкото наука. Реалността обаче е такава, че игри, които се проектират без добре изграден план е много вероятно да надвишат бюджета си или да са пълни с бъгове и слабости в логиката. Затова е много важно да се спазват съвременните парадигми за разработка на софтуер и още в началото да има ясен план, който да обхваща всеки елемент от играта.

В най-общия случай не са много подходящи модели на софтуерен процес, които са основани на последователен подход на разработка като например **каскадния модел**. Такива модели могат да се използват само за конкретни задачи, които имат ясно дефинирани спецификации. Фактът, че по време на разработка се появяват нови изисквания постоянно, означава, че най-правилно е следването на някоя гъвкава методология. Подходящо е например адаптивното разработване на софтуер, което се фокусира върху проблемите при големи и сложни системи, каквито са видео игрите. Този подход стимулира постъпково и итеративно разработване с използване на прототип, като целта му е да осигури рамка с насоки, които са достатъчни, за да не позволяят на проекта да бъде хаотичен, но не твърде много, което би потиснало творчеството⁶². Това е много полезно, защото освен същинското писане на код, цялостната фаза на разработка на игра включва и създаването на множество игрови активи, които са изцяло творчески по природа – звукови елементи и музика, визуално представяне на елементи, аватари, чудовища, NPC-та, сгради, природа и т.н.

Фазата завършва с тестване на играта. Както споменах няколко пъти в предходни глави, тестването на игровия софтуер е много по-сложно от стандартното. Трябва да се тества не само дали играта работи, но и дали функционално всички игрови елементи си взаимодействват правилно, както и дали е успяла в една от основните си задачи – да бъде забавна за игра. Това означава, че основен фокус по време на тестване е възприятието на крайния потребител. Това се случва поетапно – в началото се тестват само отделни нива на играта, за да се получи обратна връзка относно контролите, визуалния стил, сложността и други. А по-късно вече се тества играта като цяло, включително и как функционира при продължителни сесии.

Финалната фаза започва от момента, в който играта попадне в ръцете на играчите. В началото на 21 век, когато игрите вече са световен феномен, но интернет все още не достигнал своя пик, игрите се разпространяват посредством физически копия, докато днес това се случва предимно дигитално. Тази разлика се отразява и върху бъдещата поддръжка на играта. Независимо от това колко е тествана преди това и колко внимателно е работено върху нея, играчите задължително ще попаднат на бъгове, които разработчиците трябва да коригират. Такива корекции върху кода на играта се разпространяват чрез „пачове“ (от англ. *patch* - кръпка), които представляват софтуер, който актуализира кода на играта. Преди е било по-трудно тези „кръпки“ да достигат до играчите, докато днес това става почти веднага и автоматично, което улеснява изключително много поддръжката на играта.

⁶² „Подходи и методи за реализация на софтуерни системи“, 81 стр. [34]

Разгледаните дотук фази до голяма степен важат и за ММО игрите. Основните разлики са две. На първо място, аспектът за множествена игра, който разгледах подробно в предходни глави, значително усложнява техническата част на разработка. Всичките допълнителни проблеми, пред които се изправят разработчиците, увеличават времето и разходите значително. Другата разлика е по време на внедряването. За разлика от една игра, която има само режим за самостоятелна игра и на теория може да бъде приключена работата върху нея след като достигне до играчите, то при ММО работата тепърва започва. Трябва да се осигурят игрови сървъри, да се коригират бъгове възможно най-бързо, особено ако са критични, да се осигурят бързи канали за обратна връзка. Някои от решенията на специфични за ММО проблеми изискват и съществуването на екип, който да следи случващото се в играта. Всичко това прави ММО игрите вероятно най-скъпия жанр за реализация. Въпреки това, жанрът е във възход, защото трудностите се компенсират от на практика безкрайните възможности, които една такава игра може да предостави на играчите. [34] [35] [36]

5.2 Игрови двигател

В началото на разработката на всяка игра стои важното решение какви технологии ще бъдат използвани. При повечето ММО игри, това означава да се избере подходящ игрови двигател. Игровият двигател е софтуерна рамка, която е предназначена за разработката на видео игри. Обикновено включва в себе си всички необходими библиотеки и помощен софтуер⁶³. На практика двигателят определя как ще изглежда и как играчите ще вземат и взаимодействат с игровата среда. Основните характеристики, които са от значение за вземане на подходящо решение, са:

1. Какво ще представлява игровата среда (дали действието ще се развива в отворен свят или на закрито, дали ще бъде триизмерен или двуизмерен)
2. Каква ще бъде гледната точка на играчите (дали ще е от първо или трето лице)
3. Какви действия ще могат да извършват играчите и колко игровият свят ще се влияе от тях

Разработчиците имат две възможности – да използват готов игрови двигател или да създадат нов. Когато се използва готов двигател, то се спестява голяма част от работата, но за сметка на това се ограничават възможностите за игрови елементи до това, което е заложено от създателите на двигателя. Точно обратното важи при варианта за създаване на нов двигател – разработчиците имат пълна свобода да изградят архитектура, която ще позволява всички планирани игрови елементи, но това отнема време и има много повече бъгове в крайната игра. В повечето случаи, най-правилният вариант е да се използва готов двигател, който евентуално да се модифицира при нужда. Популярните игрови двигатели днес, са проектирани с много функционалности и са подходящи за различни типове игри и затова голяма част от новите заглавия използват именно такива. Главният изпълнителен директор на игровия двигател Unity дори твърди, че над половината от новите игри използват този двигател⁶⁴.

⁶³ Valencia-Garcia, Rafael; et al. (2016). [Technologies and Innovation: Second International Conference, CITI 2016, Guayaquil, Ecuador, November 23-25, 2016. Proceedings](#). ISBN 9783319480244

⁶⁴ <https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/>

Всичко споменато досега важи и за игри, които спадат към ММО жанра, но с едно много важно уточнение. Както вече споменах в предходна глава, за проектирането на почти всички ММО се използва „Клиент-Сървърна“ тип архитектура. Следователно, за разлика от игрите, които имат само режим за самостоятелна игра, то при разработката на ММО има на практика два елемента – самата игра, с която играчите взаимодействат и виждат, и сървърната част, която управлява игровата логика. Игровият двигател отговаря най-вече за клиента, като е препоръчително е да се използва двигател, който предоставя необходимата функционалност за мрежова игра, и то да има капацитета за огромен брой играчи. Сървърът може да се разработи с която и да е технология, но е желателно да се използват двигатели, с които двете (и клиента, и сървъра) се проектират едновременно.

Когато сървърът се проектира отделно от клиента, могат да се използват и игрови двигатели, които не са конкретно предназначени за мрежова игра, но предоставят функционалността за изпращане и получаване на информация. При този вариант обаче трябва да се внимава повече, защото няма рамка, която да ограничава евентуални грешки, и е възможно ключови събития да се „слушват“ и обработват при самите играчи, което е огромна слабост в сигурността на играта и улеснява нейното манипулиране. Освен това е възможно при по-голямо събиране на играчи на едно място, клиента да не може да се справи технически с визуализацията, което води до лаг или пълен срив.

Крайният резултат трябва да бъде еднакъв, независимо какъв игрови двигател или технологии се използват. Необходимо е всички играчи да възприемат еднаква игрова среда и да могат да взаимодействат с нея и помежду си по подреден и синхронизиран начин. Поради увеличения интерес в последните години към мрежова игра, днес има голям избор от технологии, които могат да се използват за разработката на ММО видео игри. Сега ще се спра по-подробно на някои от тях.

5.3 Двигателят Hero (HeroEngine)

HeroEngine е не само игрови двигател, а цялостна платформа за проектиране и управление на ММО видео игри. Основната цел на проекта е да опости максимално техническата част по създаване и интегриране, за да може разработчиците да се фокусират максимално върху игровите елементи, балансиране и цялостното игрово преживяване. Първоначално двигателят е създаден от студиото *Simutronics* за ММО играта *Hero's Journey* (също разработвана от тях). Самата игра се бави с години и през 2008 официално е спряна, а разработеният двигател, заедно с играта и цялата планирана облачна инфраструктура, са пуснати за продажба. Купувач е организацията *Idea Fabrik*, която освен да придобие интелектуалните права върху гореспоменатите технологии, наема и повечето от оригиналните разработчици, за да продължат работа по двигателя.

Самият двигател се отклоява с това, че за разлика от повечето подобни технологии, могат да се правят промени по игровия свят докато играта се играе от играчи. Много елементи от добавяне на сгради и декори до оправяне на различни бъгове може да се случи в реално време, без да е нужно изграждане на нови версии и разпространяване на промени с пачове. Това прави не само поддръжката много по-лесна, но и улеснява процеса на разработка, защото няколко души могат едновременно да работят върху игровия свят, като всеки вижда в реално време работата на другите.

Днес платформата предлага всичко необходимо за създаването на една MMO игра. Освен лиценз за работа с движителя, се предоставят и хостинг решения за готовите игрови сървъри, както и всичко свързано с поддръжката им. Това го прави изключително подходящ избор за малки екипи с ограничен бюджет. Работата с HeroEngine се извършва най-вече с *HeroBlade* – специален софтуер, който представлява набор от инструменти за създаване на виртуални светове. Тези инструменти включват всичко необходимо за изграждане на игровата среда – от топографията на терена, до природа, сгради, декор и персонажи.

Различните обекти, които населяват игровата среда, могат да бъдат напълно интерактивни благодарение на Smart Objects функцията на движителя. Играчите могат да взаимодействат с тях, като това става синхронизирано, следователно всички виждат един и същ ефект от действията на някой. HeroEngine има и система за управление на изкуствения интелект на съществата обитаващи игровия свят. Тази система е проектирана конкретно за нуждите на MMO жанра и също идва с набор от инструменти, с които да се постигне всяко желано поведение.

Освен всичките вградени функции, движителят дава възможност и да се използват външни модули за изменение на различни аспекти, например физиката на игровата среда. Това, заедно с поддръжката на скриптове, прави HeroEngine изключително гъвкав и на практика възможностите му нямат граници, но с условието, че всяко отклонение от стандартните модули усложнява работата по играта.

Едни от най-популярните MMORPG заглавия използват HeroEngine. Това включва игри като *Faxion Online*, *The Repopulation* и *Star Wars: The Old Republic*. Още едно популярно заглавие, все още в алфа фаза на тестване, използва движителя – *Exile Online*, а *The Elder Scrolls Online* използва собствен движител, който обаче заимства много елементи от HeroEngine, който е използван първоначално. [\[37\]](#) [\[38\]](#) [\[39\]](#)

5.4 Движателя Unity

Unity е едновременно и игрови движител, и интегрирана среда за разработка. Основната му задача е да създава триизмерна, интерактивна среда, която освен графики, има ясно дефинирана физика, звуци и правила за взаимодействие. Първата версия – 1.0.0 е създадена от David Helgason, Joachim Ante и Nicholas Francis в Дания. Излиза на 6 юни 2005 година, като основната цел на проекта е да предостави леснодостъпни професионални инструменти за разработка на видео игри, с което да даде повече възможности на по-дребните и независими игрови разработчици. Първоначално, софтуерът за работа с Unity е достъпен само на **Mac OS X**, а самите крайни продукти могат да се изпълняват на ограничен брой платформи. По-късно се появява версия и за **Windows** и значително се увеличава броят на поддържаните изходни платформи. Към днешна дата, най-актуалната LTS версия е 2020 LTS. [\[40\]](#)

Работата със софтуера за разработка на Unity е улеснена възможно най-много. Има няколко основни концепции, които се използват за създаване на игровата среда. На първо място са „сцени“. Сцените са обект, който съдържа всички игрови елементи в една игра или част от нея. Цялата игра може да се изгради с помощта на единствена сцена, но по-сложните игри

имат отделна такава за всяко ниво, зона, среда и т.н. като няма ограничение на броя сцени. Създаването на нова сцена е лесно, като освен празна (без никакви игрови елементи), тя може да бъде създадена с помощта на шаблони.

Самите елементи, които насяват всяка сцена, се наричат игрови обекти. Това е една от най-важните концепции в Unity. Всеки обект в играта е игрови обект – от елементи на пейзажа, през герои, създания, до елементи управляващи осветяването на сцената, гледната точка и специалните ефекти. Тези фундаментални обекти сами по себе си не правят нищо, защото са на практика просто кутии за различни „компоненти“. Като основна част от всеки игрови обект, компонентите са това, което добавя конкретна функционалност на игровия обект, към който принадлежат. Unity има голям брой вградени компоненти, но могат да се създават и нови с помощта на **скриптове**. Всеки игрови обект задължително притежава компонент, който определя точната му позиция и ориентация, но освен това няма други ограничения.

Графичните функционалности на Unity дават пълен контрол върху вида на разработваната игра, като могат да се използват за създаване на красиви, но и оптимизирани за различните платформи, графики. Възможни са всякакви модификации, за да се достигне желаният вид на игровата среда. Unity помага и за симулиране на физика в играта, като дава възможност за работа със сблъсъци, гравитация и други подобни сили. Има предоставени различни имплементации, като всяка е предназначена за различни среди (например за 2D или 3D). Вграденият физичен двигател е популярния **Nvidia PhysX⁶⁵**.

Звуковите функционалности на Unity също са доста обширни. Предоставена е възможност за работа с 3D пространствени звуци, които играчите не само ще могат да чуват, а и да усещат посоката и разстоянието им. Интегрираната среда може да работи с почти всички използвани аудио файлове, като при изграждане на крайния продукт, всички звукови активи се **компресират**, което помага за намаляване на размера на играта.

Unity предлага още много възможности, като е достъпен и специален магазин за споделяне на безплатни и платени разширения и активи, създавани от други хора. За ММО жанра обаче е важно играта да може да комуникира с други инстанции. В началото на своето съществуване, такъв режим не е бил във фокуса на Unity. Днес обаче е достъпен експериментален модул за мрежова игра, който все още е в процес на активна разработка. В основата на този модул е библиотеката **Netcode**, която е създадена специално за Unity. Благодарение на тази библиотека, разработчиците на ММО игри не е нужно да се занимават с основни мрежови протоколи или мрежови рамки, поради нивото на абстракция, което тя предлага.

⁶⁵ <https://github.com/NVIDIAGameWorks/PhysX>

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ServerBehaviour : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}

```

Фигура 18 - Шаблон за елементарен ServerBehaviour C# скрипт, с който може да се създаде игрови сървър и да се дефинират основни функционалности

Netcode по дефиниция създава клиент-сървърна архитектура. Това означава, че ще има сървърната инстанция, която на практика ще управлява какво се случва, а всички останали свързани клиенти ще показват възможно най-близко копие на игровата среда, което играчите ще възприемат. Тази синхронизация се случва, като се дефинират тези игрови елементи, които променят състоянието си поради пряко или непряко действие на играч.

Това се прави много лесно с Unity. За да се реплицира състоянието на един обект по цялата изградена мрежа например, е достатъчно да се добави нов компонент към него – NetworkObject component. Наличието на такъв компонент позволява пълен контрол над обекта в мрежовата среда с помощта на допълнителни компоненти като NetworkTransform, NetworkBehaviour и NetworkVariable. Всеки мрежов обект е притежаван от конкретен клиент или игровия сървър. Това подобрява сигурността и позволява по-ясна структура на игровата среда.

Netcode има също така вграден подход за управление на физиката в играта, който позволява симулациите да се случват на игровия сървър (това се управлява с NetworkRigidbody компонент). По този начин резултатните позиции на даден обект след действия на сили се синхронизират еднакво сред всички клиенти без смущения. Това дава възможност и да се обединяват действията на много играчи за получаване на някакъв резултат.

Въпреки че Unity позволява разработка на MMO игри, все още не е сред най-използваните двигатели за това. Непрекъснатото развитие на мрежовият модул обаче показва огромен потенциал и вероятно в следващите години именно този двигател ще бъде сред най-използваните в MMO жанра. Популярни MMO заглавия, които използват Unity, включват *FusionFall*, *Albion Online* и *Temtem*. [\[41\]](#) [\[42\]](#)

5.5 Двигателя Unreal (UnrealEngine)

UnrealEngine е вероятно най-реалистичният игрови двигател за 3D виртуални среди. Както при повечето подобни технологии, е предоставен пълен комплект от инструменти за разработка, с който всеки може да се възползва от най-големите новости в 3D реализма. Създаден е за играта *Unreal* през 1998 година, предназначен основно за *FPS* игри⁶⁶. С времето възможностите му се разрастват, като днес е възможно да се проектира всеки жанр, а освен това поради високото ниво на реализъм, двигателя се използва активно и в телевизията и киното⁶⁷.

Версията на двигателя са разделени на поколения, като съвсем скоро излиза официално версия от пето поколение – Unreal Engine 5. Написан е на C++ и изходният му код е отворен, което позволява по-лесно модифициране за конкретни нужди. На пръв поглед, високият реализъм изглежда като пречка за MMO жанра, защото огромен брой играчи на едно място означава много движещи се елементи, което по принцип затруднява дори модерните хардуери. Unreal обаче е изключително оптимизиран и са възможни огромни събирания на играчи без особен проблем. Пример за това е играта *Fortnite*. Самата тя не е представител на MMO жанра, но успешно събира до 100 души на една аrena, което би било достатъчно за случаите на пряко взаимодействие при повечето MMO игри.

Разработката на игри с Unreal следва същите принципи като при останалите разгледани двигатели, като игровите елементи следват сравнително пристрастна йерархия. Всеки обект се дефинира с *Actor* (от англ. „актьор“) клас. Самият клас е по-скоро като контейнер, към който могат да се прилагат компоненти, които реално добавят някаква функционалност на обекта. За динамични игрови елементи като аватара на играча или NPC персонажи се използва *Pawn* (от англ. „пешка“) клас, който наследява *Actor*, но добавя функционалност за управление. Обекти, представяни от *Actor* и *Pawn* класове, се групират в различни игрови зони, наречени нива, а всички нива изграждат игровия свят.

Както вече споменах, самият двигател първоначално не е предназначен конкретно за разработка на MMO игри, но има добавена допълнително такава функционалност. Мрежовата игра и при тази технология е основана на клиент-сървърна архитектура, като проектирането на системата е сравнително лесно. За да се реплицира състоянието на даден обект по цялата мрежа например, е достатъчно това да се дефинира в *Actor* класа, като се използва вградената променлива „*bReplicates*“. За да се спести обема на информация, която трябва да се споделя между играчите и сървъра, някои основни функции като издаден звук, анимации и други подобни, не могат да се реплицират, но ако променливите, от които тези основни функции зависят, са репликирани правилно, то всички играчи биха наблюдавали един и същ резултат.

Основните инструменти за разработка могат да се допълват с помощта на разширения, като една такава приставка например добавя възможността няколко души да проектират виртуалната среда едновременно и всеки да вижда работата на другите в реално време, което улеснява много процеса и комуникацията между разработчиците.

⁶⁶ Архив на новинарска статия - <https://web.archive.org/web/20000411020741/http://pc.ign.com/news/4661.html>

⁶⁷ <https://www.unrealengine.com/en-US/blog/forging-new-paths-for-filmmakers-on-the-mandalorian>

Сред ММО игрите, които използват този двигател, се открояват *Lineage II*, *Lost Ark*, *Marvel Heroes* и *DC Universe Online*. [43]

5.6 Други технологии за реализация на ММО игри

До този момент разгледах няколко конкретни технологии за създаване на видео игри. Едната специално предназначена за ММО жанра, а другите с добавена в последствие такава функционалност. Както вече споменах, възможно е да се създаде ММО игра без да се използват готови игрови двигатели, но това означава, че разработчиците трябва да създават собствен такъв за клиента, а освен това да проектират сървъра и цялата комуникация на най-ниско ниво. За тази цел може да се използва почти всеки популярен програмен език днес, като в зависимост от конкретни спецификации на играта някои ще бъдат по-подходящи от други.

Интересен случай представлява браузър ММО жанрът. Той е специфичен, защото играчите не играят играта със специално създаден софтуер, а чрез своите браузъри. Следователно се спестява голяма част от работа, която по принцип е нужна за проектиране на игрови двигател като физика и графики, а разработчиците се фокусират само върху интерфейса, чрез който играчите ще си взаимодействат. Въпреки че е възможно да имат по-сложни графики⁶⁸, за повечето игри от този жанр е характерно статично представяне на игровия свят, пълно с менюта с информация, статистики и всичко необходимо, за да може играчите да знаят какво се случва. Това означава, че една ММО игра може да се разработи дори с помощта на стандартни уеб технологии като HTML⁶⁹, JavaScript⁷⁰ и CSS⁷¹.

Браузър ММО игрите много добре показват, че няма стандартна технология за проектиране на ММО и са възможни всякакви комбинации, в зависимост от нуждите или възможностите на програмистите. Това ще демонстрирам с помощта на играта **Minecraft**. Въпреки че тя не е технология предназначена за реализация на ММО, начинът, по който е проектирана, много наподобява игрови двигатели. Въпреки ограничения контрол откъм графики, физика и други игрови ресурси, играта може да се използва, за да се създаде напълно уникално ММО преживяване, поради възможността да се добавят напълно нови функционалности с помощта на разширения. Това я прави много удобна за анализ на проблемите при проектирането на едно ММО.

5.7 Играта Minecraft

Minecraft е приключенска видеоигра, която се разработва от шведското студио *Mojang*. Към днешна дата е налична за почти всички популярни платформи и е една от най-известните игри с над 200 милиона продадени копия и близо 126 милиона активни играчи⁷².

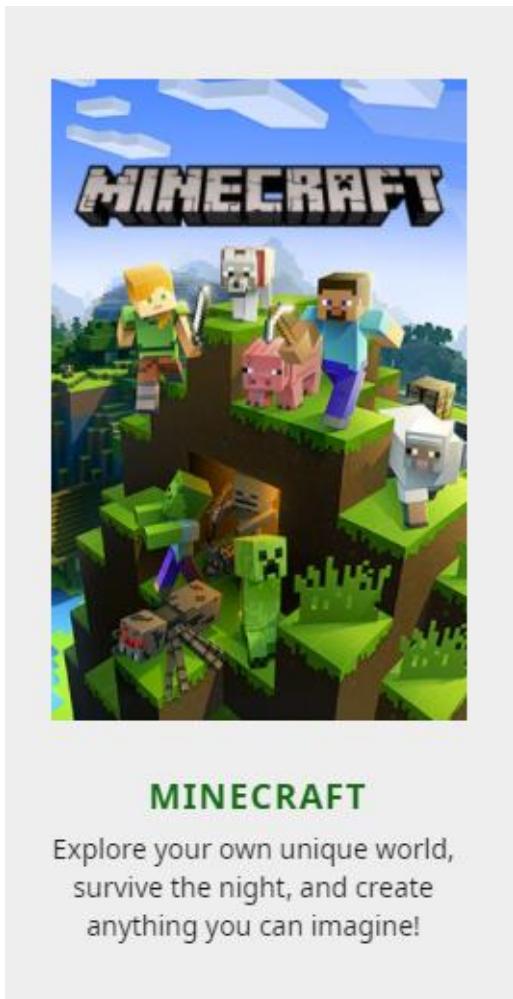
⁶⁸ С помощта на технологии като WebGL например - https://www.html5rocks.com/en/tutorials/webgl/webgl_fundamentals/

⁶⁹ <https://html.spec.whatwg.org/multipage/>

⁷⁰ <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

⁷¹ <https://developer.mozilla.org/en-US/docs/Web/CSS>

⁷² По данни на TheVerge [44] и Statista.com [45]



Фигура 19 - Официалното описание на играта. В превод: "Изследвай свой собствен уникатен свят, оцелей през нощта и създай всичко, което можеш да си представиш!"

Играта може да се разглежда като представител на няколко жанра, като преди всичко е т.н. „пясъчник“ (от англ. Sandbox). За игрите от този жанр е характерно, че те нямат ясно дефинирани начини за постигане на прогрес или извършване на някакви игрови цели. Играчите изследват един блоков, триизмерен свят, който е технически безкраен и наподобява дивата природа. Може да се взаимодейства с почти всичко в тази среда – от терена до съществата, които го обитават. Въпреки огромните възможности за игра в самостоятелен режим, в основата на популярността на Minecraft е режимът за игра с други хора – *Multiplayer* (от англ. Много играчи). За разлика от повечето такива игри обаче, тук няма предоставени и поддържани от разработчиците официални сървъри за игра⁷³. Това е оставено изцяло в ръцете на Minecraft общността.

Благодарение на бесплатно предоставения от Mojang сървърен софтуер, всеки може да създаде своя визия на играта. С помощта на свободно разпространявани библиотеки, които улесняват модификацията на сървърния софтуер, играта може да се променя по различни начини и да се добавят всякакви нови функционалности.

Играта е представител и на MMO, защото технологията ѝ позволява огромен брой хора да си взаимодействат непряко чрез игровата среда. Разбира се, се играчите могат да взаимодействат и пряко, като на теория това също няма никакво

ограничение, но от хардуерна гледна точка не повече от 200-300 души могат да се намират на една сървърна инстанция по едно и също време, което е предостатъчно.

Преди да пристъпя към практически експеримент с играта, ще направя обзор на нейната история. След това ще се спра по-подробно върху основните игрови елементи, а на края ще разгледам технологията на играта и кои от поставените проблеми тя разрешава без нужда от допълнителни модификации.

5.7.1 Кратка история на играта

Началото на Minecraft е през пролетта на 2009 година⁷⁴. Само за около седмица, Markus „Notch“ Persson създава и тества на Java най-основните механики на играта⁷⁵. Това включва

⁷³ Това изключва услугата Realms, която е хостинг услуга предлагана от „Mojang“ и само за Minecraft сървъри.

⁷⁴ Wired.co.uk [46]

⁷⁵ През тази седмица играта дори носи друго име – „Cave Game“ [47]

основни блокчета, поведението им и силно опростени варианти на игрови аватари. През следващите месеци на разработка, които днес носят определението „Класически период“⁷⁶, се въвеждат още много от характерните за играта елементи, включително и поддръжка за мрежова игра. Още в тази фаза на разработка, Minecraft вече е достъпна за тестване от играчи и то срещу цена - 13 долара. [49] [50]

Въпреки сравнително бавното развитие поради факта, че се разработва в свободното време на Notch, играта започва да трупа интерес. Към средата на 2010 година Notch спира работа по други проекти и се фокусира изцяло върху Minecraft. Малко след това учредява студиото „Mojang AB“ и разширява своя екип. Въпреки че все още не е излязла официално, Minecraft получава титлата „Игра на годината“ през 2010 от списание PC Gamer. Първата пълна версия (1.0.0) излиза на 18 ноември 2011 година, по време на събитието MINECON⁷⁷. [51] [52]



Фигура 20 - За да се отпразнува 10 годишнината на играта, пренаписана на JavaScript версия от "Класическия период" е достъпна чрез браузър на <https://classic.minecraft.net/>.

Версия 1.0.0 е последната, върху която Notch работи като главен разработчик. Всички бъдещи версии се разработват от екип, ръководен от Jens Bergensten (Jeb)⁷⁸. Следващите няколко години играта претърпява огромно развитие. Получава по един значителен ъпдейт на всеки няколко месеца. Активното добавяне на нови елементи задържат интереса на старите играчи и привличат нови. Към средата на 2013 година се достигат 10 милиона продадени копия, като само за 1 година тази цифра достига до 54 милиона. [53][54]

Датата 15 септември 2014 година е повратна точка в историята на играта. Срещу внушителната сума от 2.5 милиарда долара софтуерният гигант Microsoft купува Minecraft, заедно с цялото студио Mojang. Създателят Notch окончателно напуска компанията и вече не участва по никакъв начин в творческия процес. Въпреки притеснения сред общността, че това ще доведе до края на играта, тя продължава да расте по популярност. [55]

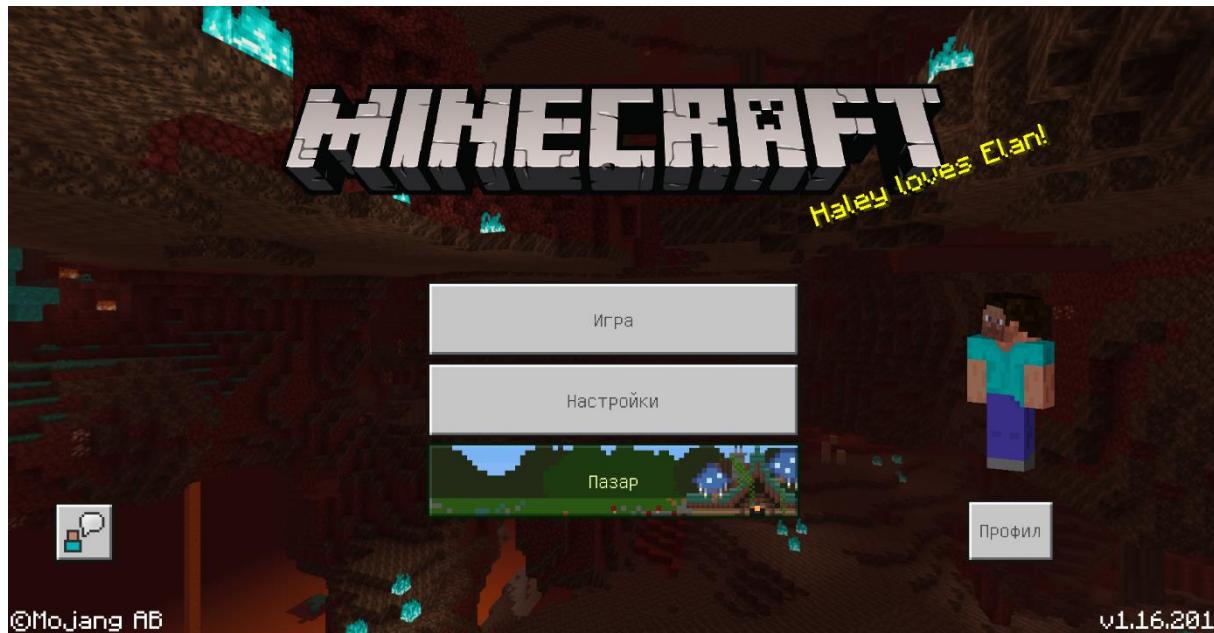
⁷⁶ За първи път се среща в „Блогът на Notch“ [48]

⁷⁷ MINECON е ежегодно събиране на фенове и разработчици на играта Minecraft. Първата версия на играта е представена на финала на MINECON 2011.

⁷⁸ Това е обявено публично в „Блогът на Notch“ на 2ри Декември 2011 година. [48]

Макар по-рядко от преди, не спират да излизат значителни версии. На MINECON 2017 за пръв път се провежда масово гласуване сред фенове на играта, което да определи един от фокусите на следващия ъпдейт. Така играчите пряко избират едно от чудовищата във версия 1.13. Тази традиция продължава и до днес. [56]

През годините играта навлиза и при мобилни устройства, игрови конзоли и дори при системи за виртуална реалност. Цената за закупуване на Minecraft днес е различна за различните устройства, като за оригиналната Java версия е €23.95.



Фигура 21 – Начален екран на Minecraft for Windows 10. Днес са обособени две различни имплементации на Minecraft – оригиналната Java версия и т.н. Bedrock версия, която е написана на C++ и е предназначена за игра на мобилни устройства, конзоли и Windows 10.

5.7.2 Как се играе играта

Действието се развива в дивата природа. Играчите започват без никакви предмети и екипировка. Какво следва, зависи от избрания режим на игра, като са налични четири. В режим „Оцеляване“ играчите трябва да събират ресурси, с чиято помощ да оцелеят възможно най-дълго време, което означава най-вече да поддържат точките си живот. Режим „Приключенски“ е почти същия като предишния, но с допълнителни ограничения, които правят играта по-реалистична, но и по-трудна. В режим „Творчески“ играчите нямат точки живот и имат достъп до безкрайно количество от всеки ресурс в играта, като целта е да развихрат творчеството си (най-вече строене на сгради, паркове, села, градове). Последният режим „Наблюдател“ не предлага игрово преживяване, а се използва за изследване на игровия свят без да може да взаимодейства с него по никакъв начин (отново в пълна безопасност).

„Оцеляване“ е най-играният режим. Всяка нова игра обикновено започва със събиране на дървесина и камък, с чиято помощ се изработват сечива. С тях се добиват по-здрави материали, от които се произвеждат по-добри инструменти и всякаква необходима екипировка и оръжия. В това време, трябва да се обръща внимание и на други нужди като събиране на храна, липсата на която води до негативни последици, а в зависимост от избраното ниво на трудност може да доведе и до загуба на точки живот. Основна цел в

играта е да бъде победен митичен дракон. За да се постигне тази цел, играчите е необходимо да извършат куп действия и да обиколят три различни измерения, като през цялото време не спират да се занимават и със своето оцеляване. Игровата среда е пълна с приятелски, неутрални и враждебни създания, с които играчите могат да се бият, сприятеляват, отглеждат и т.н.

Въпреки стандартната поредица от събития и цели, които описах по-горе, Minecraft е с толкова високо ниво на свобода, че всяка игра е на практика уникална. Началото е коренно различно, ако играч стартира в пустиня, където няма достъп до дървесина например. Освен това, нищо не кара играчите да преследват основната цел. Те могат дори да не ходят при дракона, ако желаят, а победата му не води до приключване на играта. Всичко това прави Minecraft не просто някаква история с начало, среда и край, а един отворен, жив свят. Това я прави идеална за MMO среда.

Стандартната мрежова игра не променя играта по никакъв начин⁷⁹. Не се появява нов дракон за всеки играч например, нито някой се води главен герой. Мрежовата игра просто позволява много играчи да съществуват и взаимодействат в един и същ свят. Това е и причината Minecraft да бъде MMO – нейната технология позволява всеки да взаимодейства с огромен брой хора. В стандартния си вид обаче, тя не е много добра MMO игра, защото не е обърнато внимание на всички проблеми на жанра, особено социалните. Именно затова е идеална за анализ, защото лесно ще се проследят ефектите след прилагане на различни решения на MMO проблеми.

5.7.3 Разрешени проблеми в играта Minecraft

Преди да изброя неразрешените проблеми, първо ще направя анализ на тези, на които разработчиците са обърнали внимание. Цялата архитектура на играта е основана на „клиент-сървър“ тип архитектура. Това включва и играта в самостоятелен режим. Когато се стартира нова игра, клиентът сам инициализира сървър, към който играча се свързва автоматично. Мрежовата игра е толкова дълбоко заложена в играта, че за да се превключи от самостоятелна игра, в LAN режим е достатъчно да се натисне бутона в менюто.

Тази архитектура е в основата на доста от предложените в главата за технически трудности решения. На първо място стои проблемът с обмяна на информация между играчите. Този проблем се занимава с това как всички играчи да възприемат една и съща среда. Припомням, че за да се случи това е необходимо обмяната на информация да бъде достатъчно бързо, без загуби и без възможност за манипулация.

Почти всичко, което се случва в играта, се случва на сървъра, като всеки свързан клиент съобщава за действията на своя играч. Тази комуникация използва стандартен **TCP** мрежов протокол и се изпращат и получават малки пакети с информация. Този избор не е най оптимален откъм скорост, но гарантира последователност на информацията и сигурност на връзката. За да се гарантира висока скорост, разработчиците са заложили на решението тази информация да бъде възможно най-малко. Това е постигнато, като част от действията, които сървърът извършва, се изчисляват и при локалната игра от клиента, като на края само се синхронизират резултатите. Това се важи най-вече за визуални събития и по този начин не се позволява сериозна манипулация от играча. По този начин драстично се намалява обема информация, който трябва да се сподели.

⁷⁹ Освен някои технически промени, за които ще спомена малко по-късно.

Друг основен технически проблем, който Minecraft разрешава, е свързан с взаимодействието между играчите. Това взаимодействие бива пряко и непряко. Трудностите с прякото взаимодействие са в голяма степен разрешени от направените до този момент избори, свързани с обмяната на информация. Вътрешният игрови часовник от 20 цикъла в секунда би бил неподходящ за игра от типа на стрелец от първо лице, но е напълно достатъчна за Minecraft, където битките се случват с мечове, брадви и лъкове. По този начин, всичко се случва достатъчно бързо, че играта да върви в реално време, но не прекалено много, което допълнително намалява количеството информация.

Непрякото взаимодействие се случва посредством игровия свят. Той трябва да бъде достатъчно голям, че да не ограничава ММО играта. За да се постигне това, светът не съществува винаги, а се генерира с времето. Когато играч пътува до зона за първи път, тя се създава с помощта на специален алгоритъм и се съхранява в паметта, заедно с промените, които той е направил. По този начин, всеки следващ играч, който посети същото място, ще види ефектите от предишната игра на друг. За да се избегнат технически проблеми със синхронизацията, когато много играчи се намират на едно и също място, тази зона се зарежда в оперативната памет, а крайното състояние се съхранява чак след като всички играчи напуснат мястото.

На практика повечето технически проблеми са разрешени от разработчиците. Изключение прави проблемът с интеграцията на игрови сървър, защото Mojang не поддържат официални сървъри на играта. По тази причина не е обърнато и никакво внимание на повечето социални проблеми.

5.7.4 Неразрешени проблеми в играта Minecraft

Преди всичко стои проблемът с ресурси, необходими за логическото изпълнение на всичко случващо се в играта. Колкото повече играчи са в един свят, толкова повече части от него са заредени в оперативната памет. Освен това, броят същества, които обитават средата, зависи от броя на играчите. Всяко живо нещо в играта притежава изкуствен интелект, следователно процесорът на машината, която обработва случващото се, ще започне да изостава. В такива случаи, играта започва да обработва по-дълго всеки цикъл. Ако вътрешният часовник падне под 17 цикъла в секунда, играчите започват да усещат проблеми. Затова една от основните задачи в следващата глава е именно оптимизация на игровия сървър, която да позволи достатъчен брой хора да играят едновременно.

Разработчиците не обръщат много внимание на повечето експлойти. Голяма част от тях дори с времето се превръщат в игрови елементи от типа „не е бъг, а е функционалност“. Липсва какъвто и да е икономически баланс. Възможни са ферми, които безкрайно произвеждат предмети без необходимост от сировини. Няма виртуална парична система, а единствената валута, която се използва за търговия с някои от NPC персонажите, е небалансирана и може да се произвежда безкрайно. Това допълнително се усложнява от факта, че всеки играч може да прави абсолютно всичко, без класови или други ограничения.

Един от социалните проблеми, който донякъде е разрешен, е свързан със стари и нови играчи. Казвам донякъде, защото от една страна един нов играч няма абсолютно никакъв шанс срещу друг, който носи силни брони и притежава различни консумативи, но от друга, ако се изключи екипировката, няма абсолютно никаква разлика между тях. На практика

всеки може да екипира и най-силните брони и оръжия, дори току що да е започнал игра. Следователно, разликата между тях е по-скоро в това до какви предмети имат достъп.

Не е обърнато особено внимание и на недоброжелателите. Въпреки че играта е донякъде сигурна откъм манипулации, поради начина, по който е проектирана, не са изключени различни „трикове“ за придобиване на нечестно предимство. Пропуски във валидацията на движение например, позволяват разработката на модифициран клиентски софтуер, който да позволява недоброжелателите да се движат и да извършват действия много по-бързо, да летят, да се бият от разстояние и т.н. Освен това, почти не е обърнато внимание на комуникацията. Наличен е един общ чат канал, но не са взети никакви мерки против недоброжелатели в него. Всеки може да изпраща колкото и каквото иска съобщения. Предоставена е единствено система за поставяне на пълна забрана за свързване към сървъра, но тя може да се постави само ръчно от администратор. Следователно още една от основните задачи в следващата глава, ще бъде проектиране на цялостна система срещу недоброжелатели.

5.7.5 Задачи по проектиране и експериментално тестване на сървър на играта Minecraft

След като показвах защо играта е подходяща за анализ и изясних необходимата работа е време да пристъпя към изграждане на Minecraft сървър. За целта първо ще представя какво представлява той.

Ще покажа технически как се стартира сървър на Minecraft. Ще представя по-подробно какво представлява официалния сървърен софтуер и как се използва. Ще покажа как се прилагат модификации и как се конфигурират. Когато това е изпълнено, ще имам всичко необходимо да извърша практически експеримент с готовия игрови сървър и реални играчи.

По време на експеримента ще анализирам предложени в тази дипломна работа решения на ММО проблеми. Ще обърна внимание на сигурността и различни експлойти. Ще изградя основите на стабилна икономика, като дефинирам виртуална валута, как тя ще се генерира и използва и ще изградя система за справяне с недоброжелатели, която ще се управлява от екип от доброволци.

6 Проектиране на решението - игрови сървър на играта Minecraft

6.1 Изграждане на Minecraft сървър

Създаването на нов Minecraft сървър е сравнително лесна задача. Разработчиците на играта са предоставили сървърен софтуер под формата на изпълним jar файл, който може да бъде стартиран под всяка възможност, която има Jvm. Файлът се тегли безплатно и без необходимост от никакъв вид регистрация от сайта на играта⁸⁰.

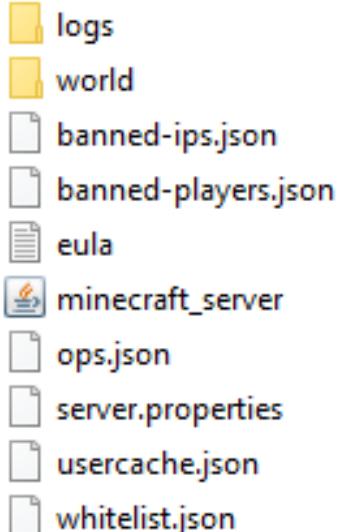
6.1.1 Стартиране на „Vanilla“ сървър

С помощта на стандартен за използваната операционна система команден интерфейс, сървърът може да се стартира със следната команда:

```
java -Xmx2048M -Xms2048M -jar minecraft_server.jar
```

Тук Xmx и Xms са Jvm параметри, които казват на виртуалната машина колко оперативна памет да задели за сървъра, а minecraft_server.jar е името на изпълнимия jar файл.

При първото стартиране се създава файлът **eula.txt**, в който трябва да се даде съгласие с **EULA** (End User License Agreement – потребителско лицензионно споразумение). След като се приемат условията (**eula=true**), сървърът трябва да се стартира отново, като вече автоматично се генерираят всички останали необходими файлове.



Ако всичко е успешно, в командния интерфейс (който вече играе ролята на конзола за управление на сървъра) трябва да изпише:

```
[Server thread/INFO]: Done (8.055s)! For help, type "help".
```

Сървърът вече работи и може да се достъпи и чрез клиентски софтуер.

Фигура 22 – Всички необходими файлове за нов сървър.

Създаден по този начин, игра в създадения сървър по нищо не се различава от игра в самостоятелен режим, освен че вече могат да се присъединяват много играчи. Затова се нарича “vanilla” (изчистен, без добавки) сървър. Така стартиран обаче, почти не може да се модифицира. На практика са достъпни само набор от опции, намиращи се в **server.properties**, които са предимно технически настройки. Няма предоставени средства

⁸⁰ <https://www.minecraft.net/en-us/download/server>

за лесно модифициране на самата игра. Дори кога зад тях не е достъпен публично. Възможно решение е да се **декомпилира** кода на сървърния софтуер и така да се правят промени. Това е доста грубо решение. Освен че при всяка промяна трябва да се компилира и разпространява наново целият проект, този подход може да доведе до куп грешки и бъгове. Най-добрият вариант да се правят промени би бил с помощта на някакво API, с което лесно да се модифицира основният сървърен софтуер.

6.1.2 Библиотеката Bukkit (Bukkit API)

Желаещите да управляват Minecraft сървъри се сблъскват с този проблем почти веднага след официалното пускане на играта. Това води до началото на **Bukkit**. Този софтуерен проект е безплатен и с отворен код, като основната му цел е да предостави възможност за модифициране на играта без да е необходимо да се променят нито клиентският софтуер, нито сървърният jar. Това е постигнато с помощта на създадената **Bukkit API** библиотека. Тя предоставя набор от методи, които всеки желаещ програмист може да използва за да прави промени по играта. Тези модификации се пакетират в отделен jar файл (наречен **плъгин** или приставка), който специално модифициран за Bukkit сървърен софтуер прилага автоматично всеки път, когато се стартира сървър. Това допълнително ниво на абстракция между кода на играта и създадените модификации, прави изключително лесно добавянето и мащабирането на различни модификации.

Освен огромното улеснение, работата с Bukkit API гарантира, че няма да се правят промени, които да причинят проблеми с комуникацията между сървъра и клиента. Това, разбира се, ограничава възможните модификации. Не е възможно например да се добавят изцяло нови предмети, блокчета, звуци, същества. Ограниченията обаче са нищо на фона на хилядите други възможности за модификация, затова Bukkit е най-използваната библиотека за това и до днес.

За разработчиците, които все пак искат да имат пълен контрол над играта и да могат да променят абсолютно всичко, са налични подобни на Bukkit проекти, които позволяват това. Такива проекти използват същата основна логика – предоставят модифициран сървърен софтуер, който не се променя, а работи с допълнителни jar файлове, които съдържат модификациите. Разликата тук е, че е необходимо играчите също да използват модифициран клиент, заедно с jar модификациите. За целите на тази дипломна работа не са необходими промени, които изискват модифициране и на клиентския софтуер, затова ще продължа с библиотеката Bukkit.

През 2014 проектът Bukkit получава **DMCA** (Digital Millennium Copyright Act – вид нарушение на авторско право) и на практика спира да се разпространява⁸¹. За няколко месеца бъдещето на десетки хиляди сървъри е под въпрос. За щастие, друг проект наречен **Spigot**, който е директно „fork“ разширение на Bukkit, успява да заобиколи законовите забрани като предоставя начин всеки сам да си компилира модифицирания сървърен софтуер. Това спасява Minecraft общността, защото като разширение на Bukkit API, всички създадени първоначално приставки работят без проблем и с модифицирания от Spigot сървърен софтуер (с много малко изключения).

За целите на тази дипломна работа, аз ще използвам друг „fork“ (разширение) – **Paper**. Този проект въобще не е разширение на Bukkit, а на Spigot. По същата причина като по-рано,

⁸¹ The day open source died - <https://blog.jwf.io/2020/04/open-source-minecraft-bukkit-gpl/>

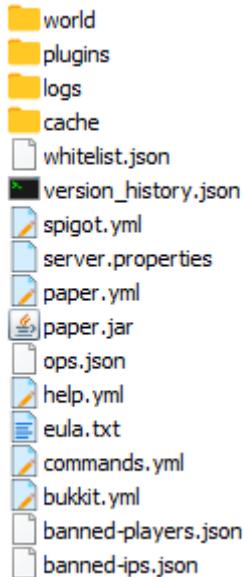
всички приставки за Bukkit (и тези, които използват Spigot API) работят и с модифицирания от Paper сървърен софтуер. Към днешна дата са активни десетки проекти, които имплементират в себе си Bukkit и Spigot, но добавят редица подобрения, промени или разширения. По-малко използвани са и напълно отделни от Bukkit библиотеки като **Sponge API**⁸² и **Glowstone API**⁸³, но аз избрах да работя с Paper, защото са направени редица подобрения и оптимизации, които водят до значително увеличение на производителността. Освен това постоянно получава актуализации, доброволци предоставят безплатна помощ при нужда и остава достъпа до хилядите приставки за Bukkit и Spigot.

6.1.3 Стартране на Paper сървър

Стъпките за стартиране на Paper сървър съвпадат със стъпките при “Vanilla”. Единствената разлика, разбира се, е да се използва предоставения от Paper модифициран сървърен софтуер. Той отново е под формата на jar файл, който се стартира по същия начин като официалния:

1. Файлът се тегли безплатно от сайта на Paper⁸⁴.
2. Стартира се един път за да се създаде текстовият файл eula.txt, чийто условия трябва да се приемат.
3. Стартира се втори път за да се създадат останалите файлове.

Ако всичко е успешно, в конзолата трябва да изпише:



[INFO]: Done (15.260s)! For help, type "help".

Сървърът вече работи и се забелязва, че има повече създадени конфигурационни файлове от „Vanilla“. Засега ще оставя всички настройки с техните стойности по подразбиране.

Важното тук е папката **plugins**. Чрез нея могат да се добавят и премахват всякакви модификации по сървъра без да се променя основният софтуер.

Могат да се правят всякакви промени, които API-то позволява. Тоест всичко, което клиентът очаква от сървъра не трябва да се променя. Например не може да се коригира времето за извършване на дадено действие, което има анимация, защото това, което ще види играчът, няма да съвпада с реалността и ще възникнат проблеми (ще са извън синхрон).

Фигура 23 – Всички необходими файлове за нов Paper сървър.

Освен да имплементира необходимите методи, модифицираният Paper сървърен софтуер прави и редица промени върху работата на различни процеси, като в основата на тези промени е идеята, че голям брой хора ще играят едновременно. Тоест целият проект обръща специално внимание на техническите

⁸² <https://www.spongepowered.org/>

⁸³ <https://glowstone.net/>

⁸⁴ <https://papermc.io/>

проблеми, които възникват от природата на ММО, което го прави още по-подходящ за проектиране на такава среда.

6.1.4 Работа с приставки

Приставките представляват някакъв код написан на Java и пакетиран в jar файл, който обикновено прави промени по сървъра, но не е ограничен до това. За да се приложи дадена приставка е достатъчно да се постави в папката **plugins**. Всеки може да си напише сам нова приставка или да изтегли готова такава. И двата варианта имат своите плюсове и минуси, които са същите като при всеки друг софтуерен проект.

Разработката на приставка отнема време, ресурси и поддръжка, но е напълно ясно какво прави, лесно се правят промени и като цяло има пълен контрол. Това е важно, особено когато трябва напълно да се изпълнят дадени изисквания без никаква възможност за промяна.



Фигура 24 – Налични приставки в spigotmc.org към края на 2020г. Всичките 43896 приставки в категорията Spigot са безплатни, а тези 1055 в категорията Premium са платени.

Да се използват готови приставки спестява време и ресурси, но винаги води до някакъв компромис, който трябва да се направи. Често готова приставка има повече от необходимите функционалности и като цяло не съвпада напълно с изискванията. Хубавото е, че повечето готови приставки идват с конфигурационни файлове, с които могат да се правят фини настройки или напълно да се изключват нежелани функционалности, но това трябва да е заложено предварително от разработчика на приставката. Възможен проблем е и качеството на кода. Сайтовете за разпространение на приставки упражняват някакъв контрол над това какво се качва, но предимно следят за приставки със зловреден код и не обръщат особено внимание на качеството. Лошо написани приставки могат да доведат до проблеми с паметта,

прекомерно натоварване на процесора, грешки, лаг и дори срив на сървъра. Хубавото е, че много от приставките са с отворен код, което дава възможност такива слабости да се коригират.

Приставките могат да работят самостоятелно, но могат и да зависят една от друга – тоест да се допълват, разширяват или просто да си взаимодействват. Paper API дава възможност това да става лесно и сигурно, като дори има система за приоритети на събития, която помага за разрешаване на конфликти. Все пак остава възможно две или повече приставки да се прочат, затова (особено когато се използват готови приставки) трябва да се знае точно какво прави всяка от тях.

Последното важно нещо, което трябва да се отбележи е, че приставките са създадени за конкретна версия на играта. Това не означава автоматично, че такива написани за по-стара версия няма да работят на по-нова. Дали ще има проблеми или не зависи от това с кой елемент на играта се занимават и дали този елемент е претърпял промени между двете версии (тази в която се използва приставката и тази, за която е създадена). Възможно е да се напише приставка, която да поддържа няколко версии на играта, но това значително я усложнява. При готовите приставки обикновено има информация с кои версии на играта работят.

След като представих всичко необходимо и избрах конкретен софтуер, време да пристъпя към проектиране. Използвайки готови и някои създадени от мен приставки, ще прилага част от теоретично разгледаните решения на проблеми при MMO като изградя сървър за играта Minecraft. С помощта на реални играчи, ще проверя дали предложените варианти водят до положителни резултати, като анализирам тяхната игра и състоянието на сървъра.

6.2 Добавяне на важни приставки

Както вече обясних, ще извършвам промени по играта с помощта на приставки. До този момент имам създаден Paper сървър по описаните в предходната глава стъпки и е време да добавя jar файлове в папката **plugins**. Създаденият сървър е на версия **1.16** от играта, което означава, че всяка модификация трябва да може да работи с тази API версия. Ще започна с приставки, които обръщат внимание на най-често срещаните MMO елементи – комуникация и икономика. Освен това, ще е необходимо да се проектира добра система за правомощия, с която по-лесно да се следи ролята на всеки в играта. Minecraft има съвсем елементарна такава, която разделя потребителите на играчи и сървърни оператори, но тя далеч не е достатъчна за целите на експеримента.

Преди да започна да инсталирам приставки, които добавят тези елементи, трябва да обърна внимание на един проблем. Всяка добавена приставка, която трябва да използва някой от новодобавените елементи като например виртуалната валута, трябва да знае как реално е проектирана тя (тоест да зависи от приставката за икономика). Paper API позволява това да става лесно от гледна точка на самата разработка, но това значително затруднява комбинирането на едни приставки с други, защото всяка нова такава трябва или да имплементира от нулата всичко необходимо, или да идва с голям списък от необходими зависимости. При положение, че има десетки хиляди приставки, това би било непрактично.

Този проблем се разрешава благодарение на „**Vault**“ (от англ. „трезор“). Това е приставка, която е приета за стандарт и представлява API за тези ключови и необходими за MMO игра елементи – икономика, правомощия и комуникация. Самият Vault не реализира тези неща, а работи като посредник. Тоест създава едно допълнително ниво на абстракция, което позволява приставки да работят заедно без реално да знайт една за друга. По този начин, ако някой иска да добави използване на виртуална валута в своята приставка, вместо да търси конкретна реализация от друг проект или да я прави сам, използва Vault. Така приставката му ще работи със практически всяка, която добавя икономика (и също използва Vault API).

6.2.1 Приставки за икономика, комуникация и правомощия

Ще започна с Vault⁸⁵. Като основна приставка може да се намери на няколко места, но ще го взема от каталога на Spigot ([spigotmc.org](https://www.spigotmc.org/resources/vault.34315/)). Използвам версия **1.7.3**. Поставям изтегления jar в папката **plugins** и рестартирам сървъра. В конзолата изписва:

```
[Vault] Enabled Version 1.7.3-b131,
```

⁸⁵ <https://www.spigotmc.org/resources/vault.34315/>

което означава, че е инсталирана успешно. Генерира се и конфигурационен файл, но той има само един ред, с който може да се изключи проверката за нова версия при всяко стартиране на сървъра.

Ще продължа с проектиране на система за правомощия (permissions system), защото почти всяка друга приставка се нуждае от такава за да работи правилно. Правомощията (permissions, от англ. - разрешения) дават възможност да се контролира нивото на достъп на всеки участник в сървъра до различни игрови елементи. Всяко разрешение представлява някакъв String (поредица от букви и цифри, които са разделени на групи с точка), който принадлежи на дадена функционалност, и с който се контролира достъпът до нея.

Проверка какви правомощия има даден играч, може да се случва практически при всяко събитие като резултата от нея променя изхода му. Неслучайно използвам термина събитие, защото тези правомощия не се използват само за достъп до нещо (например команда или функция), а влияят и върху самата игра.

Основният дизайн на системата за правомощия идва още от ранните дни на играта, като дори има останал от тогава вграден вариант за разпределение на правомощия в Paper чрез конфигурационен файл, който обаче няма да използвам. Разрешенията се разпределят на ниво потребител или група. Всеки потребител може да получи дадено разрешение пряко или като го наследи от група, към която принадлежи. Наличието на групи е изключително важно за експеримента, защото дава възможност наведнъж да се управляват правомощията на огромен брой хора, какъвто има в ММО.

Ако не се използва такава система, то най-често всички така наречени „админ“ правомощия на дадена приставка (команди, които са предназначени за управление и конфигурация) се дават по подразбиране на потребителите с роля на сървърен оператор, а останалите правомощия ги получава всеки. Това донякъде предпазва от злоупотреби, но много ограничава и практически липсва контрол над това кой какво ниво на достъп има.

За да проектирам достатъчно добра система, ще използвам приставката за правомощия **LuckPerms⁸⁶** версия 5. Съществуват алтернативи, но избирам тази, защото се актуализира редовно и предлага високо ниво на контрол. С помощта на огромен брой функции, тя позволява създаване на сложна система от правомощия, с която ще имам пълен контрол над събитията в сървъра.

Поставям jar файла в папката plugins и рестартирам сървъра.

[LuckPerms] Successfully enabled. (took 2721ms)

За разлика от Vault тук имам доста възможности за конфигурация, като се започне от вида на базата данни и се стигне до фундаментални начини на работа на самата система.

Информацията, която приставката има нужда да съхранява, е най-вече информация за потребителите – например какви конкретни правомощия има всеки играч, в кои групи е поставен, колко групи има създадени, какви правомощия има всяка група и т.н. За база с данни ще използвам тази по подразбиране – локална H2 database, която се съхранява в

⁸⁶ <https://luckperms.net/>

plugins/LuckPerms. Няма нужда да се прави нищо повече – приставката си създава, настройва и управлява сама базата.

Повечето правомощия ще разпределя по групи, а играчите ще получават необходимото ниво на достъп чрез групите, в които членуват. Започвам с основната група „**default**“. В нея ще поставя всички правомощия, които са необходими на нов играч да играе в сървъра. Тази група е специална и всеки нов потребител или такъв без група автоматично ще бъде прибавен в нея. Тук ще вметна, че ако авторите на дадена приставка използват стандартната система за правомощия, то играчите, които не са сървърни оператори, ще получат правомощия, които авторите са преценили, че всеки трябва да има. С помощта на LuckPerms обаче, ще изключа тази функционалност за да имам пълен контрол над това какви действия могат да се извършват по подразбиране.

Освен групата „**default**“ сега трябва да създам други групи, които най-общо представляват обединение на общи точки на развитие на потребителите. Ако развитието беше линейно, то това би било сравнително лесна задача (всяка следваща група наследява правомощията от предишната и играчите се „изкачват“ по време на своята игра). В игровия свят, който изграждам, всеки играч обаче трябва да може сам да избира как да играе и неговото развитие няма да бъде напълно линейно. Това може да доведе до нежелани ситуации. Ако например дадено действие на играч води до отнемане на някаква функционалност, то има проблем – като се премахне от групата, ще загуби и други правомощия. Поради нивото на свобода на действия би било невъзможно да се преценят предварително всички възможни комбинации.

LuckPerms има решение на този проблем – всяко правомощие има прикачена стойност „**true**“ или „**false**“. Така на практика има три състояния за всяко разрешение – липсата му, или наличие на една от двете стойности (макар че често липса на правомощие и стойност „**false**“ водят до един и същ резултат). Така обаче има друг проблем – какво се случва, ако играч е в две групи като в едната има дадено разрешение със стойност „**true**“, а в другата същото със стойност „**false**“?

За да избегне този проблем, LuckPerms използва системата за тежест. На всяка група се начислява естествено число, което показва с какъв приоритет е тя. Основната идея е, че при наличие на конфликт се взема стойността на групата с по-висока тежест. Така вместо да се следят стотици правомощия, е достатъчно да се подредят групите по тежест и така се разрешават повечето възможни конфликти.

Използвайки функционалностите на LuckPerms ще създам няколко основни групи, а вече при необходимост (особено с добавянето на нови приставки) ще добавям нови. Ще дам пример с групи, които съдържат правомощия за управление на сървъра. Първата група е „**Модератор**“. Създава се с командата:

```
> lp creategroup moderator
```

Основната задача на потребителите в тази група ще бъде да упражняват контрол над системата за комуникация (чата) в сървъра и да помагат на нови потребители да се ориентират, ако има нужда. Сега ще и добавя тежест. Добра практика е да се добавят тежести не с последователни стойности, а през някакъв интервал. Така е много по-лесно на по-късен етап да се добави група между две последователни (иначе ще трябва да се променят тежестите на всички групи). На тази група ще сложа тежест 100 с командата:

```
> lp group moderator setweight 100
```

Групата е готова да получава правомощия и да се добавят към нея потребители. По същия начин ще създам две други групи – „Админ“ и „Оператор“ (да не се бърка с ролята на сървърен оператор от самата игра):

```
> lp creategroup admin  
> lp creategroup operator  
> lp group admin setweight 200  
> lp group operator setweight 300
```

Потребителите в групата Админ ще упражняват контрол над цялостната игра в сървъра. Те ще могат да ограничават достъпа на конкретни играчи, да налагат наказания, да променят света, да организират състезания и да дават награди. Тяхна задача ще бъде и да откриват проблеми, които ако могат да разрешат. Всичко това се случва в самият сървър, тоест те нямат достъп до бекенда.

Групата Оператор планирам да е с най-високо ниво на достъп. Потребителите в нея отново нямат достъп до сървърните файлове, но имат достъп до команди и функции, с които могат ефективно да променят ключови настройки по сървъра.

Тези три групи се занимават с поддръжката на играта в сървъра и мога да си позволя да ги разглеждам линейно – всяка следваща на практика надгражда предишната, защото играч, който може да променя конфигурации (оператор) трябва да може и да налага наказания (админ). Вместо да добавям повтарящи се правомощия във всяка група ще използвам наследяване:

```
> lp group admin parent add moderator  
> lp group operator parent add admin
```

По този начин, групата админ наследява всички правомощия от модератор, а групата оператор от админ. Наследяването е транзитивно, следователно оператор съдържа и всички правомощия на модератор.

При наличието на ясна структура, LuckPerms предоставя възможност да се създаде „пътечка“ (от англ. track). С нейна помощ по-лесно се управляват подредени линейно групи:

```
> lp createtrack staff  
> lp track staff append moderator  
> lp track staff append admin  
> lp track staff append operator
```

```
[14:35:05 INFO]: [LP] moderator ---> admin ---> operator
```

Основните групи за поддръжка са създадени и подредени в пътечката „**staff**“. Сега ще добавя и правомощия. На този етап има само две инсталирани приставки, затова единствено ще добавя правомощия към групата оператор, за самият LuckPerms. Това ще позволи управление на системата за правомощия от самата игра без да е необходим достъп до конзолата. Поддържат се заместващи знаци (wildcards):

```
> lp group operator permission set luckperms.*
```

Вече всеки добавен потребител в групата оператор, ще има достъп до всички команди на LuckPerms. На този етап това разпределение е достатъчно, защото още няма други приставки. Има набор от правомощия, които идват от основния сървърен софтуер, но няма да използвам повечето от тях, защото са свързани с функционалности, за които ще използвам допълнителни приставки (които идват с техни си правомощия). Ще отбележа, че за нормална игра (наподобяваща самостоятелна игра, „vanilla“) не са необходими никакви правомощия.

След като положих основите на система за правомощия, следват комуникация и икономика. Ще използвам приставка, която имплементира и двете. Това е **EssentialsX**⁸⁷ версия **2.19**. Това е на практика цял проект, който предлага различни модули всеки с отделен jar. Има модули за комуникация, защити на игровата среда, интеграция с платформата за комуникация *Discord* и други. За експеримента ще използвам три от модулите – основния, който е задължителен във всеки случай и имплементира наред с други неща виртуална валута, модула за комуникация, който ще позволи допълнителен контрол над чатовете и модул за централна точка (на това ще се спра по-късно).

След като добавих трите jar файла и рестартирах сървъра, се генерираха няколко конфигурационни файла. Най-важен е **config.yml** в папката *plugins/Essentials*. Този гигантски конфигурационен файл, с почти хиляда реда настройки и обяснения, съдържа важни решения, които ще определят как играчите ще извършват основни действия в играта. Засега ще обърна внимание само на настройки в категорията „Икономика“ и „Комуникация“. Играчите ще започват играта без налични от виртуалната валута средства, а общият брой на циркулиращи пари ще оставя без лимит (по-скоро ограничен от Java double) за да не се нарушат резултатите от експеримента. Ще оставя глобален чат канал, въпреки че има опция да се ограничи с радиус в играта и ще форматирам имената на играчите, които използват чата, да съдържа информация за групите и развитието им.

Следват правомощията. С тяхна помош ще дам на играчите достъп до команди и функционалности, с които ще могат да комуникират по-добре и с които виртуалната валута реално ще може да се използва. Например команда за разплащане между играчи:

```
/pay <player> <amount>,
```

която се добавя с правомощието *essentials.pay*. За да се управляват по-добре правомощията, особено когато има много приставки на по-късен етап, ще създам нова група в LP (LuckPerms), която ще съдържа всички правомощия на Essentials, до които абсолютно всеки играч трябва да има достъп. За да покажа, а и тествам, че всичко с правомощията до този момент работи, ще създам новата група от самата игра, а не чрез конзолата.

```
/lp creategroup essentials_default_permissions
```

След което ще настроя основната група default да наследява всички правомощия:

```
/lp group default parent add essentials_default_permissions
```

Без да влизам в подробности, ще добавя в новата група правомощия за основни команди за разплащане, за изпращане и получаване на виртуална поща, за лична чат комуникация и за помощни команди.

⁸⁷ <https://essentialsx.net/>

На този етап са поставени добри основи за бъдещо проектиране на ММО средата. Вече има виртуална валута, която играчите могат да използват за търговия помежду си и подходящи комуникационни канали. Все още не съм обърнал внимание на самите проблеми, които ще анализiram, а само подгответ необходими инструменти за справяне с тях. Преди да продължа със същинските решения, ще обърна внимание на други основни приставки.

6.2.2 Приставки за допълнителна сигурност и управление

Благодарение на избора на сървърен софтуер, голяма част от слабостите са коригирани, но не всички. Затова ще добавя приставката **IllegalStack⁸⁸** версия **2.2.0**. Нейна цел е да обърне внимание на повечето познати бъгове. Благодарение на нея, ще се коригират няколко вида експлойт, включително и против копиране на предмети, което непряко помага за управляване на инфлацията по-късно. Всяка защита може да се изключи от конфигурационния файл при нужда.

След това ще добавя две използвани от стотици приставки API библиотеки. Първата е **ProtocolLib⁸⁹** версия **4.6**. Както споменах по-рано, Bukkit API ограничава възможните модификации. С помощта на ProtocolLib API са възможни допълнителни действия, които модифицират не самата игра, а пакетите с информация, с които сървърът и клиентът си комуникират. Другата библиотека е **PlaceholderAPI⁹⁰** версия **2.10**. Тя имплементира общоприет стандарт за визуализация на информация от различните приставки чрез заместители (*placeholders*). Освен това, с нейна помощ е възможно да се обменя информация между приставки без те да зависят един от друг.

След това ще инсталирам приставка, която добавя функционалност за създаване на повече игрови светове. Както споменах в предишната глава, част от стандартната игра включва посещаване на общо три измерения – основния свят, който наподобява дивата природа, подземен свят пълен с лава и чудовища и мястото, на което живее дракона. Всяко от тези измерения съществува в отделен игрови свят, което означава че е невъзможно да се стигне пеша от един свят до друг, а само с портали. С помощта на приставки за управление на повече светове вече е възможно да се създадат нови такива, което е много полезно за една ММО игра. Различните светове могат да се използват за разделение на играчи, за състезания и т.н. Има няколко популярни мениджъри на светове, но аз ще използвам **Multiverse⁹¹** версия **4.2**. Причината за този избор е наличието на много разширения и активната поддръжка.

На края ще добавя приставка, която започва като модификация на клиента, защото се използва за **тераформиране** на игровата среда. Това се случва изцяло чрез клиента, в реално време и докато играчи играят играта, което наподобява функция на HeroEngine. Ще бъде необходима, когато се строят основни инфраструктурни сгради и локации. Нарича се **WorldEdit⁹²** и задължително трябва да използвам версия **7.2** предназначена за версията на основния сървърен софтуер.

От всички добавени в тази точка приставки е необходимо да разгледам правомощията единствено на WorldEdit, защото с него ще се работи от самата игра. Тази приставка е

⁸⁸ <https://www.spigotmc.org/resources/dupe-fixes-illegal-stack-remover.44411/>

⁸⁹ <https://github.com/dmulloy2/ProtocolLib>

⁹⁰ <https://github.com/PlaceholderAPI>

⁹¹ <https://dev.bukkit.org/projects/multiverse-core>

⁹² <https://enginehub.org/worldedit/>

много важна за работата на администраторите, затова директно прилагам всички правомощия:

```
/lp group admin permission set worldedit.*
```

На този етап са инсталирани осем приставки, но само една се занимава конкретно с проблемите на ММО (IllegalStack). Останалите приставки обаче предоставят ключови инструменти за реалното имплементиране на решения. Затова сега ще продължа именно с това.

6.3 Имплементиране на решения на проблема за изграждане на игровата среда

Този проблем се занимаваше предимно с експлойти. Благодарение на използваната приставка (IllegalStack) и Paper сървърен софтуер, на практика са коригирани много от техническите проблеми. Все още обаче представляват опасност АФК експлойтите. Те са проблем в Minecraft, защото могат да се създадат различни ферми, които работят, когато играч е около тях, без да е необходимо да извършва действия.

Ще приложа комбинация от две решения. Първо, ще използвам функция в Essentials, която автоматично ще засича АФК играчи. С нейна помощ, ще се прекъсва връзката им към сървъра, което означава, че техните ферми няма да работят. Важно е да се отбележи, че света продължава да съществува, но самите ферми не произвеждат нищо, ако няма играч около тях. Този играч обаче може да бъде който и да е, не само собственикът на фермата.

В config.yml на Essentials:

```
# After this timeout in seconds, the user will be kicked from the server.  
# essentials.afk.kickexempt node overrides this feature.  
# Set to -1 for no timeout.  
auto-affk-kick: 1500
```

Така на всеки 25 минути бездействие, играчите ще бъдат изхвърлени от играта (ще се прекъсне тяхната връзка). За да не се нарушава работата на екипа, групата модератор като първа в йерархията ще получи правомощие за изключение от това действие:

```
/lp group moderator permission set essentials.afk.kickexempt
```

Вече няма да е възможно трупане на активи с бездействие. След като показвах как правомощията се използват не само за команди, но и за функционалности, вече няма да се спирам допълнително на тях, освен ако не е от голямо значение.

По този начин обаче не е обърнато внимание на опитите за заобикаляне на системата. Както поясних в анализа на този проблем, решението тук зависи силно от вида на ММО играта. В Minecraft има непряко взаимодействие между хиляди, но никога повече от 200-300 души едновременно. Благодарение на този факт, ще разглеждам играчите, които заобикалят проверката, като недоброжелатели и ще приложа решение срещу тях. По-конкретно ще създам списък с официални правила в сървъра. Ще говоря повече за това малко по-късно, когато обръщам специално внимание на недоброжелателите, но е добро решение против

опитите за заобикаляне на АФК, защото за да се натрупа реална полза от това трябва да минат дни или седмици. Това дава предостатъчно време такива нарушители да бъдат забелязани от други хора в сървъра, особено от екипа. Когато това действие се забрани, ще може да се налага наказание при извършването му, което напълно го обезсмисля.

6.4 Имплементиране на решения на проблеми с икономиката

За проектиране на икономиката ще следвам подхода, който предложих в главата с анализ. С използването на готовата игра Minecraft вече са създадени предметите в играта. С инсталирането на Essentials вече имам и виртуална валута. Остава да обърна внимание на организацията вече в контекста на готовите пари. В този експеримент ще анализирам не подхода с предварително създаване на всички пари, а този, при който те се печелят по време на игра. Следователно основният проблем е поддържане на баланса между навлизачи и излизачи от циркулация пари и контролиране на инфлацията.

Първата задача е да предоставя начини за печелене на пари. Има хиляди варианти, като избрах за основен метод да бъде извършването на различни действия в играта под формата на професии. Идеята е, че за някои от действията, които играчите могат да извършват в играта, ще се печелят пари в зависимост от резултата или условията. За целта ще използвам приставката **Jobs**⁹³. Тя предоставя интерфейс за проектиране на професии като дава възможност да се използват почти всички действия в играта – чупене и поставяне на блокчета, градинарство, развърждане на животни, битка с играчи или чудовища и много други.

Проектирах 6 професии, които са свързани с най-отличителните за играта действия:

1. Миньор – Печели пари, когато открива и добива руди в нормалното измерение.
2. Миньор в подземното царство – Печели пари, когато открива и добива руди в подземното измерение.
3. Дървар – Печели пари, когато отсича дървета и добива дървесина.
4. Рибар – Печели пари, когато лови риба.
5. Ловец – Печели пари, когато лови чудовища и животни.
6. Ловец на ендърмани – Същото като ловец, но само за конкретно чудовище. Тази професия всъщност е създадена за да коригира бъг, с който от това чудовище се печелят лесно пари.

Сега идва трудната част – дефиниране на точните стойности за действията. Това далеч не е тривиална задача. Всяка професия трябва да бъде сравнима с останалите и за да ги сравнявам, ще използвам величината общи приходи за единица време.

За да намеря стойността на тази величина обаче трябва да имам готова професия. Ще започна с професиите „Миньор“ и „Миньор в подземното царство“ (от „Mine“ в заглавието на играта Minecraft – мина, копаене). При тях ще се получават пари за чупене на руди. Трябва обаче да се подберат различни стойности за всяка руда, защото някои са много по-редки от други. За щастие са известни конкретните стойности, които алгоритъмът за генериране на руди използва в играта. С тяхна помош може да се направи оценка на

⁹³ <https://www.spigotmc.org/resources/jobs-reborn.4216/>

рядкостта на всяка руда и така да се постави подходяща цена. Те са описани в следващата таблица:

<i>Руда</i>	<i>Размер на залеж</i>	<i>Брой залежки</i>
Dirt	33	10
Iron	9	20
Redstone	8	8
Lapis	7	1
Gold	9	20
Diamond	8	1
Emerald	1	11
Coal	17	20
Quartz	14	16
Netr. Gold	10	10
Ancient Deb.	3	1

Стойностите са за откъс от картата, с размер 16x16. От значение са размерът на залежа, което показва колко индивидуални блокчета има най-често в един залеж, и брой залежи в отрязъка 16x16. Един залеж представлява група от блокове от една и съща руда, които имат общи страни. В началото на таблицата, съм добавил и обикновена пръст, един от най-обикновените материали, който ще използвам за основа на изчисленията. Дойде момент, в който за да продължа трябва да избира конкретно число. То може да бъде каквото и да е, поради факта, че се проектира виртуален свят с чисто нова валута.

Реших да използвам цената за блокче пръст от друг сървър, което ще ми позволи да направя сравнение по-късно по време на анализа. Следователно, за основа на изчисленията ще използвам **0,38\$**⁹⁴. Това означава, че цената на един залеж пръст ще бъде:

$$\text{Размер на залеж} * 0,38\$ \Rightarrow 33 * 0,38 = 12,54\$$$

Сега мога да изчисля цената за изкопаване на един залеж от останалите руди с формулата:

$$[\text{Брой залежи пръст} / \text{Брой залежи (руда)}] * \text{Цена на залеж пръст}$$

На края, използвайки размера на всеки залеж, ще получа добра оценка за цена на едно блокче от всяка руда:

⁹⁴ За символ на валутата ще използвам символа за долар, което се конфигурира в config.yml на Essentials.

Руда	Цена на залеж	Цена на блок
Iron	6,27	0,70
Redstone	15,68	1,96
Lapis	125,40	17,91
Gold	6,27	0,70
Diamond	125,40	15,68
Emerald	11,40	11,40
Coal	6,27	0,37
Quartz	7,84	0,56
Netr. Gold	12,54	1,25
Ancient Deb.	125,40	41,80

С това, професиите „Миньор“ и „Миньор в подземното царство“ са теоретично готови. Остава да се дефинират във файла **jobConfig.yml** в папката **plugins/Jobs**. След като имам работеща професия, мога да направя оценка на величината приход за единица време. Тя също може да се изведе теоретично като се вземе скоростта на чупене на блокове, но така се пренебрегват фактори като терен, чудовища, човешка реакция. За да бъде по-близко до реалността, реших да проверя експериментално колко пари се изкарват за една минута от професия „Миньор“ и да използвам тази стойност за останалите професии.

Опит №	\$ / мин
1	46,79
2	36,68
3	81,53
4	25,78
5	104,54
6	23,97
7	46,29
8	134,76
9	24,52
10	32,02

След десет сесии (таблицата вляво) установих, че се изкарват по средно **55,69\$** на минута. Поради високото ниво на случайност, което се вижда и по получените стойности, може да се приеме, че се изкарват по средно **1\$** на секунда работа по професията.

Въпреки че съм доволен от тази величина, реших да я потвърдя и теоретично. Това би било много трудна задача, но аз все пак имам стойност, затова мога да използвам силно опростен модел, с който да получа не по-точна сума, а само да валидирам сегашната. За целта използвах скоростта за чупене на един каменен блок, който е най-често срещания блок по време на копаене, **0,1** секунди. Рудите в играта се намират основно под земята, следователно зоната, в която се генерира с размер 16x64x16, където 64 е у височината, на която се намира повърхността. Ако допусна, че цялата тази зона има само камък и руди, то

тя ще бъде изкопана за:

$$16 * 16 * 64 * 0,1 = 1638,4\text{s}$$

Общо 27 минути, в които на теория са изкопани всички руди, което се свежда до около **32\$** на минута. Това е в същия порядък като получената експериментално, което я потвърждава.

Вече мога да продължа с останалите професии. Работата е идентична за всяка професия – подбирам тематични за нея игрови действия, които ще носят печалба. Правя оценка на времето за тяхното изпълнение, и конфигурирам печалбата да бъде по около **1\$** на секунда.

На този етап има навлизачи пари в циркулация и играчите могат да ги използват за търговия помежду си благодарение на командите на Essentials. При много ММО игри на този етап се проектират NPC магазини, които са друг източник на пари потребителите. За

да добавя нужен на играчите игрови аспект, но без да вкарвам нов „безкраен извор на пари“ (заштото въпреки балансирането на професиите, те са безкраен източник на пари), ще използвам приставката **DynamicShop**⁹⁵ версия 2. Тя дава възможност за създаване на GUI магазини, в които играчите могат да купуват и продават предмети.

Трудното отново са цените на предметите. Тук е и силата на DynamicShop. Магазините не са просто безкраен източник на предмети, а имат определена наличност, която нараства, когато играчи продават предмети, и намалява, когато купуват. Благодарение на тази наличност, цените не са фиксирани, а се изчисляват с формула в зависимост от това дали има недостиг или излишък на предмети.

$$\text{Крайна цена} = (\text{Отклонение} * \text{Базова цена}),$$

където отклонението е отношението между зададена средна стойност и текуща наличност. Благодарение на динамичната цена, не е нужно да се прави перфектна оценка на всеки предмет, а само да се направи грубо предположение основано на рядкостта на предмета например.



Фигура 25 – Пазар за предмети в сървъра. При взаимодействие с продавачите се отваря меню, от което могат да се купуват или продават предмети.

Използвайки тази формула, ще съзdam няколко магазина по категории на предметите в тях. Допълнително ще дефинирам общ баланс на пазара, което премахва опасността и от безкраен източник на пари. Когато играчи купуват предмети, наличните средства на магазините ще растат, а когато играчи продават предмети, средствата ще намаляват. Така на практика потребителите имат нов начин за печелене на пари, но тези пари не влизат в циркулация от нищото, а вече са били притежавани от други и просто си обменят собствениците. Допълнително поставена такса пък, взима част от парите, които биха влезли в баланса на търговците при покупка на предмет, и ги изкарва от игра. Следователно целият пазар не само не влошава инфлацията, а активно спомага за нейното овладяване.

⁹⁵ <https://www.spigotmc.org/resources/dshop.65603/>

Други такси ще поставя за създаване на защитена земя (ще спомена за това по-късно), за извършване на допълнителни (но не основни за играта) действия, за създаване на личен магазин от играчите, който освен това също има такси при всяка автоматична транзакция и т.н. За по-скъпи предмети ще добавя и автоматичен аукцион с приставката **AuctionHouse**⁹⁶.

Допълнително ще избегна обезценяване на парите, като проектирам още начини, с които да ги изкарвам от игра. Добър начин са платени услуги, които се предлагат на играчите. Те могат да са свързани с всякакви елементи на играта, като например този, че инструменти, оръжия и брони се чупят с времето. Благодарение на това ще добавя NPC ковач близо до пазара, който срещу заплащане ще ги поправя.

С това не се изчерпват методите за изкарване на пари от циркулация, но ще спомена за тях в следващите точки, защото те са обвързани с други решения на проблеми, които не са свързани с икономиката.

6.5 Имплементиране на решения на проблеми с взаимодействието между нови и стари играчи

В предложените решения на първо място стоеше наличието на някаква система, която да оценява нивото на развитие на даден играч. В оригиналната игра на практика има система за нива. Различни действия като добиване на руди и битка с чудовища носят точки опит необходими за всяко ниво. Тази система обаче не е това, за което говорих в главата с анализ, защото не може да се използва за оценка на цялостното развитите на играчите. Спечелените нива са по-скоро специална валута, с която играчите могат да омагьосват предмети, а освен това се губят абсолютно всички нива, когато играч загуби точките си живот.

Следователно трябва да проектирам отделна система за нива, с които да следя развитието на всеки. За целта ще използвам LuckPerms групи. Идеята е всяко ниво да представлява група, към която играчите принадлежат, и която им носи определени правомощия като всички нива следват йерархия подобна на тази с групите на екипа и всяко следващо наследява наградите от предходните. С помощта на правомощията, потребителите ще имат достъп до повече функционалности и команди, с които се улеснява тяхната игра.

Проектирах 29 нива, които не се печелят автоматично, а играчите сами трябва да отключват. Това се отличава от стандартните за ММО системи, но има огромни ползи. На първо място, играчите я виждат не просто като някакъв статистически елемент, а допълнителна и напълно нова за играта цел. Тази цел е в допълнение към нормалните цели в Minecraft, а не ги замества. Всяко ниво има определена цена, която играчите трябва да заплатят за да го отключат. Цената е разделена на виртуални пари и предмети. По този начин допълнително се изкарва огромно количество пари от циркулация, а освен това се премахват и предмети.

За да не се наруши баланса между нови и стари играчи, наградите се натрупват постепенно и плавно. Например един от бонусите, които се отключват с ниво 4 е бонус +1% печалба от професиите. Цената за нивото е 3500\$ като изключим предметите. Същият бонус +1%

⁹⁶ <https://www.spigotmc.org/resources/auctionhouse.61836/>

се отключва и с ниво 28, чиято цена е 42,000\$. Това ниво обаче наследява всички останали бонуси, затова играчи на този етап печелят доста повече от играч ниво 4. Играчите отключват нива със специално написано GUI меню с помощта на приставката **GUIPlus**⁹⁷.

Друг основен проблем представлява сложността на игровата среда. В случая това се свежда до битките с чудовища и други опасности. Избрах да следвам подход, при който по начало ще направя света максимално труден и предназначен за развити играчи, но със специални условия за нови играчи. За целта ще създам приставка, която да модифицира щетите, които играчите получават, в зависимост от ниво или времето им на игра.

За целта ще създам ново Java приложение. Кръстих го **MobIgnore**. Ще използвам средата за разработка IntelliJ, с която съм запознат най-добре. За да използвам предоставените от Paper методи, първо трябва да го добавя като зависимост на проекта. Това става най-удобно с помощта на инструменти за управление на проекти като Maven или Gradle. С помощта на IntelliJ. Ще използвам Maven като в основния конфигурационен файл pom.xml добавям:

```
<repository>
    <id>papermc-repo</id>
    <url>https://papermc.io/repo/repository/maven-public/</url>
</repository>
.

.

<dependency>
    <groupId>io.papermc.paper</groupId>
    <artifactId>paper-api</artifactId>
    <version>1.16.1-R0.1-SNAPSHOT</version>
    <scope>provided</scope>
</dependency>
```

Така необходимите зависимости ще се изтеглят и заредят автоматично. Приставката, която създавам, не притежава main() метод. Нейната основна логика се дефинира в клас, който разширява основен клас в BukkitAPI:

```
public final class MobsIgnore extends JavaPlugin {}
```

Този клас трябва да пренапише методите onEnable() и onDisable(), които са на практика входна и изходна точка на приложението.

Самата задача на приставката ще бъде да модифицира щетите, които нови играчи получават (да ги намали). За целта ще създам нов клас, който имплементира интерфейс на Bukkit – Listener. Всеки път, в който едно същество нанесе щети на друго в играта, този клас ще изпълни своята логика. Ще провери с помощта на правомощия дали играчът има защита и ще намали щетата с дефинирана в конфигурационен файл променлива, ако това е така:

```
@EventHandler
public void onDamageEvent(EntityDamageByEntityEvent event) {
```

⁹⁷ <https://www.spigotmc.org/resources/guiplus-simply-create-clickable-guis-ingame-gui-builder-1-7-1-18.38664/>

```

Entity damager = event.getDamager();
Entity damaged = event.getEntity();
//If it's a projectile from a player, skip damage adjustment.
if (checkIfProjectileFromHumanEntity(damager)) return;
//If damage is from player, skip damage adjustment (main functionality).
if (!damager.getType().equals(EntityType.PLAYER)) {
    if (damaged.hasPermission(SHIELD_PERMISSION)) {
        event.setDamage(event.getDamage()/damageModifier);
    }
}
}

```

Сега остава да се регистрира този клас в метода `onEnable()` и приставката да се компилира. С помощта на Maven, готовата приставка се пакетира в jar файл, който ще поставя в папката `plugins`. За да бъде зареден успешно в игра, е необходимо jar файла да съдържа файл с необходима мета информация – `plugin.yml`.

С помощта на тази приставка, играта ще бъде много по-приятна за нови играчи без това да пречи на развитите. Остава само да се добавят правомощията. Всеки нов играч ще получи достъп до правомощието, което добавя защитата, за две седмици с помощта на функция на LuckPerms за временни правомощия. Защитата ще получат и всички играчи между ниво 0 и 5.

Това на този етап е достатъчно. Няма да прилагам забрани срещу битки между нови и стари играчи, защото благодарение на нивата всеки играч може да оцени противника си, а новите играчи нямат достатъчно добра плячка следователно не представляват интерес. Това обаче не важи, ако развит играч е недоброжелател, но повече затова в следващата точка.

6.6 Имплементиране на решения на проблеми с недоброжелатели

Недоброжелателите са последния социален проблем, на който ще се спра. Той е и един от основните проблеми в по-късни етапи на една ММО игра, когато вече са коригирани повечето експлойти. Вече започнах да работя по него още с инсталациейта на LuckPerms.

Първото място, на което недоброжелателите извършват най-много щети, е основния комуникационен канал. В оригиналната игра не са предоставени достатъчно добри инструменти за справяне със ситуации в чатовете, защото предлагат единствено пълна забрана за влизане в сървъра. Това често е прекалено тежко действие срещу играч, който например използва нецензорни думи, не само защото разговорите не влияят директно на игровата среда, а и защото се случва играчи да се скарат и да си изпуснат нервите. Ето затова е нужна по-фина система за контрол, с която да могат да се поставят по-подходящи наказания. Добро решение предлага приставката **LiteBans**⁹⁸ версия 2. С помощта на правомощия, които ще се раздадат основно между админи и модератори, тя предоставя набор от по-меки наказания, не само за нарушения в чата, а като цяло в играта в сървъра.

⁹⁸ <https://www.spigotmc.org/resources/litebans.3715/>

На първо място е отнемане на правото за разговор в играта. Това включва общите чат канали, изпращане на виртуална поща и лични съобщения. С това наказание ще може такива нарушители да продължат своята игра, но им се отнема възможността за продължават с вредни за комуникацията действия.

Това наказание има и временен вариант. С негова помощ се поставя гореописаното наказание с пълната си сила, но за определен период от време. Много е подходящо в ситуации, в които по принцип сериозни играчи изпускат нервите си. Времевият диапазон може да бъде от секунди, до месеци като с помощта на правомощия и конфигурации, може да има различни гранични стойности в зависимост от това какъв член на екипа поставя наказанието.

За по-дребни нарушения в чата като заяждане, лек спам и т.н. могат да се поставят официални предупреждения. Тази функция на LiteBans позволява да се дефинират различни по-тежки наказания, в зависимост от броя на поставени такива предупреждения. Това се конфигурира от config.yml в папката plugins/LiteBans:

actions:

- '3:/kick -s:\$silent \$player Достигна 3 предупреждения: \$reason'
- '4:/tempmute -s:\$silent \$player 1 day Достигна 4 предупреждения: \$list'
- '5:/tempmute -s:\$silent \$player 3 day Достигна 5 предупреждения: \$list'
- '6:/tempmute -s:\$silent \$player 7 day Достигна 6 предупреждения: \$list'
- '7+:/tempmute -s:\$silent \$player 30 day Достигна максимално предупреждения: \$list'

Така конфигурирано, играчите няма да получат никакво наказание за първите две предупреждения, а при третото ще бъдат изхвърлени от играта с предупреждение. Ако продължат да нарушават, ще им се налага автоматично все по-дълго заглушаване, което достига до месец при 7 предупреждения. Тази система е много удобна за екипа, защото без да се налага да проверяват историята на играчите, ще се налагат подходящи наказания, а освен това се щадят играчи, които по принцип не са нарушители, но са прекрачили някое правило.

А именно правилата са това, по което се определя дали дадено действие или съобщение на играч заслужава наказание. В частта с експлойти споделих за правило, което забранява методите за заобикаляни на АФК системата. Сега ще допълня правилата като поради самата природа на човешката комуникация е невъзможно да се изброят всички забранени думи или словосъчетания, затова правилата са с по-обща, но в същото време доста специфична насока. Пълен списък на използваните правила ще кача в Google Drive, където всеки играч ще може да се запознае с тях:

Официални правила на сървъра: bit.ly/3KsOpnT ⁹⁹

Правилата имат различна цел. Някои са предназначени за контрол над производството на предмети в играта, като правилото за височина на фермите. Други пък забраняват действия, срещу които трудно могат да се проектират автоматични защити. Най-общо правилата дават възможност да се взимат мерки против недоброжелатели и да се избягват спорове между тях и екипа.

⁹⁹ Пълен линк, в случай че съкратения престане да работи: https://docs.google.com/document/d/1_efEX7ciAg6xp-mXGhn5Ln7tbPUm24pSZXHqVgC5ao/edit?usp=sharing

Друг проблем, на който ще обърна внимание, са **DDOS** атаки по сървъра. Както споменах в анализа, игрите са особено уязвими от този тип атака, защото е необходима непрекъсната връзка. В следващата точка ще обърна внимание на решение, с което ще предпазя сървъра от атаките с голям трафик. Игри обаче са уязвими и на много по-слаби атаки, които не се засичат от стандартните защити, ако тези атаки имитират реален играч. Няколко стотин заявки в секунда са достатъчни за причиняване на забавяне в играта, защото се извършват огромен брой операции при всяко свързване на играч със сървъра.

За да предпазя сървъра от този тип атака, ще създам допълнителен **прокси** сървър. С негова помощ, ще мога да затворя достъпа до играта, а играчите ще се свързват към проксита, което ще ги препраща към истинския сървър. На самото прокси вече мога да използвам приставки, които ще имат за цел да валидират дали заявката идва от реален играч преди сървъра дори да е разбрал за нея, следователно се пести огромно количество ресурс.

Това е възможно благодарение на софтуери, създадени за проектиране на мрежа от игрови сървъри на играта Minecraft, в които играчи могат да обикалят от една сървърна инстанция на друга. Популярен пример е **BungeeCord**¹⁰⁰, който се поддържа и разпространява от хората зад Spigot. Подобно на основния сървърен софтуер обаче, аз ще използвам негово разширение – **WaterFall**¹⁰¹, което е оптимизирано за Paper. Необходими са конкретни промени в конфигурациите за да знае сървърния софтуер, че заявките идват от прокси сървър, но освен това нищо на практика не се усеща различно.

На този етап сървърът е готов за игра. Създадох напълно уникална интерпретация на играта, която обръща внимание на много от проблемите, които произлизат от природата на ММО. Остава да обърна внимание на един последен проблем, на който не е обърнато внимание в оригиналната игра.

6.7 Техническа оптимизация на сървъра

Въпреки че технологията на Minecraft позволява взаимодействие на огромен брой играчи, необходимите изчислителни ресурси за да могат повече от 20 души да играят едновременно започват да достигат до границите на възможното днес. Това е така, защото много от решенията са правени с идеята за самостоятелна игра. Paper по подразбиране обръща сериозно внимание на този проблем и значително подобрява производителността. Задачата обаче не може да се изпълни докрай, защото има различни типове Minecraft сървъри. Някои, като този, който проектирах за текущия експеримент, надграждат оригиналната игра. Други пък само използват играта като игрови двигател и създават напълно различни игрови среди с коренно различни цели. По тази причина няма напълно универсална оптимизация, защото някои решения са добри в една ситуация, а вредят в друга.

Разработчиците на Paper обаче са помислили за това, затова са предоставени конфигурационни файлове, с които могат да се променят игрови елементи, но по-важното да се правят допълнителни оптимизации. Сега ще направя именно това като предварително ще отбележа, че всички използвани конфигурационни файлове се намират в същата директория, в която е jar-а на основния сървърен софтуер.

¹⁰⁰ <https://ci.md-5.net/job/BungeeCord/>

¹⁰¹ <https://papermc.io/downloads#Waterfall>

Ще започна от това, че колкото по-голяма част от света е активна, толкова повече ресурс ще бъде необходим. Затова първо ще намаля размера на игрова среда, която се зарежда в паметта, около играчите. Този размер се определя като радиус за единица от картата, като ще припомня, че света е разделен на отрязъци с размер 16×16 . По подразбиране този размер е 10 такива части, а аз ще го намаля на 5. Това външност е настройка, която идва от оригиналната игра и се прави в конфигурационния файл **server.properties**. Промяната на тази стойност води до по-ниска видимост в играта. Това обаче няма да се отрази сериозно на играчите, защото те рядко виждат надалеч, поради скриване на хоризонта от гори, хълмове и планини. На практика от 400 заредени части 16×16 , вече ще са активни само 100.

Все пак ще жертвам малко производителност за допълнителна видимост, благодарение на функционалност на Paper, която позволява да се зареждат визуално повече части, но без в тях да се извършват логически операции. Това става в **paper.yml**:

```
viewdistances:  
    no-tick-view-distance: 7
```

Следващата оптимизация е свързана с друга интензивна за процесора задача – изкуствения интелект. В оригиналната игра е заложено да се появяват определен брой чудовища и животни около всеки играч, когато са спазени определени условия. Този брой обаче е прекалено голям и не се скалира добре, затова ще го намаля. Това става в конфигурационния файл **bukkit.yml** като променя *spawn-limits* на:

```
spawn-limits:  
    monsters: 20  
    animals: 8  
    water-animals: 3  
    water-ambient: 1  
    ambient: 1
```

Тази промяна със сигурност ще помогне, но идва с цена. Играчите ще усетят липсата на същества, но най-вече на чудовища. За да коригирам това, ще използвам друга настройка, с която се дефинира максималния радиус, в който се появяват чудовища. В оригиналната игра има много чудовища, но те са разпръснати в сравнително голям откъс от картата. Следователно е достатъчно да намалявам радиуса, докато плътността на чудовища за единица от картата е същата като в оригиналната игра. Оптималната стойност е с единица по-малко от общото разстояние, което дефинирах в **server.properties**. Затова конфигурирам в **spigot.yml**:

mob-spawn-range: 4

След това ще конфигурирам друг радиус, като всички същества извън него ще бъдат обработвани по-рядко, което допълнително спестява ресурс без да се отразява на играта, защото се случва на разстояние от играчите. Отново в **spigot.yml**

```
entity-activation-range:  
    animals: 8  
    monsters: 8  
    raiders: 48  
    misc: 8  
    water: 8
```

С настройка в `paper.yml` ще оптимизирам друг елемент, който играчите използват за създаване на машини. В оригиналната игра, с този елемент могат да се изработват специални блокчета, които извършват логически операции. Благодарение на тях са строени напълно работещи 8 битови компютри изцяло в игровата среда. При огромен брой хора и много машини е задължително този елемент да бъде оптимизиран, затова ще настрои сървъра да използва по-оптимизиран алгоритъм за обработка на събитията свързани с този предмет:

```
use-faster-eigencraft-redstone: true
```

Ще извърша още много оптимизации, но те вече не са свързани пряко с огромния брой хора и природата на ММО, затова няма да ги описвам подробно и ще спра до тук. Остава единствено да се осигури хостинг и екип.

6.8 Подготовка за начало на експеримента с реални играчи

Сървърът е напълно готов да посрещне играчи. Освен описаните в тази глава стъпки, са изпълнени още много задачи, които не са пряко свързани с проблемите на ММО или ги засягат в много малка степен. Все пак ще спомена на кратко за тях.

Инсталирах няколко приставки, които визуално подобряват игровото преживяване в сървъра. Това включва интерактивни менюта, hologrami, форматиране на общите комуникационни канали и други. Добавих още функционалности, които са в полза на екипа за следене на състоянието на средата като статистики за предметите на играчите, невидимост за незабелязано наблюдение на съмнителни играчи, комуникационен канал за екипа и други.

Обърнах внимание и на пропуск в оригиналната игра, който не е конкретно проблем при ММО, а при всяка мрежова игра. Става дума за липсата на защити на постройки в сървъра. Без намеса от моя страна, всеки би могъл да унищожава всичко създадено в играта, което би я обезсмислило. За да предотвратя това, въведох система за купуване на земя от играчите, която е защитена от унищожение. Това допълнително спомогна и за изкарване на пари от циркулация.

Някои от приставките беше нужно да актуализирам и компилирам допълнително, защото не работеха под избраната версия на основния сървърен софтуер. Крайния резултат са над **70** приставки, всяка от които специално подбрана и конфигурирана без да възникват конфликти. Освен това отделих значително време и за проектиране и строене на сгради и важни места в играта като начална точка, пазар и арени.

За хостинг избрах наст **VPS** сървър с достатъчно добър процесор, който да работи с до около 35 души едновременно без проблем и SSD памет, което е задължително поради огромния брой I/O операции. Единственото нещо, което трябваше да подгответя, беше да инсталрам JVM (версия за Linux, инсталран на виртуалната машина), инструмент за по-лесен достъп до конзолата след стартиране на сървъра - GNU Screen¹⁰² и да добавя мрежови правила против DOS атаки. Избрах този вид хостинг и поради факта, че предлага безплатна защита против DDOS атаки, без която експеримента би бил изключително уязвим.

¹⁰² <https://linuxize.com/post/how-to-use-linux-screen/>

Екипът от доброволци, които да поддържат игровата среда, също вече е подбран. Това е така, защото експериментът се извършва под името на най-стария публичен сървър на играта Minecraft в България. Играчите, които ще са част от екипа, вече са доказали, че могат да се справят със задачата, в предходни сезони¹⁰³. Освен екип, това означава, че вече има и играчи, които ще започнат още от първия ден. Това е и в основата на избора за хостинг, защото предположението за около 35 души беше направено на база популярност в предходни сезони. Тези играчи не са подбирани специално за експеримента, което означава, че резултатите от него би трябвало да важат за всяка органична група хора, които се интересуват от този жанр.

Вече всичко е готово за провеждане на експеримента. Основна работа в това време ще бъде да се следят ефектите от приложените решения както и цялостното състояние на сървъра. Да се отбелязват по-важни събития и да се оправят възникнали проблеми.

¹⁰³ Всяка нова имплементация на сървъра се нарича „сезон“. Всеки сезон е напълно независим от предходните, особено този, на който ще се провежда експеримента.

7 Практически експеримент с игрови сървър на играта Minecraft.

7.1 Кратък обзор на експеримента. Валидиране на получените резултати

Експериментът се проведе от октомври 2020 година до юни 2021 година. Общите уникални посещения са над 6000, а играчите са прекарали общо над **55560** часа в играта. Наличните виртуални пари в циркулация на края са **2,930,709.79\$**. Размерът само на игровата среда е близо 60GB.

Броят на играчите определено е достатъчен за да бъде играта масова. В него обаче са включени играчи, които само са посетили сървъра и по една или друга причина не са решили да играят. Следователно реалния брой ще бъде по-малък. Около 2300 души са прекарали повече от половин час игра в сървъра, което е достатъчно да се повлияе на средата, следователно резултатите от експеримента могат да се приемат за валидни.

Затова мога да пристъпя към анализ на конкретните приложени решения с цел тяхното валидиране. Преди това кратък обзор на техническо състояние на сървъра по време на експеримента. Като цяло играта в сървъра беше технически издържана. Повечето време не се забелязваха никакви проблеми, но поради по-висок интерес към сървъра от очакваното в определени часови периоди (пикови часове) влизаха едновременно повече от 40-50 души. Това предизвикваше забавяне на игровия часовник, което се отразяваше негативно върху играта. Въпреки това нямаше сривове, което е доказателство за стабилността на сървъра. Малко след старта на играта, създадох допълнителен свят, в който се провеждаха игри и състезания в пиковите часове. По този начин, голяма част от играчите се намираха на едно място, отделено от основния свят, което облекчаваше много работата, която трябваше да извършва процесора.

По време на експеримента постоянно се извършваха актуализации, основно на сървърния софтуер, но и на доста от приставките. Тези актуализации предпазиха сървъра от експлойти, с които бяха саботирани други активни по времето български сървъри. Играчи съобщаваха, че в сървъра не работят слабости, които иначе използват другаде. Спокойната игра в сървъра позволяше по-голям фокус върху останалите анализирани проблеми.

7.2 Профил на участниците в експеримента

В експеримента участваха играчи, които не са специално подбирани. Това означава, че резултатите ще са автентични, но за сметка на това няма никаква демографска информация, която е нужда за пълноценен анализ.

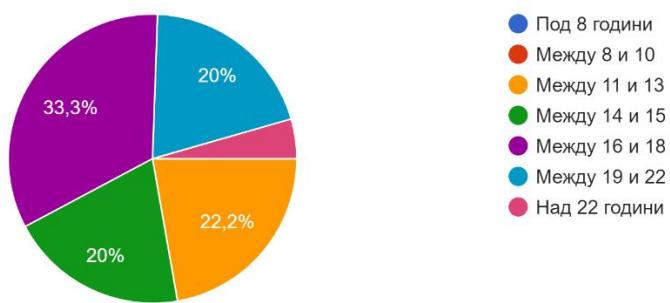
Затова по време на експеримента проведох анкета, която има за цел да отговори на някои от тези въпроси. Тя е напълно анонимна, което е с цел участниците да бъдат напълно откровени. Анкетата се проведе с помошта на Google Forms в края на третия месец като участниците бяха 45 от най-активните играчи. Част от въпросите са свързани с информация за потребителите, а друга за наблюденията им относно игрови елементи свързани с

приложените решения на проблеми с ММО. На втората част ще се спра малко по-късно като анализирам взаимодействието между играчи. Абсолютно всички са отбелязали, че са от България, което е очаквано, защото много от игровите елементи са преведени на български и би било трудно за играч, който не знае езика, да играе пълноценно.

Според общественото мнение, видео игрите са основно за малки деца, затова предоставените опции на въпрос за възрастта се фокусират върху тази група. Резултатите показват, че най-много играчи са в групата между 16 и 18 години, 20% от играчите са в студентска възраст, а 4,4% са на възраст по-голяма от 22. Нито един от анкетираните не е под 10 годишен. Това за мен е очакван резултат, защото играта в сървъра е сравнително сложна, особено ако се играе с цел висока позиция в класациите. 84,4% от анкетираните са мъже, съответно само 15,6% жени.

На колко години сте?

45 отговора



Фигура 26 – Разпределение на играчите по възраст. Възможните отговори „Под 8“ и „Между 8 и 10“ не получават нито един глас, а „Над 22 години“ е отбелязан от 4,4%.

Почти всички анкетирани имат повече от три години опит с играта като само седем играчи са отбелязали по-малко. Това отново е очаквано, защото проектирания сървър надгражда сериозно оригиналната игра и е сравнително трудно да се играе за пръв път.

На въпрос какви други ММО игри играят, най-споменавана игра е World Of Warcraft (13% от играчите), което не е изненада. Неочаквано обаче беше, че 18 души са отбелязали, че не играят други ММО игри, освен Minecraft. Това може да се обясни от факта, че играчите масово не знаят какво точно представлява ММО жанра. Тази теза се потвърждава от огромния брой споменати игри, които в никакъв случай не са ММО (дори не се спори за тяхната класификация в интернет). Анкетиран играч дори признава, че не знае какво означава ММО. Резултатите от този въпрос допълнително потвърждават твърдението ми, че жанрът е много слабо изследван, въпреки че е от най-играните.

Анкетата внесе яснота за играчите, затова вече мога да пристъпя към анализиране на резултатите.

7.3 Анализ на икономическо състояние на сървъра

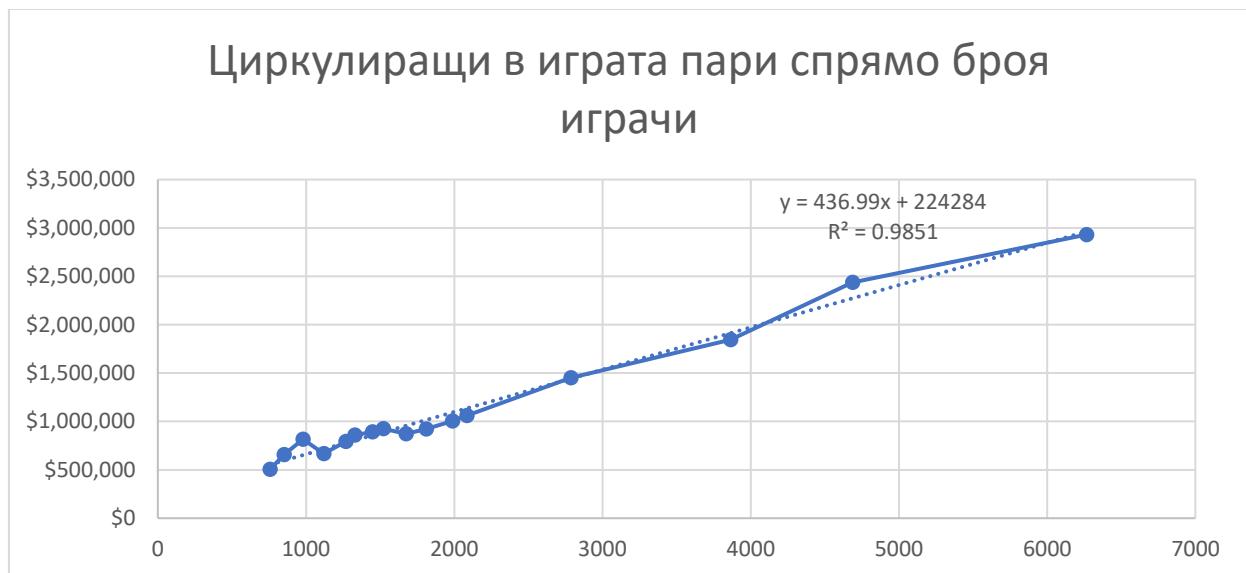
Играчите започнаха веднага да вкарват пари в циркулация, благодарение на професиите. В началото нямаше особено големи харчове, защото нямаше развити играчи. Липсата на предмети, оръжия, екипировка, бази се отразяваше с липса на търговия в първите дни. Това

обаче започна да се променя бързо с всяка изминалата седмица. След потребителите започнаха да използват парите по-активно, вече започнах да следя дали се наблюдава инфлация.

Поради факта, че постоянно идват нови играчи, се очаква парите в циркулация да растат. Това обаче не означава, че тяхната покупателна стойност намалява, защото, въпреки че има повече пари, има и все повече играчи, които да ги използват. Следователно за да проверя дали виртуалната валута е стабилна, трябва да я съпоставя с броя хора. Ако успея да намеря подходяща линейна математическа функция, която да описва разпределението, то мога да заключа, че виртуалната валута е стабилна.



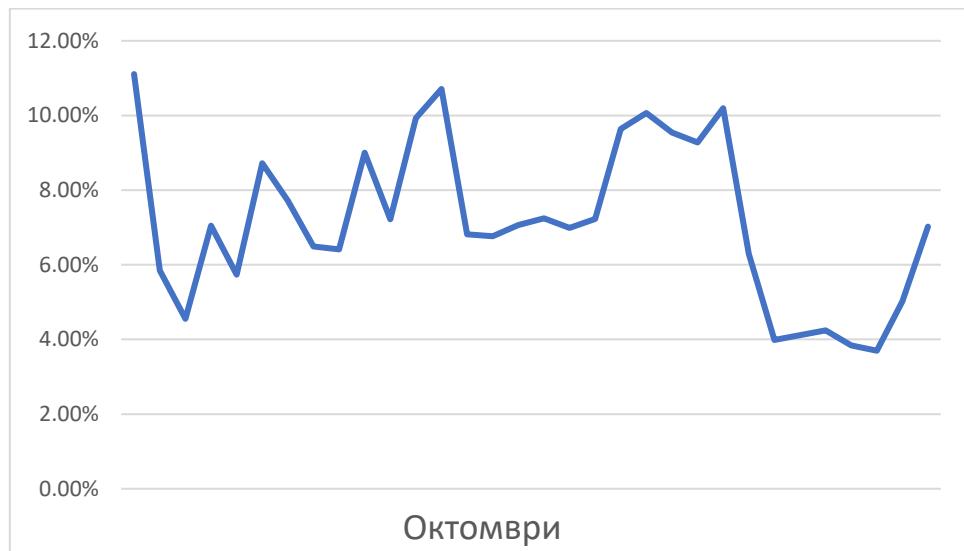
Фигура 27 – Графика на растежа на пари спрямо броя хора в сървъра за месец октомври. Кофициентът на детерминация R^2 (квадратът на корелационния коефициент) е 0,95, което се интерпретира като много висока корелация между разпределението на данните и предложението с уравнението на графиката линеен теоретичен модел.



Фигура 28 – Графика на растежа на пари спрямо броя хора в сървъра за периода ноември 2020 – юни 2021. Стойността на R^2 е 0,98, което се интерпретира като много висока корелация между разпределението на данните и предложението с уравнението на графиката линеен теоретичен модел.

Получените резултати не показват никакви признания на инфлация, което е абсолютен успех. В някои периоди дори има абсолютно равенство между навлизации в циркулация пари и излизации. Някои от по-сериозните отклонения се обясняват с играчи, които събират пари за нещо скъпо, най-често за да отключат няколко нива наведнъж. Като цяло разпределението на парите между играчите в сървъра също остава сравнително постоянно.

Казвам постоянно, но в никакъв случай равностойно, защото със сигурност се забелязва натрупване на богатство при определени играчи:



Фигура 29 – Какъв процент от всичките циркулиращи пари се притежават от най-богатия играч в сървъра за месец октомври.

Друг интересен резултат се наблюдава и при пазара с динамични цени. Търговците бяха почти винаги с нулев баланс, което превърна пазара в бартерен такъв. Играчи купуват необходими неща и веднага си връщат парите като продават ненужни такива. Цените на предметите, особено тези, които се добиват най-лесно от играчите, масово се сринаха. Това се очакваше за най-често срещаните блокчета като дървесина, камък, пръст, но реално почти нямаше предмет, който да поскъпва. Това показва, че играчите много повече продават предмети, отколкото купуват и обяснява вечната липса на налични средства у търговците. Единствените предмети, които отбелязват ръст, като при някои е огромен, са консумативи. Особено храна и продукти за отвари.

White Wool	Spruce Log
Buy: 0,63 ↓ 84.55%	Buy: 0,15 ↓ 94.27%
Sell: 0,10 ↓ 84.55%	Sell: 0,02 ↓ 94.27%
Cooked Porkchop	Steak
Buy: 2,68 ↑ 98.88%	Buy: 3,92 ↑ 154.79%
Sell: 0,40 ↑ 98.88%	Sell: 0,59 ↑ 154.79%

Фигура 30 – Някои от най-осезаемите промени в цените. В червено е обикновена вълна и дървесина, а в зелено са пържоли (храна).

На този етап имам интересни резултати. Благодарение на решението да използвам цена на пръст от друг сървър при определяне на печалбата за час от професиите, имам възможност и да сравня икономическото състояние с това на друг игрови сървър.

При около 2500 уникални играчи и горе-долу сравними цени на предмети, другият сървър имаше в циркулация над 60 милиона виртуални единици. Това са близо 24,000\$ на играч, при около 500\$ в сървъра от експеримента.

Една от основните разлики с другия сървър беше, че NPC търговците притежаваха безкраен баланс и фиксирани цени. Това сравнение води до извода, че основен виновник за инфлация в MMO игрите са места, на които играчи могат да продават предмети без ограничения.

Успех има и с таксите като повечето пъти не правят впечатление. Изключение обаче се оказаха таксите в аукциона. Най-вече по време на предварително уговорена търговия, в която играч продава предмет за определена сума, но получава по-малко и директно усеща загубата.

Като цяло икономическата система в сървъра е напълно успешна. Оправдаха се повечето очаквания, но имаше и изненади като големите разлики при динамичните цени. Резултатите определено потвърждават приложените решения като се затвърждава и тезата, че най-доброто решение е да се приложат много методи за изкарване на пари от циркулация и да се сведат до минимум крайните източници на пари. Най-желан краен резултат е система, която стимулира обмен на парите между самите играчи, а не между сървъра и играчите.

Повече взаимодействия обаче водят до по-голям шанс за конфликт. Именно тогава се намесващ екипът.

7.4 Анализ на работата на екипа и на поведението на недоброжелателите

В периода октомври – юни временно е отнето правото за комуникация на играчи над 700 пъти. Това не включва наказанията поставени автоматично за натрупване на предупреждения. Официално играчите са предупреждавани над 2000 пъти. Пълна забрана за игра е поставена над 800 пъти, повечето от които обаче временни. Най-много играчи са получили пълна забрана за игра поради използването на зловреден софтуер.

Общо по време на експеримента помагаха 11 модератори, 4 администратори и един оператор, освен мен. Това беше напълно достатъчно повечето време. Все пак се наблюдаваха периоди в денонощието без членове от екипа на линия, но това се компенсираше с налична външна система за връзка, с която играчите можеха да извикат модератори при нужда.

Въпреки че екипът доказа своята полза, огромния брой поставени наказания води до заключението, че е било по-добре да се добави и някаква автоматична система за засичане на често срещани вулгарности и обиди. Въпреки че такава система има ограничено действие, тя би била много по-ефективна в комбинация с екипа.

Действия на недоброжелатели, които екипът няма как да предотврати, са технически атаки по сървърната инфраструктура. По време на експеримента многократно се доказва, колко важно е да се предприемат защитни мерки предварително. Сървърът беше атакуван по всеки възможен начин.

Първата по-сериозна ситуация се случи още в началото на експеримента. Поради моя конфигурационна грешка, първите няколко седмици реално не беше активна защитата

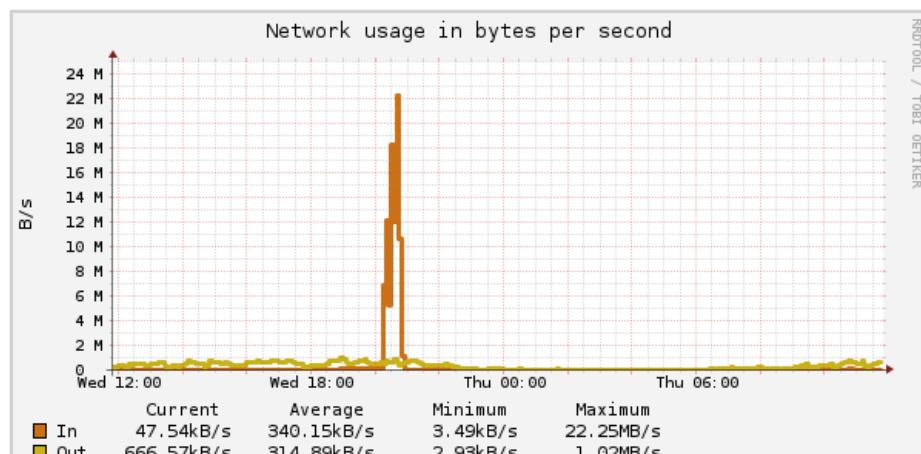
против DOS атаки. Благодарение на този пропуск и използвайки пакети, които симулират реални играчи, работата на сървъра беше временно прекъсната от недоброжелател, който запълни всички свободни места в сървъра с фалшиви играчи:



Фигура 31 – Атака по сървъра. Всички свободни 120 места в сървъра са заети, а освен това процесорът не може да обработи игровата логика на всички фалшиви играчи, което води до забавяне на игровия часовник до 13 цикла в секунда от нормално 20.

След като коригирах пропуска, този тип атака повече не се наблюдаваше. Това обаче не спря недоброжелателите, защото хостинга съобщи неведнъж за предотвратена масивна трафик атака. Недоброжелателите подбираха конкретни моменти за извършване на вредното действие, в които има доста играчи на линия, но без администратори и оператори. Това е най-вероятно с цел възстановяването в случай на успешна атака да бъде възможно най-бавно. За щастие опитите за срив бяха като цяло неуспешни, но успяхаха да причинят неудобство на играчите, защото обикновено минават няколко секунди, в които атаката все

Network usage



Фигура 32 – Атака на 22 април 2021г. Количество трафик е така подбрано (малко), че да не се засече веднага от хостинг защитите, което позволи на атаката да продължи близо половина час, през който всички играчи имаха трудности с играта.

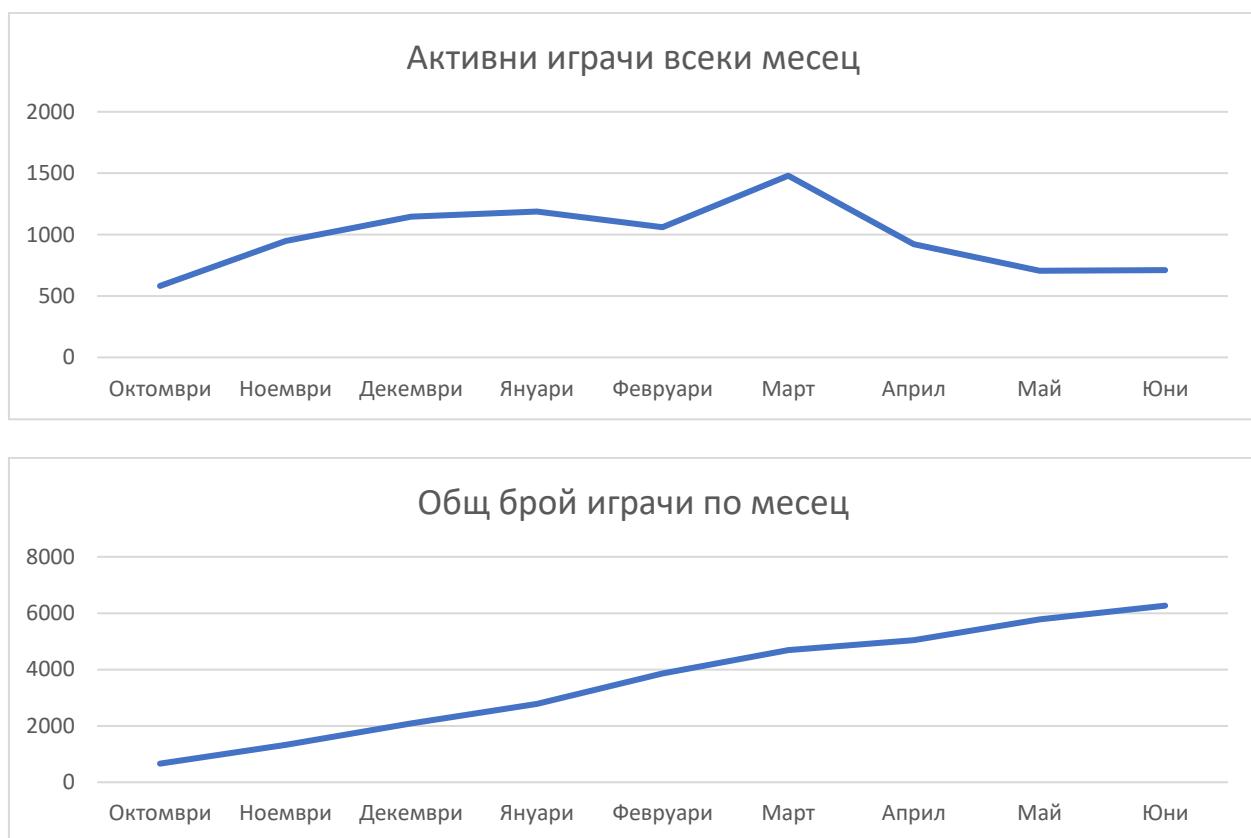
още не е отблъсната от хостинг защитите, което причиняващо проблеми с комуникацията на играещите потребители (невъзможност да извършване на действия, лаг, визуални неточности).

Изводът е, че независимо какъв е мащабът на една ММО игра, тя задължително ще бъде обект на атаки и е невъзможно да не се обърне сериозно внимание на този проблем и всичките усложнения, които идват от природата на видео игрите като цяло.

7.5 Анализ на играчите и решенията пряко свързани с тях.

Последните проблеми, които ще разгледам, са най-предизвикателни за анализ, защото трудно може да се направи математически модел на неща, които са сравнително субективни и например нещо сложно за един играч може да бъде нещо тривиално за друг.

Статистиките, които все пак могат да се обработят, начертават положителна картина. Интересът към сървъра от стари играчи е сравнително постоянен и не затихва и интереса на нови играчи:



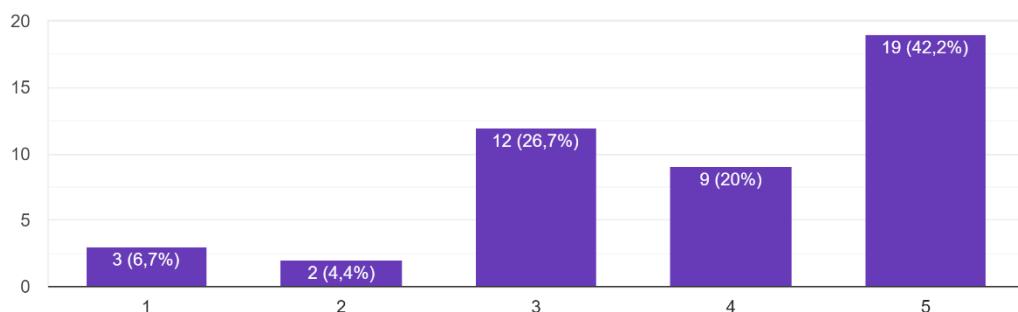
Фигура 33 – Графики показващи брой играчи, които са се свързали поне веднъж през дадения месец (горе) и общо население към края на дадения месец (долу). Графиките показват постоянен интерес като лекият спад през март може да се обясни с идването на пролетта, следователно по-малко време за игра.

За да направя подходяща оценка на ефективността на приложените решения, ще се допитам до самите играчи. Това е втората част от анкетата, за която говорих в точката за профил на участниците. Въпросите са свързани с изразяване на мнение по ликъртова скала от едно до пет, където 1 отговаря на най-негативно мнение, а 5 - на най-позитивно. Играчите споделиха наблюденията си по различни теми като избрах да използвам тази скала за да получа по-прецизна и по-лична информация.

На първо място ще разгледам техният отговор относно взетите мерки за защита на нови играчи. Резултатите показват среден до добър успех като на въпроса „Колко трудно ви беше да започнете играта този сезон?“ 26,7% са поставили оценка 3, 20% оценка 4, 42,2% са заявили, че началото им е било изключително лесно, а само 11,1%, че е било много трудно. Въпреки че част от анкетираните са започнали игра още от началото, съвсем не е имало развити играчи, липсата на ниски оценки показва, че мерките за защита на нови играчи работят поне в някаква степен. Решението игровата среда да се проектира за развити играчи, а да се приложат облекчения на нови, не доведе до негативно влияние на притока на нови и активността на стари играчи.

Колко трудно Ви беше да започнете играта този сезон?

45 отговора

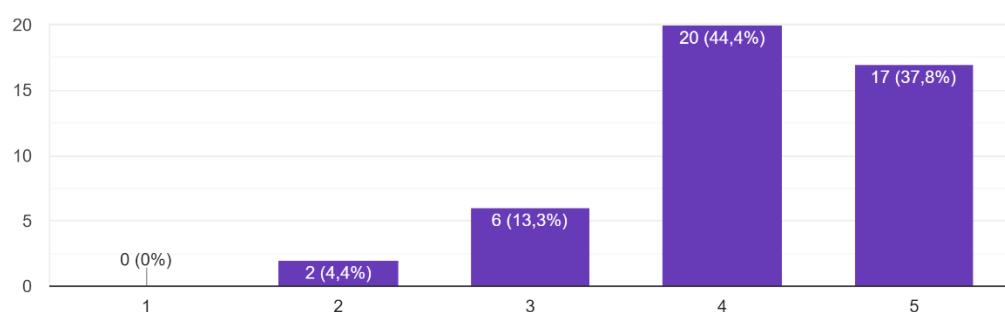


Фигура 34 – Диаграма на получените отговори на въпроса „Колко добре се справя Екипът с контрола над хакери“.

На въпросите „Колко добре се справя Екипа с контрола над хакери?“ и „Колко добре се справя Екипа с контрола над чата?“ никой не поставя оценка 1. Най-преобладаващата оценка обаче е 4 (44,4% от анкетираните), а не 5 (42,2 %), което не е изненадващо, поради факта, че няма система за превенция и играчите виждат нарушението дори екипът веднага да се заеме със случая. Това е допълнително доказателство, че въпреки ефективността на реален човек, е силно препоръчително да се проектира и автоматична система за цензура на неподходящи съобщения.

Колко добре се справя Екипът с контрола над чата?

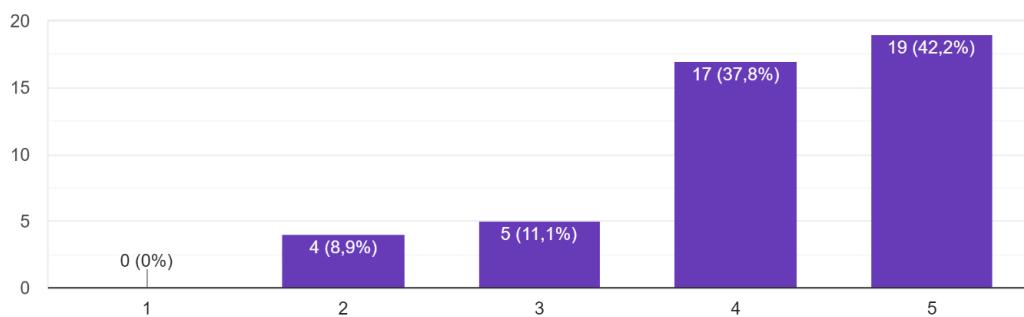
45 отговора



Фигура 35 – Диаграма на получените отговори на въпроса „Колко добре се справя Екипът с контрола над чата“.

Колко добре се справя Екипът с контрола над хакери?

45 отговора



Фигура 36 – Диаграма на получените отговори на въпроса „Колко добре се справя Екипът с контрола над хакери“.

Същите съображения са валидни и спрямо действията против недоброжелатели, които използват зловреден софтуер. Играчите виждат летящи играчи (действие, което по принцип не може да се случи без модифициран софтуер или специално правомощие) преди да бъдат наказани от екипа, което оставя лошо впечатление. Изводът е, че особено при помащабни MMO игри е задължително да се проектират автоматични системи за превенция на нарушения, която не само да следи чата, но и да анализира действията на играчи с цел да забележи автоматизации или невъзможни действия.

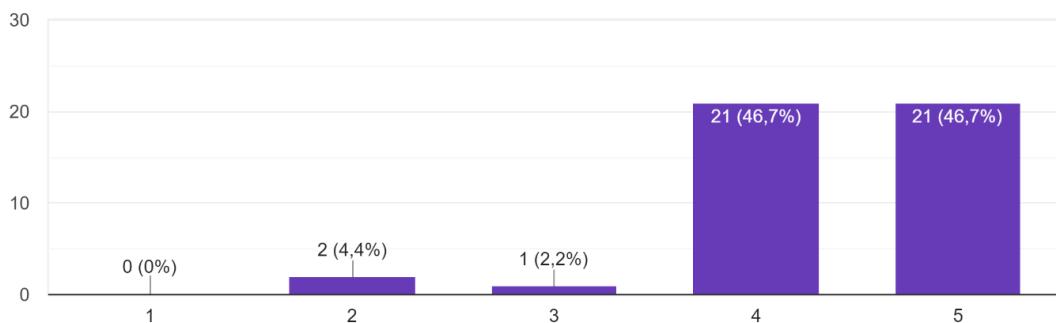
7.6 Обобщение на анализа

С помощта на статистически анализ и допитване до играчите бяха успешно валидирани конкретни решения на MMO проблеми. Най-добри резултати се видяха в икономическото състояние на сървъра, където всички цели бяха постигнати. Играта в сървъра беше технически издържана, но показва нужда от добър метод за предварителна оценка на очаквания брой играчи. Огромния брой наложени наказания доказва нуждата от реални хора, които да упражняват контрол над игровата среда и конкретно недоброжелатели, но мнението на играчите показва и сериозна нужда от автоматична система за превенция на вредни действия и комуникация. Технически атаки по сървъра доказваха и че е задължително да се проектират добри защити независимо от мащаба на играта.

В заключение на анализа ще покажа резултатите от доста общ и субективен въпрос. Ще отбележа, че анкетата беше напълно анонимна и без обещания за награди или друг тип възнаграждение спрямо отговорите. Участието в нея беше и доброволно. На въпрос „Колко сте доволни от играта в сървъра“, почти всички играчи отговарят положително. Отговорите са съгласно 5-степенната скала на Ликърт както следва: 1 - много недоволен, 2 - по-скоро недоволен, 3 - не мога да преценя, 4 - по-скоро доволен, 5 - напълно доволен. Този абстрактен въпрос не валидира конкретно решение, но дава цялостна оценка на игровото преживяване и благодарение на нея, експериментът може да се приеме за напълно успешен.

Колко сте доволни от сървъра?

45 отговара



Фигура 37 – Диаграма на получените отговори на въпроса „Колко сте доволни от сървъра“, което представя оценка на играчите за свършената по време на експеримента работа.

8 Заключение

8.1 Обобщение на постигнатите резултати

Целта на тази дипломна работа беше да се изследват проблемите при ММО видео игри, като се проектира, тества и внедри игрови сървър на играта *Minecraft*, на който да се проведат експерименти с реални играчи, имащи за цел анализ и валидиране на предложените решения.

Всички задачи бяха изпълнени. Беше направен анализ на ММО жанра, който включи извеждане на точна дефиниция, кратък исторически обзор и представяне на примери. След това беше направен сравнителен анализ на проблемите, които възникват по време на разработката на ММО видео игри, породени от спецификата на жанра. Те бяха разделени на две групи – технически (породени от големия брой хора и техническото им взаимодействие) и социални (породени от големия брой хора и социалното им взаимодействие). На края на всяка група проблеми бяха предложени и подходящи решения.

Анализът продължи с преглед на използваните по принцип технологии за разработка на ММО видео игри. Беше представена концепцията за игрови двигател и бяха дадени примери. На края беше представена играта *Minecraft* като бе обоснован изборът ѝ за провеждане на експеримент с реални играчи.

Дипломната работа продължи с проектиране на игровия сървър като бяха описани всички основни стъпки както и кои решения ще се приложат. Експериментът продължи 9 месеца като тезата завърши с анализ на получените резултати. С тяхна помощ теоретично поставените възможни решения на проблеми при проектирането на ММО видео игри бяха успешно валидирани.

8.2 Насоки за бъдеща работа

Въпреки че експериментът потвърди положителния ефект от решението, в някои случаи се виждаше, че са необходими допълнителни стъпки за пълното справяне с изследваните проблеми.

На първо място беше грешната оценка на очакван брой хора. Поради по-слаб от необходимото хостинг, имаше периоди, през които потенциално нови играчи бяха отблъснати, защото изпитват технически проблеми още при първото си влизане в играта. Това наблюдение е важно, защото води до извода, че лоша производителност ограничава органичния приток на нови играчи. Необходимо е да се намери по-добър начин за правене на такава оценка.

Друг важен момент беше работата на екипа. Въпреки че доброволците оправдаха очакванията, мненията на играчите и статистиките за поставени наказания водят до извода, че са необходими допълнително и автоматични системи, които не само да цензурират

неподходящи съобщения, но и да проверяват дали действията на играчите се извършват от тях, а не от компютър. Бъдеща работа ще бъде анализ как точно да работят такива системи, което вероятно ще бъде различно за различните MMO под-жанрове.

Необходим е и много по-сериозен теоретичен анализ на жанра. Фактът, че някои играчи дори не знаят какво представлява той, а други не могат да определят правилно дали една игра е представител на този жанр, показва сериозната липса на консенсус по въпроса. Подробен анализ на жанра например би обяснил защо много MMO игри имат едни и същи елементи, въпреки че те не са задължителни по дефиниция, което би трябвало да доведе до извеждане на допълнителни критерии, с които по-недвусмислено да се класифицират игри.

MMO е уникален жанр, защото предлага безкрайно много възможности за игра. В основата му стоят основни човешки взаимодействия, които не могат да се симулират по никакъв начин с наличните днес технологии за изкуствен интелект. Може би някой ден ще се появят видео игри, в които NPC персонажите са толкова умни, че няма да бъде възможно да се различават от реален играч. Въпреки това смяtam, че дори такива игри няма да могат напълно да осигурят същото игрово преживяване породено от човешката непредсказуемост и изобретателност. Доказателство за това е фактът, че обикновена игра с блокчета и няколко чудовища е една от най-играните за всички времена и е все още популярна десетилетие по-късно. Затова този жанр не трябва никога да се забравя, а да се надгражда и развива.

Използвана литература

- [1] SteamSpy, Games released in previous months overall (2020)
Достъпно на <https://steamspy.com/year/>
- [2] Dictionary.com, Definition for MMO (2009)
Достъпно на <https://www.dictionary.com/browse/mmo/>
- [3] Techopedia, What does Massively Multiplayer Online Game mean (2017)
Достъпно на <https://www.techopedia.com/definition/27054/massively-multiplayer-online-game-mmog>
- [4] Your Dictionary, MMOG Definition (2006)
Достъпно на <https://www.yourdictionary.com/mmog>
- [5] Lexico, Definition of MMOG in English (2019)
Достъпно на <https://www.lexico.com/en/definition/mmog>
- [6] Bell, Mark W.
Toward a Definition of "Virtual Worlds"
Journal of Virtual Worlds Research, Vol 1, No 1 (2008)
ISSN: 1941-8477
- [7] Schroeder, Ralph
Defining Virtual Worlds and Virtual Environments
Journal of Virtual Worlds Research, Vol 1, No 1 (2008)
ISSN: 1941-8477
- [8] Bartle, Richard A.
From MUDs to MMORPGs: The History of Virtual Worlds.
In Hunsinger, Jeremy; Klastrup, Lisbeth; Allen, Matthew (eds.): International Handbook of Internet Research (Springer, 2010)
ISBN 9781402097898
- [9] History of Information, Maze War, the First Networked 3D Multi-User First Person Shooter Game (2012)
Достъпно на <https://www.historyofinformation.com/detail.php?id=2023>
- [10] Broadband Now, ARPANET – The First Internet (2020)
Достъпно на https://broadbandnow.com/internet/i/ii_arpanet.htm
- [11] NG Alphas - Meridian 59
Next Generation Magazine, Page 111 (November, 1996)
- [12] Web Archive, Archive of GameDaily
Достъпно на
<https://web.archive.org/web/20010426130147/http://www.gamedaily.com/features/meridian/page2.html>

- [13] Talking: Is Richard Garriott really building a better world?
Next Generation Magazine, Page 09 (March, 1997)
- [14] Ultima Online, Spring Contest 2016 (2016)
Достъпно на <https://uo.com/spring-contest-2016/>
- [15] MMO Populations, World of Warcraft Yearly Population Chart (2020)
Достъпно на <https://mmo-population.com/r/wow>
- [16] LeagueFeed, How many people play league of legends? (2020)
Достъпно на <https://leaguefeed.net/did-you-know-total-league-of-legends-player-count-updated-2020/>
- [17] Imperia Online, Timeline (2020)
Достъпно на <https://imperiaonline.bg/company/timeline/>
- [18] Discovery Freelancer, Gallery (2020)
Достъпно на <https://discoverygc.com/>
- [19] George Coulouris, J. Dollimore, T. Kindberg, G. Blair
Distributed systems – concepts and design (Addison-Wesley, 2012)
ISBN: 9780132143011
- [20] Database Programming with JDBC & Java, Second Edition (O'Reilly Media, 2000)
ISBN: 9781565926165
- [21] Yahyavi, Amir & Kemme, Bettina
Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey (2013)
ACM Computing Surveys (CSUR) 46
DOI: 10.1145/2522968.2522977
- [22] TheBalanceSMB, 8 Popular Types of Web Hosting Services (2019)
Достъпно на <https://www.thebalancesmb.com/types-of-web-hosting-services-2532072>
- [23] Gabriel Gambetta
Fast-Paced Multiplayer (2010)
Достъпно на <https://www.gabrielgambetta.com/client-server-game-architecture.html>
- [24] CISA, Alert (AA20-133A)
Top 10 Routinely Exploited Vulnerabilities (2020)
Достъпно на <https://us-cert.cisa.gov/ncas/alerts/aa20-133a>
- [25] GamingXploits, Liam Bartlett
World of Warcraft Fast AFK XP Glitch (2020)
Достъпно на <https://gamingxploits.com/world-of-warcraft-fast-afk-xp-glitch/>
- [26] EveOnline
Drone damage exploit notification (2014)
<https://www.eveonline.com/news/view/drone-damage-exploit-notification>

- [27] ExtraCredit
MMO Economies - How to Manage Inflation in Virtual Economies (2014)
Достъпно на <https://www.youtube.com/watch?v=W39TtF14i8I>
- [28] Conor Stephens and Chris Exton
Measuring Inflation within Virtual Economies using Deep Reinforcement Learning (2021)
Достъпно на <https://www.scitepress.org/Papers/2021/103928/103928.pdf>
- [29] Machinations
What Is Game Economy Inflation? (2021)
Достъпно на <https://machinations.io/articles/what-is-game-economy-inflation-how-to-foresee-it-and-how-to-overcome-it-in-your-game-design/>
- [30] Daniel Terdiman, CNET News.com
Cheaters slam 'Everquest II' economy (2005)
Достъпно на
https://web.archive.org/web/20061128103439/http://news.zdnet.com/2100-1040_22-5829403.html
- [31] Samuelson, Paul A. and Nordhaus, William D.
Economics, Nineteenth edition (McGraw-Hill/Irwin, 2009)
THE THREE PROBLEMS OF ECONOMIC ORGANIZATION, Page 7
ISBN 978-0-07-351129-0
- [32] Cloudflare
What is a DDoS attack? (2017)
Достъпно на <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>
- [33] Yenala, H., Jhanwar, A., Chinnakotla, M.K. et al.
Deep learning for detecting inappropriate content in text. (2018)
Int J Data Sci Anal 6, 273–286
Достъпно на <https://doi.org/10.1007/s41060-017-0088-4>
- [34] С. Илиева, В. Лилов, И. Манова
Подходи и методи за реализация на софтуерни системи
(Ун. издателство „Св. Климент Охридски“, 2010)
ISBN 978-954-07-2999-2
- [35] Erik Bethke
Game Development and Production (Wordware Publishing, 2003)
ISBN 978-1-55622-951-0
- [36] Richard Rouse
Game design: theory & practice (Wordware Publishing, 2005)
ISBN 978-1556229121
- [37] Adele Caelia
Hero's Journey: A Visit to Simutronics (MMORPG.com, 2007)
Достъпно на <https://www.mmorpg.com/previews/a-visit-to-simutronics-2000115066>

- [38] Heroengine.com
About Hero's Journey (2021)
Достъпно на <https://www.heroengine.com/he-games/heros-journey-classic>
- [39] Hero Engine Official Documentation
HeroEngine and HeroCloud Tech Features (2021)
Достъпно на <https://www.heroengine.com/heroengine>
- [40] John Haas
A History of the Unity Game Engine (2014)
WORCESTER POLYTECHNIC INSTITUTE
- [41] Unity Official Documentation
Docs and guides to work with the Unity ecosystem (2021)
Достъпно на <https://docs.unity.com/>
- [42] Unity Official Documentation
Multiplayer networking in Unity (2021)
Достъпно на <https://docs-multiplayer.unity3d.com/>
- [43] UnrealEngine Official Documentation
Unreal Engine 4 Documentation (2021)
Достъпно на <https://docs.unrealengine.com/>
- [44] TheVerge
Minecraft still incredibly popular (2020)
Достъпно на <https://www.theverge.com/2020/5/18/21262045/minecraft-sales-monthly-players-statistics-youtube>
- [45] Statista.com
Number of active players of Minecraft worldwide as of May 2020 (2020)
Достъпно на <https://www.statista.com/statistics/680139/minecraft-active-players-worldwide/>
- [46] Tom Cheshire, Wired
Changing the game: how Notch made Minecraft a cult hit (2014)
Достъпно на <https://www.wired.co.uk/article/changing-the-game>
- [47] Apollo Colligan and Ryden Sar.
The History of Minecraft (2019)
Достъпно на <https://www.thescienceacademytmmagnet.org/2019/12/20/the-history-of-minecraft/>
- [48] Markus Persson
The World of Notch (2013)
Достъпен на <https://notch.tumblr.com/>
- [49] Minecraft Gamepedia
Java Edition Multiplayer Test (2021)
Достъпно на https://minecraft.gamepedia.com/Java_Edition_Multiplayer_Test

- [50] Intelligencer
What Is Minecraft, and Why Did Microsoft Just Pay \$2.5 Billion for It? (2014)
Достъпно на <https://nymag.com/intelligencer/2014/09/why-microsoft-paid-25-billion-for-minecraft.html>
- [51] TechRadar
The history of Minecraft
Достъпно на <https://www.techradar.com/news/the-history-of-minecraft>
- [52] PC Gamer
Minecraft - PC Gamer UK's Game Of The Year (2010)
Достъпно на <https://www.pcgamer.com/pc-gamer-uks-game-of-the-year/>
- [53] Paul McNamara, NetworkWorld
Minecraft PC/Mac sales top 10 million (2013)
Достъпно на <https://www.networkworld.com/article/2224420/minecraft-pc-mac-sales-top-10-million.html>
- [54] Polygon
Minecraft console sales surpass PC, 'almost 54M' copies sold in total (2014)
Достъпно на <https://www.polygon.com/2014/6/25/5843358/minecraft-console-sales-54m-copies-sold>
- [55] TheGuardian
Minecraft sold: Microsoft buys Mojang for \$2.5bn (2014),
Достъпно на <https://www.theguardian.com/technology/2014/sep/15/microsoft-buys-minecraft-creator-mojang-for-25bn>
- [56] MinecraftNet
MINECRAFT LIVE: VOTE FOR THE NEXT MOB (2020),
Достъпно на <https://www.minecraft.net/en-us/article/minecraft-live-vote-for-next-mob>

Речник на използвани термини и абревиатури

Бета версия – Ранна версия на софтуер, в която са завършени повечето планирани функционалности, но все още не е готова.

Бета тествър - Потребител, който получава достъп до бета версия на софтуер, с цел да валидира работата му и да открие слабости чрез използване на софтуера.

Бот – Вид софтуер, който е предназначен да извърши автоматични действия напълно автономно.

Двуфакторна автентикация – Метод за предоставяне на достъп до уеб сайт или друга апликация, в който потребителят трябва да представи две доказателства, че е този, за който се представя.

Експлойт - Слабост в даден софтуер, която може да се експлоатира за лична изгода.

Каскаден модел – Модел на софтуерен процес, който се характеризира със систематизиран и последователен подход към разработването на софтуер.

Компресиране – Процес, при който размерът на даден файл се намалява без загуба на информация.

Лаг - Осезаемо забавяне в играта.

Онлайн – Състояние на свързаност към мрежата.

Плъгин – Разговорна форма на думата „приставка“. Идва от английската „plugin“.

Прокси сървър – Вид сървърен софтуер, който служи като посредник между клиент и друг сървър.

Тик – Една итерация на игрови цикъл

Хостинг - Интернет услуга, която позволява на лица и организации да обслужват съдържание в Интернет.

Үпдейт – Процес на актуализация на софтуер с негова по-нова версия.

CAPTCHA – Автентикация, която има за цел да провери дали потребителят е човек.

JVM – Абревиатура за Java Virtual Machine. Виртуална машина, която позволява на системата, на която е инсталриана, да изпълнява Java код.

LTS версия – Абревиатура за Long Term Support. Версия на софтуер, която ще се поддържа дълго време.