

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Жағабаев Аңсар Махмутұлы

Unity қозғалтқышында, Oculus Quest виртуалды
шындық көзілдірігіне арналған жоба жасау

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. канд., ассистент-
профессор

 Н.А. Сейлова
« 28 » 05 2021 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Unity қозғалтқышында, Oculus Quest виртуалды
шындық көзілдірігіне арналған жоба жасау»


5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған

Жағабаев А.М.

Ғылыми жетекші

техн. ғыл. магистрі, лектор

 Аристомбаева М.Т.
« 28 » 05 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – Ақпараттық жүйелер

БЕКІТЕМІН

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. канд., ассистент-

профессор



Н.А. Сейлова

« 23 » 05 2021 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы Жағабаев Аңсар Махмутұлы

Тақырыбы: «Unity қозғалтқышында, Oculus Quest виртуалды шындық көзілдірігіне арналған жоба жасау»

Университет Ректорының 2020 жылғы «24» 11 №2131-б бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзім « 3 » 06 2021 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Берілген тақырып бойынша әдебиеттерге шолу кезінде жиналған мәліметтер, теориялық материалдар жинау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

1. VR жобалар аясын талдау және шолу;
2. Көріністер карталарын құру;
3. VR қосымшаны құру;

Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың _____ слайдтан тұратын презентациясы көрсетіледі. Ұсынылған негізгі әдебиет 9 кітаптан тұрады.

Дипломдық жұмысты даярлау
КЕСТЕСІ

Бөлім атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімі	Ескерту
VR жобалар аясын талдау және шолу	10.01.2021 – 08.02.2021	
Көріністер карталарын құру	05.02.2021 – 10.03.2021	
VR қосымшаны құру	11.03.2021 – 28.04.2021	

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдердің атауы	Кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған мерзімі	Қолы
БҚ жасау	Аристомбаева М.Т., лектор, магистр		
Норма бақылаушы	Дуйсенбаева А.Н. лектор, магистр		

Ғылыми жетекшісі _____ Аристомбаева М.Т.

Тапсырманы орындауға қабылдаған білім алушы _____ Жағабаев А.М.

Күні « _____ » _____ 2021 ж.

АҢДАТПА

Бұл дипломдық жұмыс садақ ату ойынына арналған. Жоба әр түрлі ортада садақ атуды үйренуді, жаттығуды қажет ететін пайдаланушыларға арналған қосымша.

Castle&Bow қосымшасы нақты уақыттағы даму платформасы Unity ойын қозғалтқышында, C# тілінде жүзеге асырылды. Виртуалды мазмұнды ойнататын құрылғы ретінде Oculus Quest 2 көзілдірігі таңдалынды.

Ойын кез-келген уақытта, қай жерде болсаңыз да, садақ атудан жаттығуға мүмкіндік береді.

АННОТАЦИЯ

Данная дипломная работа посвящена игре в стрельбу из лука. Проект-приложение предназначено для пользователей, которым необходимо научиться стрелять из лука в различных средах и обстоятельствах.

Приложение Castle&Bow реализовано на игровом движке, платформе разработки в реальном времени Unity, на основе языка C#. В качестве устройства для воспроизведения виртуального контента были выбраны очки Oculus Quest 2.

Игра позволяет тренироваться в стрельбе из лука в любое время, где бы вы ни находились.

ANNOTATION

This thesis is dedicated to the archery game. The project is an application for users who need to learn archery, practice in different environments.

The Castle & Bow application was implemented on the Unity Game Engine, a real-time development platform, in C#. Oculus Quest 2 glasses were chosen as a device for playing virtual content.

The game allows you to practice archery at any time, wherever you are.

МАЗМҰНЫ

КІРІСПЕ	9
1. Пәндік саланы зерттеу және талдау	10
1.1.Жобаның жалпы тұжырымдамасы және қолдану аясы	10
1.2.Жобаның (ойынның) өзектілігі	11
1.3.Қолданыстағы жобаларға шолу.....	13
1.4.Есептің қойылымы.....	14
2. Жобаны(Ойынды) әзірлеу	16
2.1.Функционалды талаптар	16
2.2.«Castle&Bow»	16
2.2.1. Жобаны іске қосу құрылғысы	16
2.2.2. Дамытуға арналған қозғалтқыш	18
2.2.3. Бағдарламалау тілі	20
2.3.Жобалау	22
3. Ойынның сипаттамасы	28
3.1.Көріністер және қолданушы интерфейсі.....	28
ҚОРЫТЫНДЫ	37
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	38
А ҚОСЫМШАСЫ	41

КІРІСПЕ

Виртуалды шындық (VR) - бұл техникалық құралдармен құрылған, адамға оның сезімдері: көру, есту, тұрту және басқалары арқылы берілетін әлем. Виртуалды шындық әрекетті, әрекетке жауаптарды да имитациялайды. Шындық сезімдерінің сенімді кешенін құру үшін виртуалды шындықтың қасиеттері мен реакцияларының компьютерлік синтезі нақты уақыт режимінде жүзеге асырылады.

Оқыту мен оқу тәжірибесін жақсартуға мүмкіндік беретін ойын тәсілдері мен виртуалды шындық (VR) орталарының бірігуі, сонымен қатар нарықта қол жетімді VR бағдарламалық жасақтамасы мен жабдықтарының кең қол жетімділігімен негізделген, келешек зор болашаққа ие. Пайдаланушылар пассивті бақылаушылардан гөрі, осы оқу орталарына белсенді қатысушы ретінде тартылып, зерттеуге негізделген оқыту парадигмаларын дамытуға мүмкіндік береді.

Виртуалды шындықты (VR) оқыту құралы ретінде пайдалану барған сайын танымал болып келеді. VR - бұл қолданушылар объектілермен әрекеттесе алатын және нақты ортада жүргендей қозғалатын иммерсивті сандық кеңістік. Виртуалды шындық технологиясы нақты әлемде оқыту логистикалық тұрғыдан қиын, қауіпті немесе практикалық емес жағдайларға қызығушылық тудырды. Мысалы, хирургтарға, ұшқыштарға және өрт сөндірушілерге арналған оқыту бағдарламалары виртуалды шындық ұсынатын шынайылық пен икемділікті пайдаланады. VR оқытудың негізгі жорамалы - алынған дағдылар мен тәжірибелер нақты өмірге ауысады.

Менің мақсатым - спортшыларға және кез келген қызығушылық танытқанға, ортаны ойын мәнерінде имитациялау арқылы садақтан атуға, жаттығулар арқылы үйрету және физикалық сипаттамаларын арттыру.

1 ПӘНДІК САЛАНЫ ЗЕРТТЕУ ЖӘНЕ ТАЛДАУ

1.1 Жобаның жалпы тұжырымдамасы және қолдану аясы

Спорттық жаттығуларға виртуалды шындықты қолданудың бірнеше артықшылығы бар.

- Бұл нәтиже салыстыру үшін адамдарға қажетті спорттық ортаға (мысалы, әр түрлі қашықтықтағы нысандарға) немесе бірнеше жаттығу серіктестеріне қол жеткізбей жаттығулар жасауға мүмкіндік береді.
- Спорттық жаттығуларға виртуалды шындықты белсенді енгізу пайдаланушыларға нәтижелерін тіркеуге және олардың дамуын мұқият қадағалауға мүмкіндік береді.
- VR масштабталатын және виртуалды ортаны әр түрлі тәсілдермен құруға және басқаруға үлкен еркіндік береді. Бұл артықшылықтар оқудан жоғары өнімділікке қол жеткізу үшін дәлелді әдістерді қолдануға үлкен мүмкіндік береді. Мысалы, бақылау-өткізу жүйелері сияқты дәлелді оқыту зерттеулері көрсеткендей, адамдар үнемі қиындықтарға кезіккенде дағдыларын жақсы дамытады. Бұл барлауды виртуалды ортаға оқытуға өзгергіштік қосу және қолданушының мүмкіндігіне негізделген қиындық деңгейін жүйелі түрде түзету арқылы енгізу оңай.

Оқыту ортасын жақсарту мүмкіндіктерін барынша арттыру үшін, қалай құрылымдауға болатындығын ескеру маңызды. Виртуалды шындықтағы бейсбол жаттығуларына көшуді тексеру үшін зерттеу жүргізілді. Шынайы әлемде доптан ұрудың қайталанатын тәжірибесін жасаған топ, виртуалды ортада доптан ұрудың қайталанған тәжірибесін жасаған топпен және виртуалды ортада адаптивті жаттығулар жасаған топпен салыстырылды. Зерттеушілер оқыту адаптивті болған кезде (жаттығу кезінде сәттілік деңгейіне бейімделген) тек қайталанатын тәжірибе алған екі топпен салыстырғанда нақты көрсеткіштер айтарлықтай жақсарғанын анықтады. Адаптивті оқыту адамның шеберлік деңгейі мен оқу тапсырмасының күрделілігіне байланысты өзгеріп отыратын оқытудың пайда болуы үшін өте жақсы ақпарат қажет деген тұжырымға негізделген. VR оқытудың мәні шынайы әлемдегі оқуды қайта құру емес, нақты өмірде құрылымы қиын дәлелді практиканың принциптерін енгізу болып табылады.

VR-ді зерттеу кезінде қатысушының тәжірибесі қатысудың құрылымымен анықталуы мүмкін. Қатысу туралы анықтама жиі талқыланады, бірақ әдебиеттегі жалпы тақырыптарды анықтауға болады. Қатысу - бұл жасалынған виртуалды ортада «бар болу» субъективті сезімі және осы делдал орта нақты әлем болған сияқты өзін ұстау және сезіну. VR-дегі жоғары деңгей VR модельдеуіндегі пайдаланушы тәжірибесін жақсартады. Бұрын жүргізілген зерттеулер қазіргі кездегі технологиялармен дағдыларды үйрену үшін VR-ді қолданудың нақты бағасы әлі зерттелмегенімен, қатысу деңгейі жоғарырақ болған кезде тапсырма мен жұмыс тиімділігі жоғары болатындығын көрсетеді. VR-дегі зертханалық зерттеулер кішігірім үлгілерді

тартуға бейім болғандықтан, бойлық дизайнды қамтитын бірнеше зерттеулер бар, біздің қатысу туралы түсінік тәжірибеден кейінгі қысқа мерзімді есептермен шектелген.

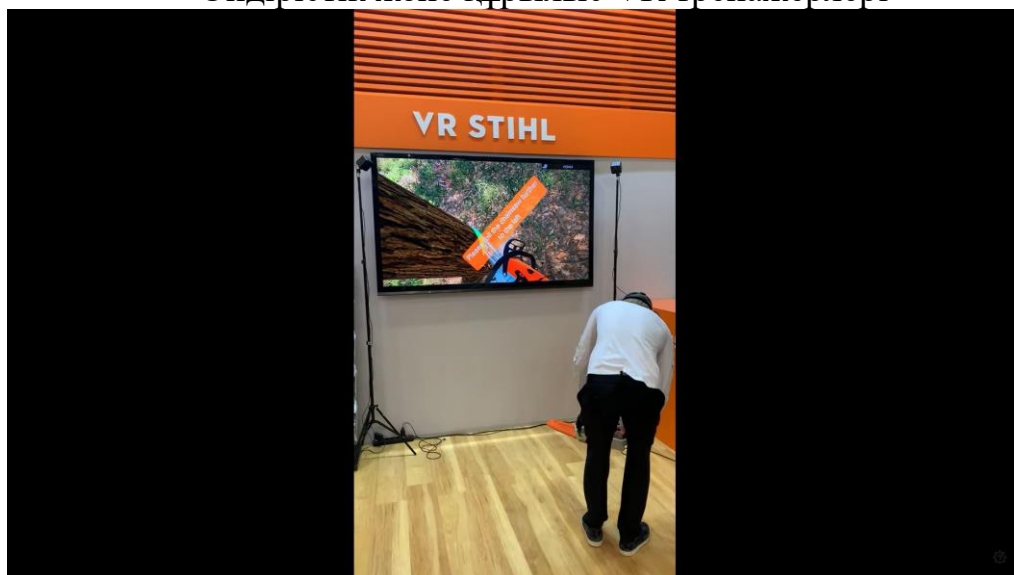
Болу тәжірибесіне көптеген айнымалылар әсер етеді. Виртуалды ортаның сенімділігі, яғни имитациялық мазмұнның шынайы әлеммен қаншалықты ұқсастығы, қатысуға ықпал ете отырып, виртуалды ортаның сенімділігін арттыруға мүмкіндік береді. Пайдаланушының виртуалды ортаға қатысу және оны өзгерту қабілеті немесе олардың интерактивтілік деңгейі қатысуға әсер етуі мүмкін. Сол сияқты, виртуалды экологиялық мазмұнға қатысу және қызығушылық туралы сілтеме жасай отырып, пайдаланушының қатысуы да қатысуға ықпал етеді. VR бағдарламалық жасақтамасы мен аппараттық құралдарының техникалық сипаттамаларының өзгеруі, болу тәжірибесіне үлкен әсер етеді. Техникалық факторларға дисплейдің объективті қасиеттері, навигация әдістері және пайдаланушы интерфейстері, сондай-ақ визуалды және дыбыстық сапа кіреді. VR жабдықтарының техникалық сипаттамалары виртуалды ортаның сенімділігіне, интерактивтілігіне және қатынасу деңгейіне әсер етуі мүмкін.

Қатысу сезімін арттыратын факторлардан басқа оны төмендететін факторлар да бар. Алаңдататын немесе сапасыз технологиялар мен интерфейстер қатысу сезімін азайтады. Кейбір VR модельдеуі пайдаланушыларда дезориентация немесе жүрек айну сияқты жағымсыз әсерлер тудыруы мүмкін. Бұл әсерлер иммерсивті технологиялармен жасалуы ықтимал және болу тәжірибесіне кедергі келтіруі мүмкін. VR қолданбасы мен бағдарламалық жасақтаманың өзгеруі олардың қатысуын қамтамасыз ету ролінде жеткілікті түрде қарастырылмаған. Мысалы, аппараттық немесе бағдарламалық жасақтаманың қандай белгілері қатысудың маңызды болжаушылары болып табылатындығы және технологияның / бағдарламалық жасақтаманың «заманауи» және қатысу сапасы арасындағы тікелей корреляцияны болжауға бола ма, ол жағы толық анық емес.

1.2 Жобаның (ойынның) өзектілігі

Виртуалды шындықта кез-келген жағдайды модельдеуге болады. Сондықтан ол білім беруде және басқа салаларда кеңінен таралуда. Бұл ең аз шығындармен және қажет болған жағдайда адамды әдетте материалдық шығындармен байланысты ортаға батыруға мүмкіндік береді. Виртуалды шындықтың басты артықшылығы - бұл ойын түріндегі қатысуды арттыру, қателіктердің салдарын визуалды түрде көрсету, пайдаланушының барлық әрекеттерін тіркеу және талдау мүмкіндігі. Әдетте, VR тренажерлері практикалық жаттығуларды толығымен алмастырмайды, бірақ олар практикаға дайын болуды арзан және жақсырақ етеді.

Өндірістік және құрылыс VR тренажерлері



Сурет 1 – STIHL компаниясының бензопила жаттықтырушысы

Өнеркәсіп пен құрылыстағы жұмысшылардың еңбегі әрдайым тәуекелмен байланысты. Жаңадан келгендердің тәжірибесіздігі және қауіпсіздік техникасын ескермеуі мезгіл-мезгіл қайғылы салдарға алып келеді. Қызметкерлерді олар үшін жаңа технологиямен жұмыс істеуге үйрету үшін оны тәжірибе үшін қамтамасыз ету керек, оны өндірістік процесстен алып тастау керек, және бұл көбіне мүмкін емес. Өндірістің барлық процесі тәуелді болатын жабдықтың істен шығуы жағдайындағы эвакуациялық іс-әрекеттер немесе іс-әрекеттер туралы айтудың қажеті жоқ.

Медициналық VR тренажерлері



Сурет 2 – Ассистентпен операция жасау жаттықтырушысы

Имитациялық медицина - бұл жыл сайын көп инвестицияларды қажет ететін сала. VR симуляторы тренажердің құнын бір муляжда қолдануға

болатын әр түрлі техниканың арқасында айтарлықтай төмендетуі мүмкін. Сонымен бірге оқушының дұрыс емес және дұрыс әрекеттерінің салдарын бейнелеуге, оның әрекеттерін есепке алуға және бағалауға болады.

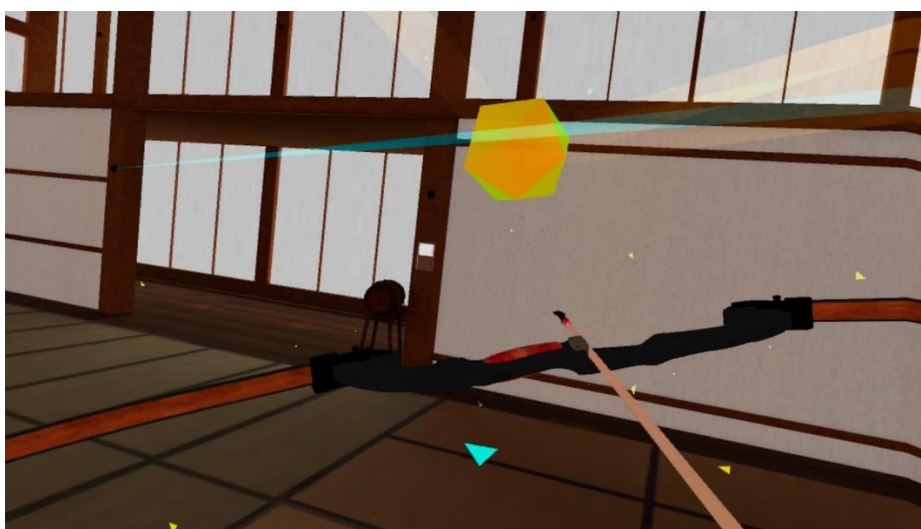
1.3 Қолданыстағы жобаларға шолу

Steam және Oculus ойың
аландарында ұқсас жобалар



Сурет 3 – Steam дүкенінің SteamLab жобасы

SteamLab-Valve корпорациясы әзірлеген және HTC Vive гарнитурасына арналған Microsoft Windows платформасында 2016 жылдың 5 сәуірінде шығарылған виртуалды шындық компьютерлік ойын. Ойын Портал әлемінде өтеді. Ойын ойыншыға виртуалды шындықты пайдалану мүмкіндіктерін және масштабты VR эксперименттерінің жиынтығын көрсетеді.



Сурет 4 – Oculus Store дүкенінің Holopoint жобасы

Holopoint - садақ атуда негізделген ойын. Жауапты нысандардың, самурайлардың және өте қауіпті ниндзялардың толқындары арқылы өту - осылардың барлығын атып салу, түйістіру және ату кезінде тез жарасыз айналып өту.

Ойыншының шеберлігі мен жаттығуына негізделген. Көптеген аркада стиліндегі ойындар сияқты, жеке жетістіктер мен жоғары балл - бұл мақсат. Бұл ойын тез қиынға түседі және белсенді ойлауды қажет етеді. Холопоинт бірнеше белді садақшылар ғана соңғы толқынға жететіндей етіп жасалған.

1.4 Есептің қойылымы

Тәжірибе керемет етеді. Мүмкін, бұл әйгілі фразада шындық бар шығар, кез-келген шеберлікті игеру, ол артқа орыннан секіру орындау, гитарада ойнау немесе теннис допымен ойнау жаттығуды қажет етеді. Бірақ практиканы жетілдіретін не? Ғасырдан астам уақытқа созылған зерттеулерге қарамастан, оқытудың әлеуетін барынша арттыру үшін оқытуды қалай құрылымдау керек деген пікірталас туындайды. Айқын нәрсе - спорттағы білікті болу үшін спортшылар жаттығуға көп уақыт бөлуі керек. Дегенмен, көбінесе оқытуды ұйымдастыру қиын, материалдық жағынан күрделі және өте қымбат болуы мүмкін. Мысалдар қарастырайық, садақтан атқанда адамға не бір тірі жанға тию мүмкіндігі бар не керекті алаң, бүкіл ойыншылар алаңынан өтуді дәл анықтауы керек квотербекті, қармен жауған тауға шығу керек шаңғышы және бос бәйге жолында көлікке жету керек жеңіл автокөлік жүргізушісі. Осы кедергілерді жеңу үшін заманауи, үнемді және икемді оқыту әдістерін табуға үлкен талаптар қойылды. Виртуалды шындық (VR) шынайылық пен икемділіктің арқасында нақты дағдыларды үйрету құралы ретінде қызығушылық тудырды.

Технология тез жетілдіріліп, қол жетімді бола бастаған кезде көптеген салалар мен кәсіптер VR-ді нақты дағдыларды үйрету құралы ретінде қолдана бастады. Хирургтар, ұшқыштар және өрт сөндірушілер сияқты әр түрлі мамандықтар, атап айтсақ, VR-де оқудың пайдасы тиетіні дәлелденді. Әдетте «оқу арқылы білім» қамтитын осындай оқыту бағдарламаларына байланысты тәуекелдің деңгейіне байланысты, VR-ге негізделген оқыту бағдарламаларына ауысудың өсу үрдісі байқалады. VR қауіпсіздікті, шынайы және интерактивті оқыту ортасын қайталап тәжірибе жасауға мүмкіндік береді, кері байланыс пен стандарттарды қолдайды, нәтижелілікті өлшейді. Дегенмен, VR-дің өз әлеуетіне жетуі үшін әлі де жол бар және одан әрі технологиялық жетістіктерге жету ол біздің болашақ құруымыздың тағы бір кілті.

VR авиация және хирургия сияқты салаларда барған сайын танымал бола бастағанымен, оны спорттық жаттығуларда қолдану әлі де шектеулі. Спорттық жағдайда VR-ге негізделген жаттығулардың көптеген артықшылықтары бар. VR-ді команда мүшелері мен қарсыластарының қатысуын модельдеу үшін

қолдануға болады, жаттықтырушыларға ойыншыларға арналған сценарийлер жасауға мүмкіндік береді, практика орындаушының шеберлік деңгейіне қатысты құрастырылуы мүмкін, пайдаланушылар олардың нәтижелерін тіркеп, олардың дамуын және көптеген спорттық орталарды мұқият бақылап отырады. модельдеуге болады. Қауіпсіз және қайталанатын ортада жаттығулар жасау мүмкіндігінің қосымша артықшылығы бар, VR спорттағы шынайы моториканы жақсартудың перспективалық алаңы.

Перспективалы бола тұра, VR жаттығулары барлық спорт түрлерінде жаттығулар жасау үшін пайдалы немесе практикалық емес екенін атап өту маңызды. Мысалы, суда жүзу сияқты спорт түрлерін VR-да оқыту мүмкін емес. Технологияның қазіргі күйі дәлдікпен, кері байланыс пен, мульти ойыншылар өзара әрекеттесуге негізделген дағдыларды оқытуды модельдеу қиынға соғады. Техникалық шектеулер мен виртуалды жаттығу орталарын құруға байланысты шығындар VR спорттық жаттығуларын қабылдауға айтарлықтай кедергі келтіреді. Дегенмен, кейбір спорт түрлері қазіргі уақытта өте сенімді және жоғары сенімділікпен имитациялау қымбатқа түссе де, VR бейнелері бақылауды үйрену үшін пайдалы болуы мүмкін екенін ескеру қажет. Пайдаланушылар өздерінің шеберліктерін демонстрацияны немесе өздерін батырған сезінетін нұсқаулық бейнені пассивті түрде көру арқылы арттыра алады.

Біздің қазіргі нақты жағдайымызда, адам үйде жұмыс істей алатын болса, сонымен қатар 2020 карантин бізге үлкен шектеулердің шынайы екенін көрсетті. Кофе-дүкендегі нақты физикалық қатысу, спортпен шұғылдану және тіпті жұмыс орнында болу әр адамның құзыреті. Дәл солай болды, немесе дәл осындай бұйрық қайталануы мүмкін, спортзалға бару үшін сізге заң бұзу немесе жасырын түрде бару және т.б. қажет пе? Немесе басқа нұсқа бар ма?

Бұл дипломдық жұмыстың басты мақсаты - сіз кез-келген жерде және кез-келген уақытта жанды тыныштықпен ойната білетін, садақ ату тәжірибесін молайта алатын әлемге батыра алатын ойын әзірлеу.

Жоспарланған нәтижеге жету үшін белгілі бір міндеттер қойылуы керек. Дипломдық жұмыста келесі негізгі міндеттерді орындау қажет:

- Жобаны іске қосу құрылғысың таңдау;
- Дамытуға арналған қозғалтқыш және жүзеге асыру әдісін таңдау;
- Бағдарламалау тілің таңдау;
- Пайдаланушының әрекеттерінің, мақсаттарының және қоршаған ортасының дәйектілігін жобалау;
- Ойынды дамыту;
- Достық қолданушы интерфейсін дамыту;

2 ЖОБАНЫ(ОЙЫНДЫ) ӘЗІРЛЕУ

2.1 Функционалды талаптар

Бұл ойын мобильді және стационарлық қосымша ретінде жасалуы керек. Бағдарламалық жасақтама Android және Windows амалдық жүйелері бар құрылғыларда жұмыс істеуі керек.

Бұл қосымшаның келесі мүмкіндіктері болуы керек:

- садақ ату
- нысандардың кездейсоқ қозғалысы
- жебенің тиюін тіркеу
- раунд санау
- көріністер арасында ауысу
- дыбыстық эффекттер

2.2 «Castle&Bow»

2.2.1 Жобаны іске қосу құрылғысы

VR платформасы нарығында қазіргі уақытта VR дамуының векторын белгілейтін екі негізгі алыптар бар. Бұл Steam жеке компаниясы және Oculus мемлекеттік компаниясы. Екеуінде де өзіндік дизайнғағы шлемдер, сондай-ақ VR бағдарламаларын сатуға арналған жеке дүкендер бар. Менің таңдауым Oculus-қа түсті.

Менің таңдауым 2-ші Quest түсті, іріктеу процесі өте оңай болды, өйткені Quest 2 гарнитурасының артықшылығы оның кемшіліктерінен және қалған гарнитуралардың барлық артықшылықтарынан басым болды.

Oculus Mobile OS қайта жаңадан Android 10-ға негізделіп жасалған және соның арқасында өте әсерлі көрінеді. Ондағы мобильді платформаны негізі алынғанын тану мүлдем мүмкін емес.



Сурет 5 – Oculus Quest 2 гарнитурасы

Oculus Quest-тің артықшылықтары:

1. Автономия

Бірінші Oculus Quest сияқты, екінші Quest-ті іске қосу үшін компьютердің мүлдем қажеті жоқ! Яғни, гарнитураның тікелей жобаларды салып қолданатын өзінің қосымшалар дүкені бар.

Oculus Quest 2 шлемінің бір заряды шамамен 2-3 сағатқа созылады. Ойын барысында ол сәл аз жұмыс істейді, шамамен 2 сағат, ал 3 сағат бейнені көру және қарапайым интерактивті қосымшаларды пайдалану арқылы жетуге болады. Батареяны зарядтауға шамамен 2,5 сағат кетеді, ал егер сіз розеткадан алыс болсаңыз, дулыға сыртқы батареяны пайдаланып зарядталуы мүмкін.

2. Салмағы мен мөлшері

Кім не айтпайды, бірақ VR тәжірибесін жинақтау үшін көзілдірік тағу керек, ал көптеген VR көзілдірік мойынға үлкен ыңғайсыздық, шаршау және ауырсыну әкеледі. Дәл осы Quest 2 көзілдірігінде мұндай проблема жоқ деуге болмайды, оған қарамастан бұл қазіргі кездегі ең тиімді шешім, өйткені шлемнің салмағы тек 503г құрайды. Ия, орташа салмақ, бірақ қалғандарымен салыстырғанда ешқандай дау-дамай тең дәрежеде болмайды. Сонымен қатар, оның көлемі үлкен емес болғандықтан, көптеген көрнекті құрылғыларға қарағанда тән айтарлықтай алдыға қатты тартылыс бермейді.

3. Бағасы

Бар болғаны \$300 үшін 4K экранды, алты дәрежелі еркіндікпен, контроллерлермен және дербес компьютерге кем дегенде кабель арқылы, тіпті толықтай сымсыз қосылу мүмкіндігі бар, толыққанды автономды шлем ала аламыз. Quest 2 пайда болғанға дейін, шлемде ұсынылған барлық мүмкіндіктері бар ең арзан VR шлемі бағасы \$700 басталған болатын, бұл бәріне көтерімді сома емес.

4. Өнімді қолдау және дамыту

Oculus Facebook-тің негізін қалаушы Цукербергтің компаниясы болғандықтан, ең болмағанда ірі компанияның қолдауы өте тартымды, өйткені сіздің виртуалды шындық көзілдірігіңіз бір жылдың ішінде пайдасыз, бөлме шетінде кірпіш болып қалмайды. Жіне барлық жаңа чиптер мен бағдарламалық жасақтама Oculus Quest 2 үшін арнайы дайындалған болады.

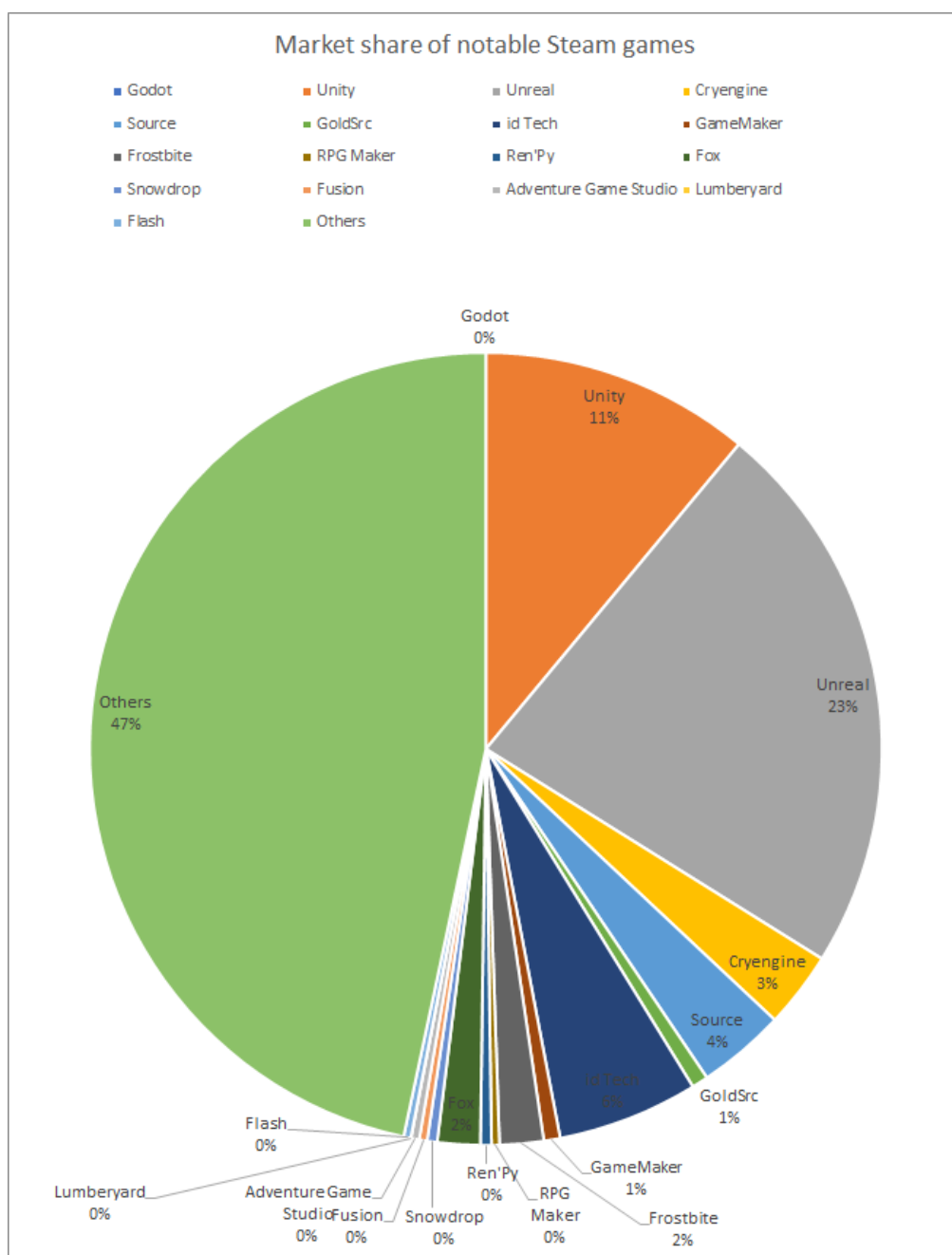
Кемшіліктердің ішінен тек бір пунктті қосуға болады:

1. Бұл әлі де жылжымалы көзілдірік болып саналады

Компьютерге қарағанда қосымшалардың сапасы төмендеуі болады. Мұны түсіну үшін компьютердің нұсқаларын және олардың Квестке бейімделуін салыстыру арқылы көз жеткізуге болады. Көлемі бойынша да олар бірнеше рет ерекшеленуі мүмкін, текстуралар қысқарады, жарық кесіледі және т.б. Бірақ, сонымен бірге, екінші квестке бейімделген нұсқа біріншісіне қарағанда жақсы көрінеді.

2.2.2 Дамытуға арналған қозғалтқыш

Қазіргі уақытта таңдау өте үлкен, сол себепті қозғалтқыш таңдауының көптігінде адасып кету оңай болады. Бірақ олда дұрыс көзқарас емес, себебі ойын индустриясында Electronic Arts, Ubisoft және т.с.с алпауыттар деңгейінде, барлығына өз ойының құруға мүмкіндік беретін екі мастодон бар. Олар Unity3D және Unreal Engine. Осы 3D ойынқозғалтқыштары бұрын VR дамыту тәжірибесі жоқ адамдарға, өнеркәсіп ардагерлерімен тең дәрежеде VR шешімдерін жасауға мүмкіндік береді. Бұл қиын таңдауда, менің көзім Unity-ға түсті, ендігі оның себептерін қарастырайық.



Сурет 6 – Steam ойын дүкенінде әр түрлі қозғалтқыштарда жасалған ойындардың нарықтағы үлесі

Неге Unity?

Unity, басқа "ойын қозғалтқыштары" сияқты, "бағдарламалық платформа" немесе әзірлеушілердің деректерді көрсетуді, физиканы және енгізуді жеңілдететін құралдар жиынтығы, оларды виртуалды кеңістікті нөлден құру қажеттілігінен босатады. Қозғалтқыш үш өлшемді (3d) және екі өлшемді (2D) виртуалды әлемдер үшін құрылыс блоктарын қамтамасыз етеді, бұл айтарлықтай міндет. Мысалы, біз қарапайым деп қабылдаған сансыз табиғат заңдарын қарастырыңыз - ауырлық күшінен бастап кез-объекттің реакциясына дейін. Виртуалды кеңістікте ойнатылатын әрекеттерді қайталауға мүмкіндік беру үшін, өмірдің осы негізгі аспектілерін ескеру қажет, ал ол мүлдем оңай емес.

2009 жылға дейін ойын қозғалтқыштарының көпшілігі меншікті болды және оларды компаниялар қатты күзеткен. Басынан бастап Unity бұл модельден алшақтап, әуесқойларға құралдарды тікелей ұсынды. Қозғалтқыштың өсуі «модинг» қозғалысының арқасында қарқын алды, энтузиастар ойыншыларға баспагерлер ұсынған әділ код негізінде ойындарды өзгертті.

Бұл ойын Unity ақылы бағдарламадан тек белгілі бір мөлшерде табыс тапқаннан кейін әзірлеушілерге есеп айырысатын бағдарламаға ауысуға мүмкіндік берді: қазірге уақытта ол \$100,000 мөлшерінде. Осылайша, олардың бизнес-моделі тәуелсіз және мобильді әзірлеушілердің ынта-жігері мен жұмысының сәттілігіне негізделген. Сонымен қатар, оның танымалдығы өскен сайын, қозғалтқыш жаңа нарықтарға, оның ішінде корпоративті бағдарламалық жасақтама мен цифрлық анимацияға ұласты.

Бұл компания жеткізгісі келетін имиджге сәйкес келеді: «бүкіл әлемдегі әзірлеушілерге бай, интерактивті, 2D, 3D, VR және AR тәжірибелерін құру құралдарын беру». Шын мәнінде, соңғы уақыттағы иммерсивті технологияның жоғарылауының себептерінің бірі аппараттық құралдардың арзандауымен қатар, бағдарламалық жасақтаманың үйлесімділігі мен қолданудың қарапайымдылығынан болуы мүмкін. Әзірлеушілер бірнеше құрылғыларда мазмұн құра алады және оны Apple App Store және Steam ойын кітапханасы сияқты арзан нарық алаңдары арқылы тарата алады, бұл коммерциялық VR бағдарламалық жасақтаманың көптігін тудырды. Сонымен қатар, Google, Facebook-ке тиесілі Oculus және Microsoft сияқты платформалардың алпауыттары мазмұн құруды жеңілдету үшін Unity-мен гарнитураларын құрастыру барысында бірікті.

Unity платформасының мүмкіндіктері

Unity басқа платформалармен салыстыруға болатын бірқатар мүмкіндіктеі бар. Біріншіден, ол үйлесімді. Бірнеше платформалар мен консольдарға жариялауға рұқсат ету үшін Unity «құрастыру және іске қосу» протоколын жасады, оған тек жүктеу және құрылғылардағы мазмұнды ойнатуды бастау үшін бар болғаны батырманы басу қажет. Код жаңа форматтарға арналған жиынтықтар үшін үнемі жаңартылып отырады.

Сонымен бірге бизнестің кеңеюі, өзінің деңгейлік нарығын ашуды да қамтыды. Олардың «Asset Store» әуесқойлар мен кәсіпқойларға басқа қолданушыларға қолдан жасалған көріністерді, кодтарды, қондырмалар мен аватарларды және т.с.с. қосалқы активтер жүктеуге (ақысыз немесе ақылы) және виртуалды кеңістікті толтыруға мүмкіндік берді.

Соңында, қозғалтқыш интраоперациялық кодты көп қолданады. Unity «пакеттерін» әзірлеушілер өздерінің мазмұнын басқа компьютерлерге оңай экспорттау үшін ғана емес, сонымен қатар жабдық өндірушілері жазған кодты импорттау үшін де қолдана алады. Пакеттерге әдетте «сахналар» немесе аппараттық және бағдарламалық жасақтама мүмкіндіктері модельденетін және бағаланатын виртуалды әлем параметрлері кіреді. Егер «Leap Motion» қол трекері сияқты жаңа құрылғы пайда болса, сахналар қозғалтқыш арқылы жүктеуге қол жетімді болады, сондықтан әзірлеушілер жабдықты тексеріп, оны ойындар мен қосымшалар жасау үшін қолдана алады.

Осылайша, Unity өзін таптырмас делдал ретінде көрсетеді. Ол виртуалды кеңістіктегі мазмұнды топтастырудан және қайта құрудан бастап, жариялау мен таратуға дейінгі өндірістің әр кезеңінде интраоперациялық платформалар мен нарықтар әлемімен өзара әрекеттестіреді, бұл оны платформаның символдық құралы етеді.

Әзірлеушілер қоғамдастығы

Жыл сайын әзірлеушілер XR-де мүмкін болатын шекараны алға шығарады. Unity3D және Unreal Engine екеуі де әзірлеушілерге тиісті құжаттама мен оқулықтарды ұсыну үшін барын салады, бірақ бұл салада сіз қай нұсқаны таңдау керектігін шешуге мәжбүр боласыз. Бұл жерде ойын қозғалтқышын қоршап тұрған қоғамдастықтың рөлі, әсіресе, жаңадан пайда болып жатқан XR индустриясында маңызды болады. Unity3D, еркін қол жетімді қозғалтқыш ретінде Unreal Engine 4-тен гөрі ұзақ болды, сондықтан оны қоршаған инди-әзірлеушілердің үлкен қауымдастығы бар. Мысалы, Unreal Engine форумында C++ бағдарламалау бойынша 12000 тақырып және VR және AR дамыту бойынша 4600 тақырып бар. Ал Unity3D форумында 128000-нан астам сценарий тақырыбы және 6100 астам VR дамыту тақырыптары бар. Бұл дегеніміз, егер проблема туындайтын болса, шешімін табуға көмектесу үшін сізде осындай проблема бар біреуді табу ықтималдығы жоғарылайды және жоба дамуын жеңілдетеді.

2.2.3 Бағдарламалау тілі

Ойынды дамыту ешқашан оңай болған емес. Unity сияқты ойындарды дамытуға арналған платформалар қарапайым 2D платформерлерінен бастап, жан-жақты егжей-тегжейлі 3D әлемдер мен VR-ға арналған әлемдерге деін жасауға мүмкіндік береді. Unity кішігірім әзірлеушілер үшін тегін, және редакторды өзіңіздің идеяларыңыздың прототипін жасау үшін, қалай қолдануға болатындығы туралы көптеген нұсқаулықтары бар.

Unity бағдарламасын қалай пайдалану керектігін білу қажет, бірақ та ойынның нақты өзегің, оның мінез-құлқын анықтайтын тек қана жазылған код болады. Ойынды дамыту үшін, қай тілді таңдау керектігі қиынға соғуы мүмкін, бірақ Unity жағдайында бұл оңай шешім.

Қазіргі уақытта Unity үшін жазуға болатын белгілі тілдер, ол:

1. C#
2. JavaScript
3. Boo
4. IronPython
5. Lua
6. C/C++
7. Rust

Ұсынылған 7 тілдің тек үшеуін ғана сеніммен қарастыруға болады.

1. C#

Unity-ден бастағандар үшін немесе объектілі-бағдарлы бағдарламалауды бұрыннан білетіндер үшін C# - бұл Unity-да жоба жасап үйрену үшін дұрыс тіл. Шын мәнінде, C# - бұл платформа үшін үйренуге тұрарлық жалғыз тіл деп саналады.

Себебі, Unity Microsoft .NET фреймворк-платформасы болып табылатын Mono-ны қолданады. C# .NET-тің негізгі тілі болып табылады және барлық Unity кітапханалары C# кодының көмегімен жасалған. C# - Unity тілі десек, артық айтқандық болмас. Және Unity олардың C#-ты болашақта қозғалтқышта қолданылатын жалғыз тіл деп санайтындығын анық көрсеткен болатын.

Бұл жағымды жаңалық, өйткені C# қуатты және үйренуге оңай тіл. Unity - C# тілінің оқудың көптеген жақсы себептерінің бірі, сонымен қатар жаңадан бастаушылардың барлығы, оны одан да қол жетімді деп санайды. Ойындарды дамыту оқудың құрылымын қамтамасыз етеді, ал жобаға негізделген мақсаттар жаңа пәндерді тереңірек түсінуге көмектеседі.

2. JavaScript

Unity сонымен қатар UnityScript ретінде белгілі JavaScript-ті қолдайды. Шығарылғаннан бастапқы JavaScript Unity-дің толыққанды даму тілі ретінде C#-пен қатар болды. Unity сценарийлері сілтемесіне кітапхананың көптеген элементтері үшін C# және JavaScript кодтарының мысалдары енгізілді.

Бұл JavaScript жасаушылары үшін пайдалы болды, өйткені код құрылымындағы айырмашылықтарға қарамастан, олар таныс синтаксисті қолдана алды. Алайда мәселе туындады.

UnityScript JavaScript-ке ұқсас болғанымен, олай емес. UnityScript-те JavaScript-те жоқ класстар бар. Бірнеше айнымалы декларациялар және қосымша үтірлер сияқты JavaScript мүмкіндіктері UnityScript-те қол жетімді емес.

Мүмкін, ең бастысы, Unity жобаларында JavaScript анықтамасын іздеу әрдайым түсініксіз болып келген, өйткені көптеген адамдар оны UnityScript емес, JavaScript деп атайды. Веб-дизайнның да, ойынның да нәтижелері

біріктіріліп, тілдер арасындағы айырмашылық таза JavaScript әзірлеушілері үшін дау туғызды.

Бәлкім, UnityScript-ті қолдауынан бас тартатындығы таңқаларлық емес, ендігі оның зейнетке шығу кестесі құрылған. JavaScript-ті Unity-де әліде қолдануға болады, бірақ оның қолдауының аяқталатынын білген соң, менің таңдауым айқын.

3. C/C++

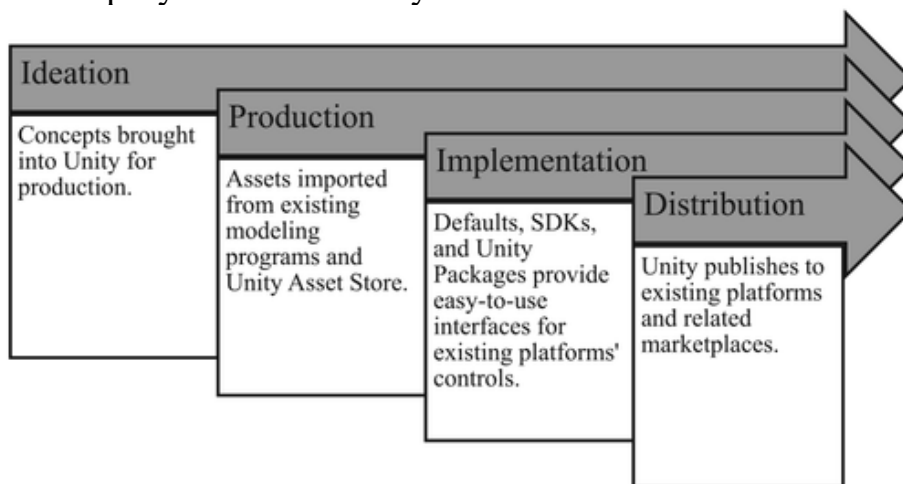
Бұл тілдерді қолдану, жылдамдық пен сенімділік маңызды жобалар үшін әдеттегі тәжірибеге айналды. Unity-дің мықты кітапханасына және C# ұсынатын барлық құралдарға қарамастан, кейде сізге өзіңіздің плагиндеріңіз қажет болуы мүмкін. Плагиндерді таңдаудың негізгі себептеріне жылдамдық немесе басқа тілде жазылған код негізіне кіру жатады. Бұл сценарийлерді DLL плагиндеріне ендіру кодты қайта өңдеу уақытын үнемдейді және кейбір жағдайларда өнімділікті жақсартады.

Көп жағдайда C++ плагиндер жасау үшін қолданылатын тіл болады, бірақ C мен бірдей жұмыс істейді. Код DLL-ге ендірілгенше, оны Unity плагиндері қалтасына орналастыруға және кодқа сілтеме жасау арқылы жетуге болады.

Unity-де жұмыс жасайтың болғандықтан, C# тілін үйрену қажет, бұл тілді елемей алмаймыз. C# тілінде оқыту мәлімет көздері және пайда болатын мәселелердің шешімдерінің мөлшері де әлдеқұрлым көбірек жазылған. Сондықтан таңдауым C#-қа түскен. Тек оған тоқталудың қажеті жоқ, платформа әр түрлі бағдарламалау тілдерін қолдайды, сондықтан әркім өзіне ұнайтын нұсқаны табады. Жалғыз ерекшелік – JavaScript-те. Unity командасы одан бас тартуға шешім қабылдады және болашақта қайтып келуі екіталай, сондықтан оны Unity жұмыс істеу үшін зерделеудің қажеті жоқ. Әйтпесе, ешқандай шектеулер жоқ, кез келген аталған тілдермен жұмыс жасай аламыз.

2.3 Жобалау

Ойынды құру Unity-да төрт кезеңнен тұрады: идеялар немесе «тұжырымдамалық кезең», өндіріс немесе «контентті құрудың 3D форматы», іске асыру және тарату немесе «босату».



Сурет 7 – Unity-да ойынды құру төрт кезеңі

Идея

Идея кезеңінде алғашқы түсініктер, эстетика, механика және деңгейлер “brain storm”-ын жасайды, содан кейін қозғалтқышқа енгізіледі. Пайдаланушы ойын ұсынған кезде де бірлік платформа құралы ретінде бекітіледі. Мазмұнның ішінара тұжырымдамадан тарауға ауысуының қарапайымдылығы оны қолданушылар үшін тартымды етеді. Unity-ді таңдаудың себептерінің бірі қарапайым сүйреу және түсіру функционалдылығында болды, яғни өнерді кодтау өте аз немесе мүлдем импорттауға болатындығын білдірді.

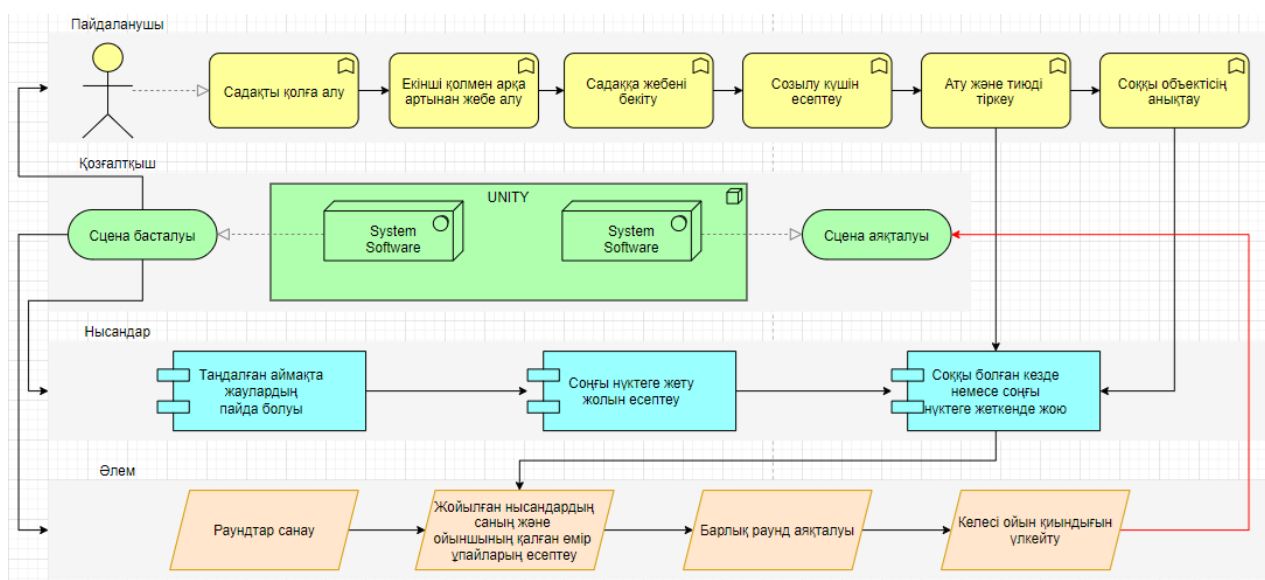
Идея деңгейінде виртуалды әлем жұмыс істейтін әлемнің логикасы (архитектурасы) ойластырылды. Төменде іс-қимыл тәртібінің диаграммасы берілген.

Бірінші тармақ - толыққанды жоспарланған ату үшін белгілі бір ретпен тағайындалған пайдаланушының әрекеттері. Әрекет тәртібін бұзу мүмкін емес, өйткені ол әрқашан басынан аяғына дейін өтеді.

Екінші тармақ - ойнатылатын көріністің басы мен соңын анықтайтын қозғалтқыштың өзі. Ресурстар мен жүктелген деректерді басқарады.

Үшінші тармақ – мишень (жаулардың) логикасы. Мишеньдердің рельеф картасында кездейсоқ пайда болуы керек, сонымен қатар жаңа жолмен, соңғы нүктеге дейін жүруі керек. Соңында раунд деректерін толтыру үшін бұзылады.

Төртінші тармақ - раунд реттілік логикасы. Раундқа тағайындалған жаулардың санын анықтайды, сондай-ақ зардап шеккен нысандар мен өткен раундтардың санын, соның ішінде ойыншының денсаулығын да есептейді.



Сурет 8 – Ойын қолданба архитектурасы

Өндіріс

Өндіріс деңгейдің дизайнымен қатар ойынның виртуалды кеңістігінде қолданылатын «активтерді» (таңбалар, объектілер және басқа материалдар) құруды қамтиды. Кейбір функциялар (мысалы, жарықтандыру, анимация және «қарабайыр» нысандар) қозғалтқышқа әдепкі бойынша енгізілген, бірақ Unity сонымен қатар Maya және Blender сияқты күрделі 3D графикалық қосымшаларымен интеграциялауды ұсынады. Ойындарды дамыта немесе модельдей білетіндер үшін бұл оларды үйлесімді бағдарламалық жасақтамада блоктайды. Жаңадан келгендер үшін Unity компаниясы түйреудің тағы бір жолын ұсынады: алдын-ала орнатылған активтер дүкені өндірісті жеңілдету үшін мазмұнды, плагиндерді, пакеттерді және басқа да утилиталарды сатады.

The screenshot displays the Unity Asset Store website. At the top, a blue banner reads "Save up to 95% on Mega Bundles to chart your course". Below this is a navigation bar with the Unity logo and "Asset Store" text, followed by menu items: "Assets", "Tools", "Services", "By Unity", "Industries", and "Sale". A search bar is present with the placeholder text "Search for assets".

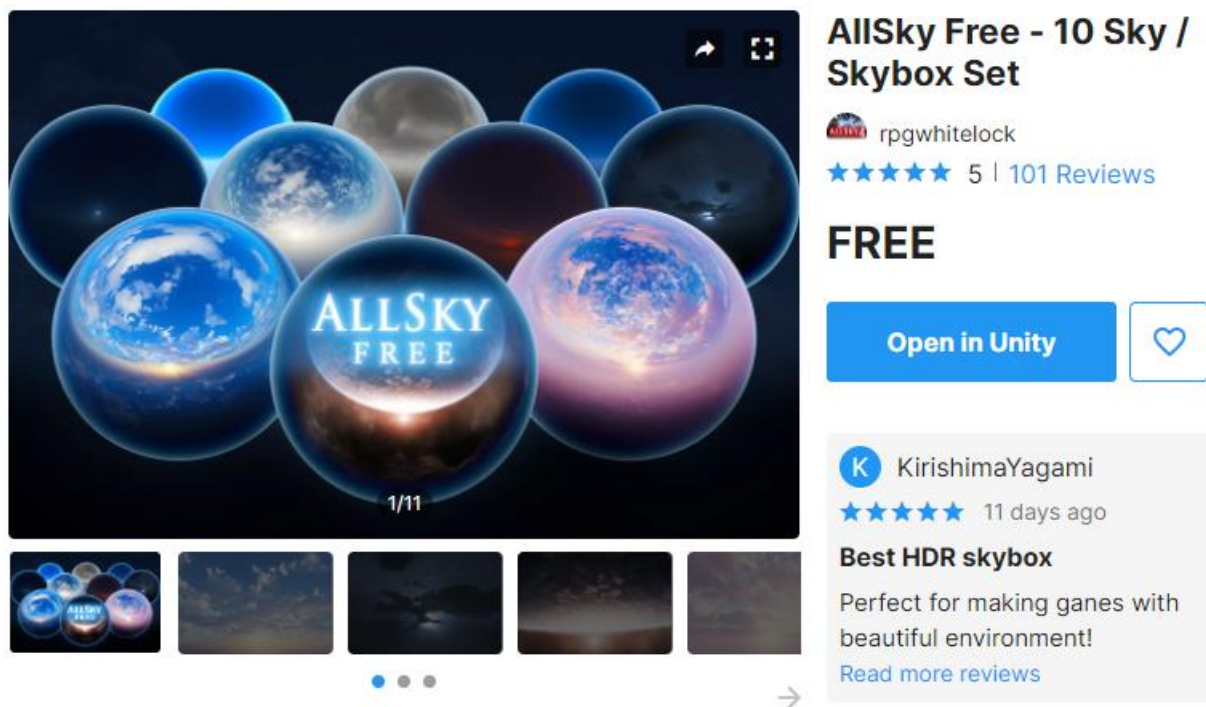
Below the navigation bar, three statistics are shown: "Over 11,000 five-star assets", "Rated by 85,000+ customers", and "Supported by 100,000+ forum members".

The main content area is titled "Top paid assets" and includes the subtext "See the top-ranked paid assets on the Asset Store." Below this, there is a section for "Unity Technologies" with a grid of asset cards. Each card features a thumbnail image, the asset name, the creator's name, a star rating with the number of reviews, and the price.

Asset Name	Creator	Rating	Price
MicroSplat - Ultimate B...	JASON BOOTH	★★★★★ (3)	\$180
A* Pathfinding Project ...	ARON GRANBERG	★★★★★ (715)	\$100
Odin - Inspector and S...	SIRENIX	★★★★★ (532)	\$55
Poly Art Animals Forest...	MALBERS ANIMATIONS	★★★★★ (26)	\$143
RayFire for Unity	RAYFIRE STUDIOS	★★★★★ (39)	\$170
Grabbit - Editor Physic...	JUNGLE	★★★★★ (30)	\$40
Easy Save - The Compl...	MOODKIE		
Final IK	ROOTMOTION		
Mega Animations Pack	KEVIN IGLESIAS		
DETAILED - Medieval C...	ASTROFISH GAMES		
Shapes	FREYA HOLMÉR		
Amplify Bundle - SAVE \$35	AMPLIFY CREATIONS		

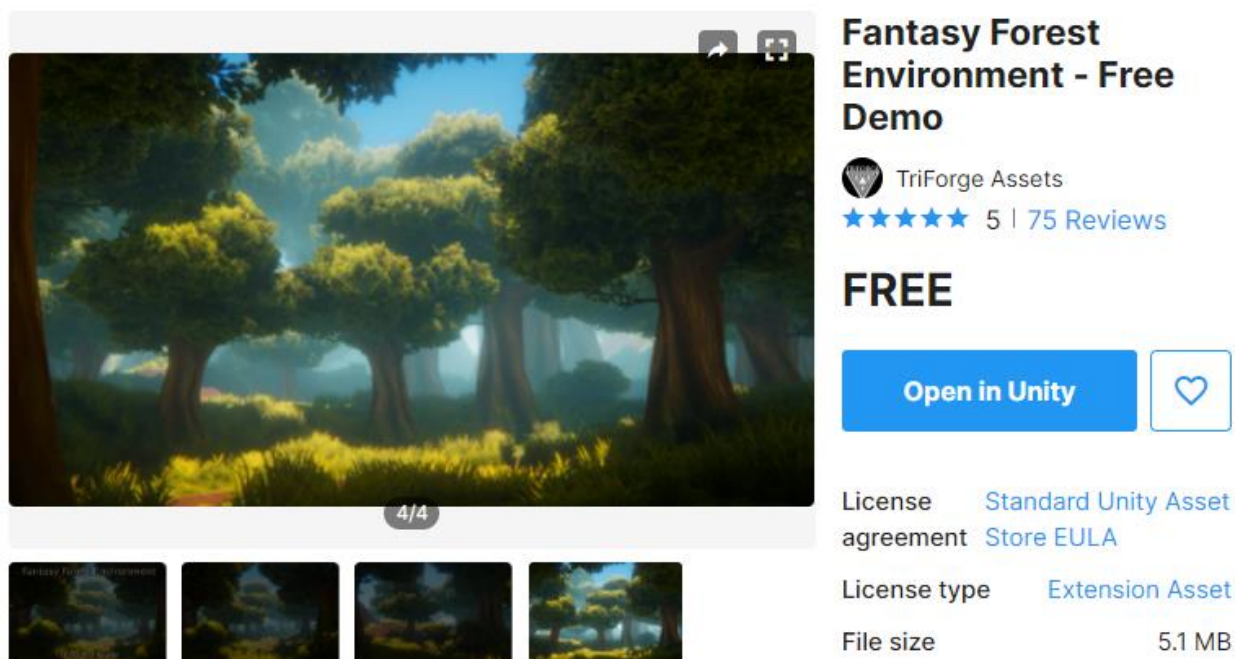
Сурет 9 – Unity Asset Store активтер кітапханасы

Қолданылған активтер тізімі



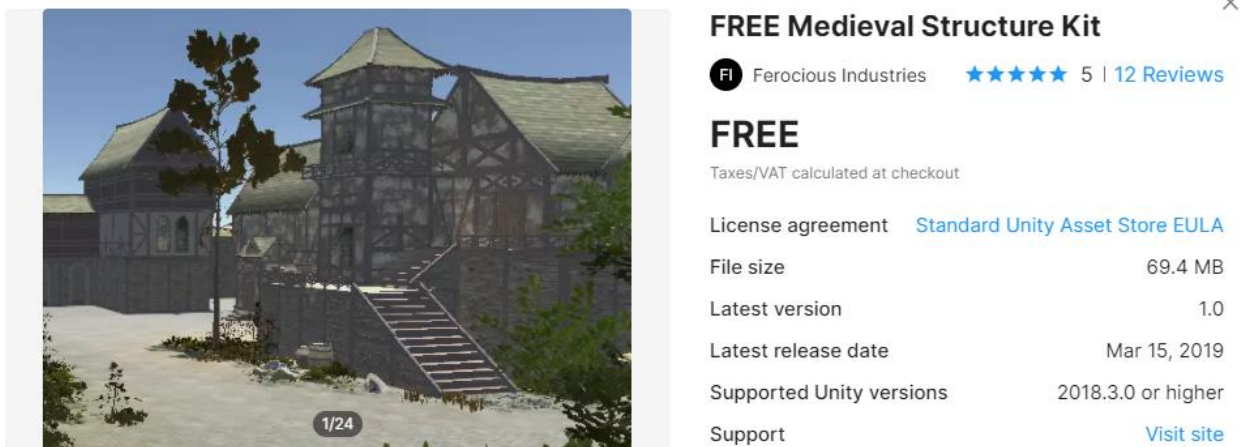
Сурет 10

AllSky Free - 10 Sky/Skybox Set - активтің аты айтып тұрғандай виртуалды әлемнің аспаның толтыру үшін қолдандым.



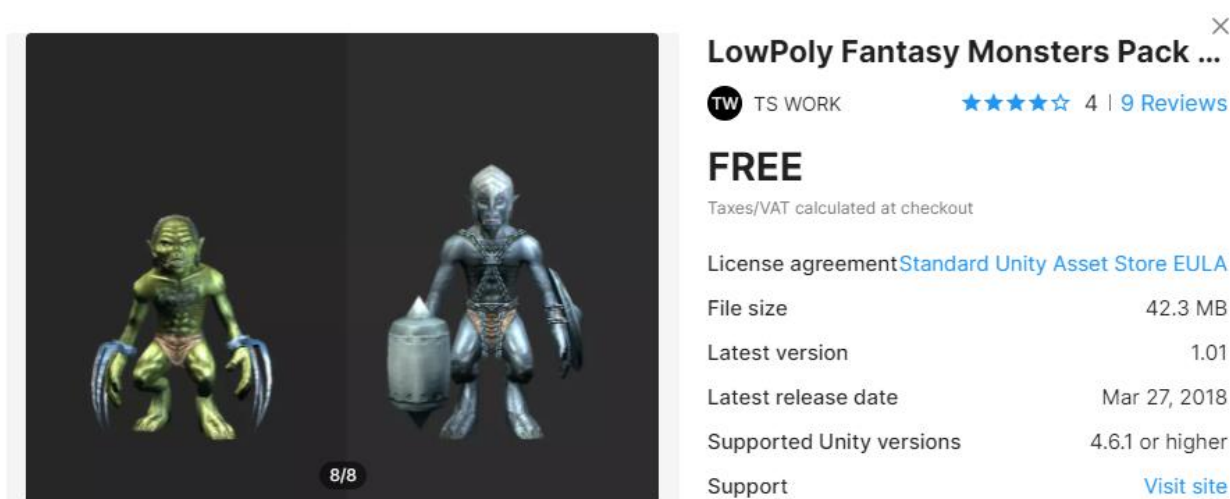
Сурет 11

Fantasy Forest Environment – активтің сахна әлемің ағаш және шөппен толтыру мақсатында алдым.



Сурет 12

Medieval Structure Kit – активі әлемді ортағасырлық ғимараттар мен толтыру мақсатында қосылған.



Сурет 13

LowPoly Fantasy Monsters Pack – бұл активті жау нысана орның толтыру үшін.

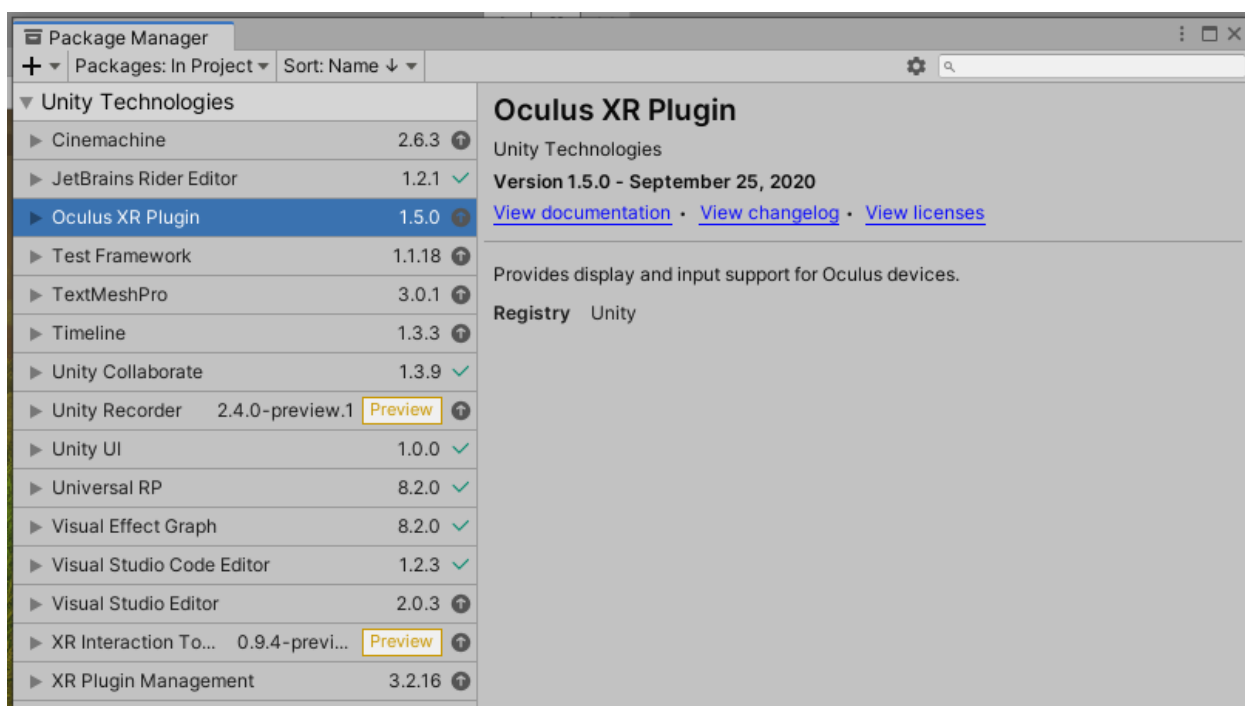
Іске асыру

Жүзеге асырылған кезде ойыншылардың басқару элементтері, активтермен және нысандармен өзара әрекеттесуі сияқтылар кодталады. Бағдарламалық жасақтаманы дамыту жинақтары (SDK), 8 API және пакеттер жобаны ойын басқару элементтерін және интерактивтіліктің басқа түрлерін қосу арқылы өмірге әкеледі. Бұл процесс көбінесе ойыншыны, басқа заттарды және кейіпкерлерді ойын ойнайтын қажетті жабдыққа сәйкес кодтауды

камтиды. Мысалы, VR навигациясы мен өзара әрекеттесуі үшін контроллерлер мен гарнитураларды қажет етеді. Бағдарламалауды екеуіне де жеңілдету үшін активтік дүкенде VR аппараттық құрал өндірушісі Oculus жасаған Oculus XR Plugin сияқты Asset Store дүкенінде ресурстар бар. Ол сонымен қатар пайдаланушыларды белгілі бір платформаларға арналған жобалар жасауға мәжбүр етеді. Басқа жабдыққа ауысу үшін олар мазмұнды жаңартуы керек, көбінесе басқа бағдарламалық жасақтама пакеттерін қолданады. Unity мұндай кодты өзінің қозғалтқышында стандарттауға тырысады, бұл оны аппараттық жасаушылар үшін одан да маңызды етеді.

XR жобаны әзірлеу, маңызды плагиндарсыз мүмкін емес, оларды пайдалану тәртібін әр компания Unity-мен қосылып орнату құжаттамасында жазады, сонымен қатар Oculus құрылғысын біріктіру үшін оған арналған қосымша жазу қажет. Oculus гарнитурасын және оның контроллерлерің байлау, негізгі үш плагині:

- Oculus XR Plugin
- XR Interaction Toolkit
- XR Plugin Management

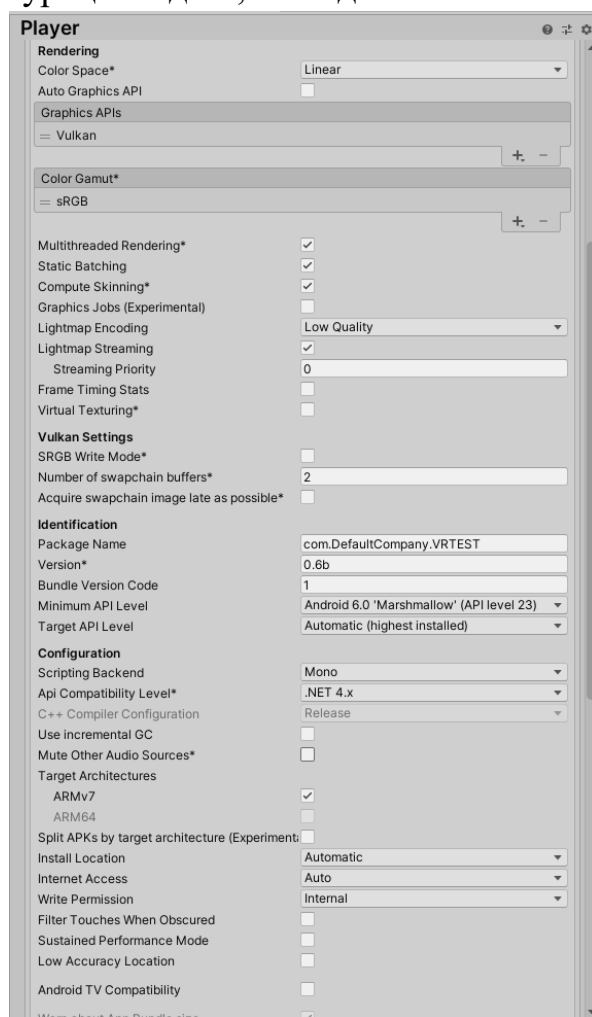
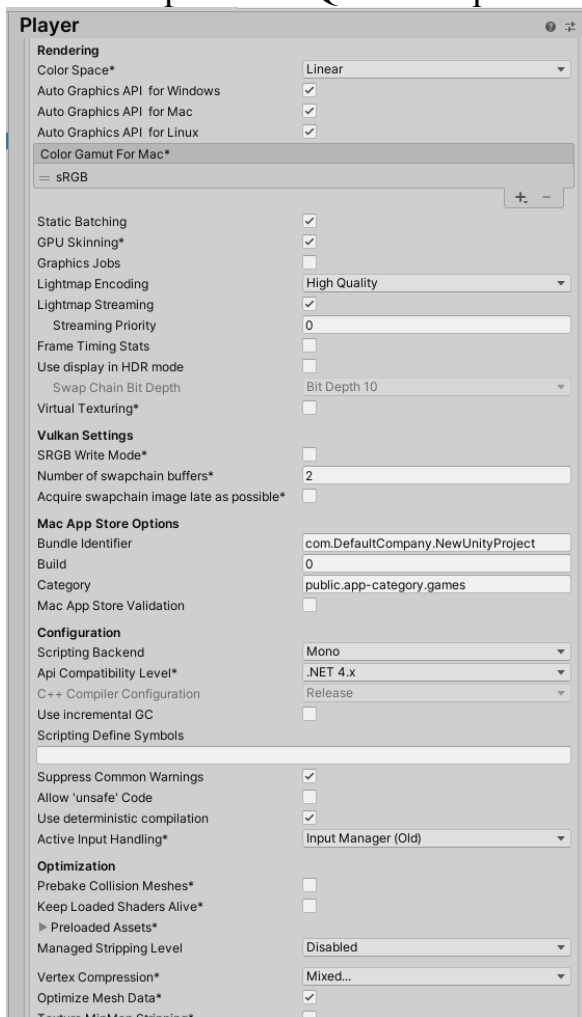


Сурет 14 – Плагиндер басқару терезесі

Тарату

Бұл этапта ойын ортасы, код және ресурстары ұялы телефондар, компьютерлер, консольдар және т.с.с. үшін компиляцияланады. Unity жағдайында тарату «құрастыру және іске қосу» функциясы көмегімен жүзеге асырылады. Әзірлеушілер үшін Unity ойындар, VR және басқа да 3D мазмұнын шығаруға арналған аренаардың төрешісі.

Ойынды барынша көп платформада қосу мүмкіндігін беру үшін, жобаны компьютерге және Quest-ке арнап конфигурациялады, жинадым.



Сурет 15 – WindowsOS

Сурет 16 – AndroidOS

Сурет 15 - компьютерде Oculus гарнитурасында жобаны іске қосу үшін негізгі құрастыру параметрлері. Windows операциялық жүйесіне арналған.

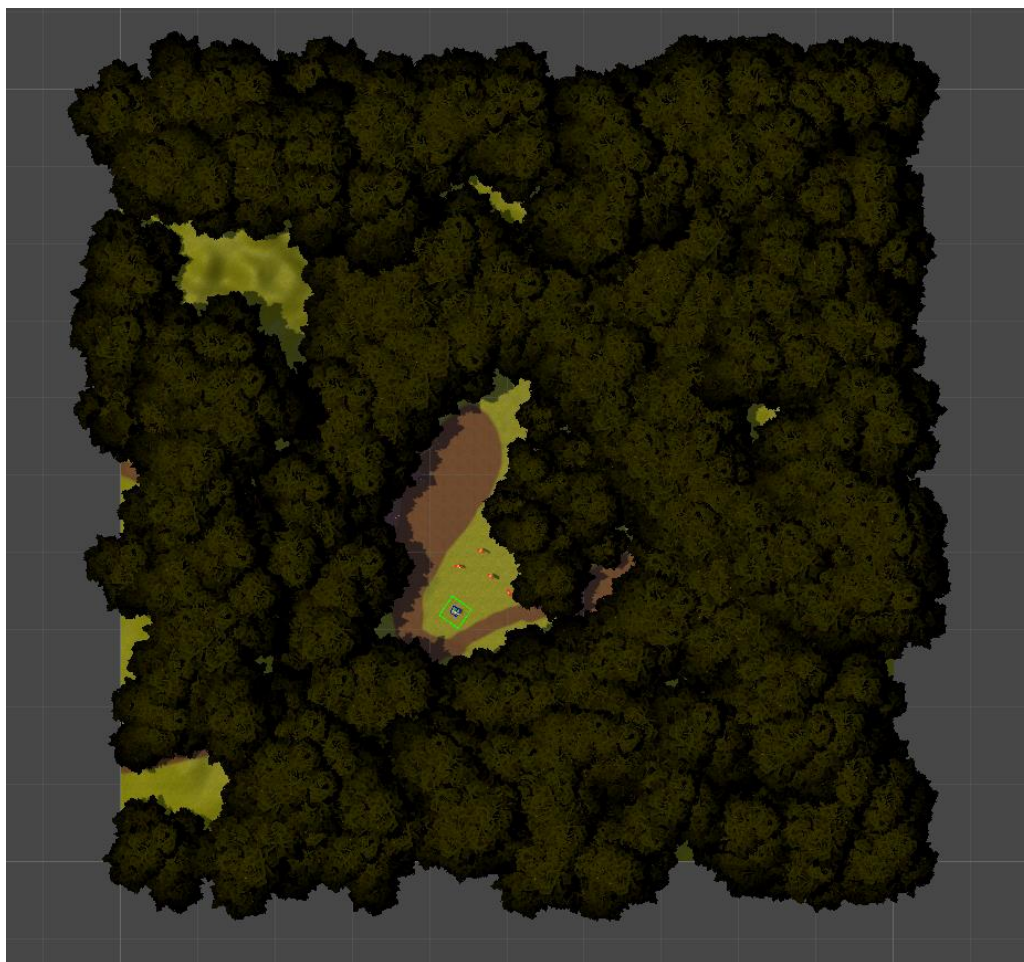
Сурет 16 - Quest көзілдірігіне арналған конфигурация жиынтығы. Android 8.0-ден үлкен версиялы мобильді операциялық жүйесіне арналған.

3 ОЙЫННЫҢ СИПАТТАМАСЫ

3.1 Көріністер

Бірінші көрініс - бұл мәзір сахнасы, негізгі экранды кәдімгі қосымшадағыдай ауыстырады, тек еркін қозғалу және ату үшін бірдей мүмкіндіктер береді.

Төменде пайдаланушының виртуалды ортамен алғашқы танысуы үшін мүмкіндігінше жеңілдетілген бастапқы көрініс картасы берілген, сондықтан ол триггерлер мен түймелерді түртіп, қарап, гарнитураның ішіндегі қозғалысқа үйренеді. Бұл жаңа ойыншылар үшін түсініксіз әрекеттерден қорықпау үшін жиі кездесетін тәжірибе.



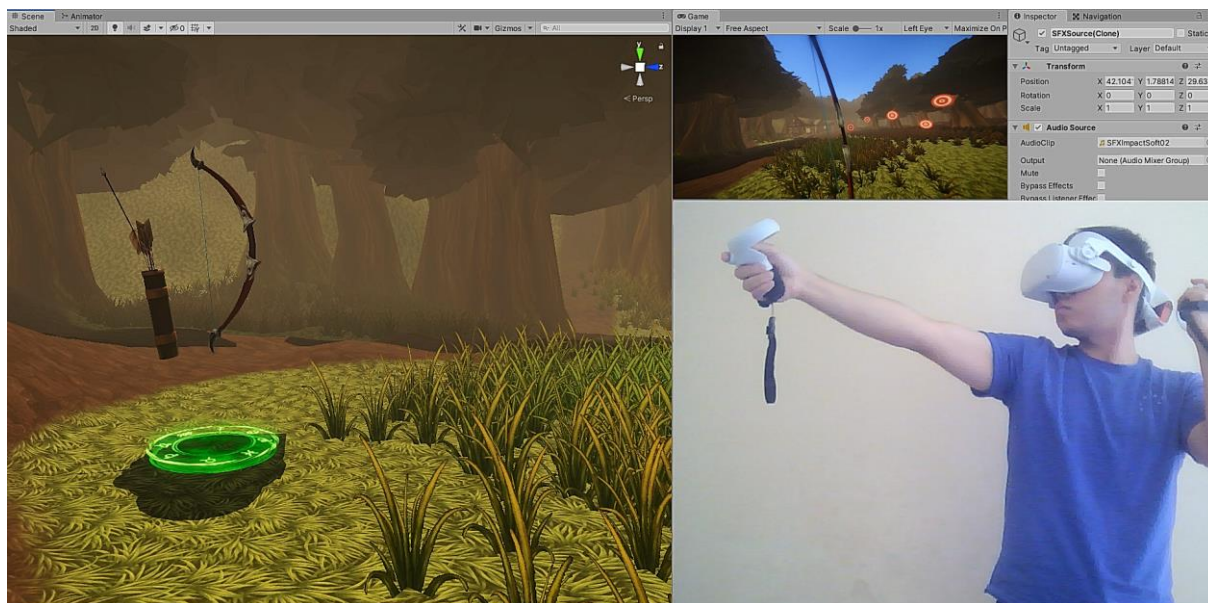
Сурет 17 – Бірінші көрініс әлем картасы

Әлем орманның бір бөлігі түрінде жасалған, онда сіз жаттығуға, айналаңызды зерттеп, ойын бастай аласыз. Бұл көріністе нысанға дейінгі қашықтықты анықтауға және соққының шамамен бұрышын есептеуге арналған 5 әр түрлі қашықтықтағы нысана бар.

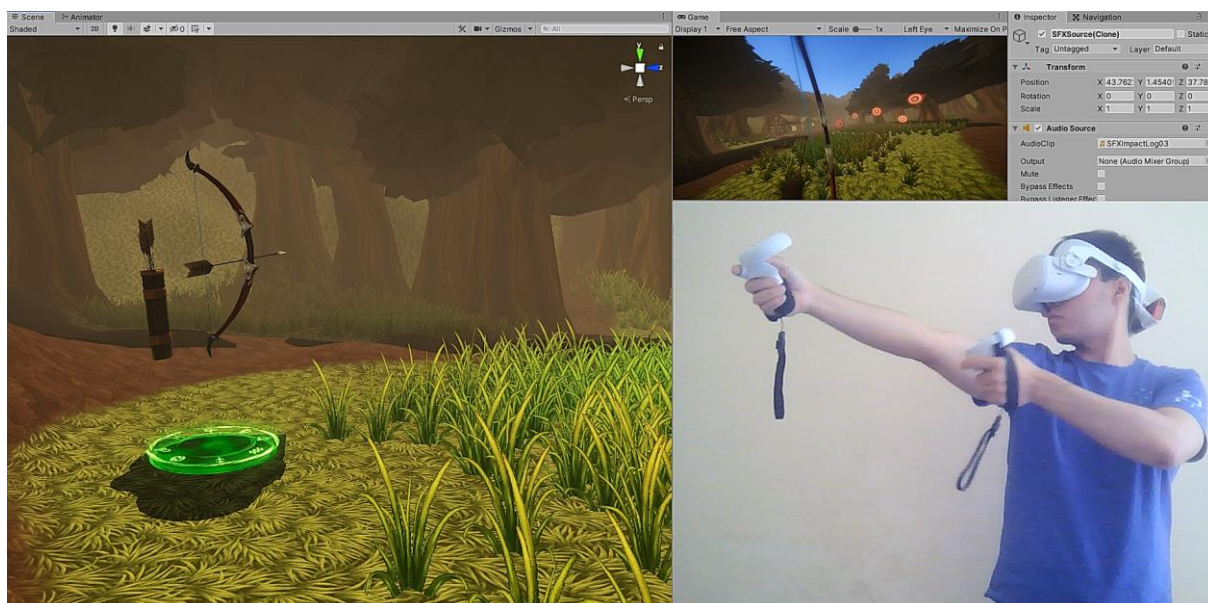


Сурет 18 – Бірінші көрініс, садақтан атушы мен мишень

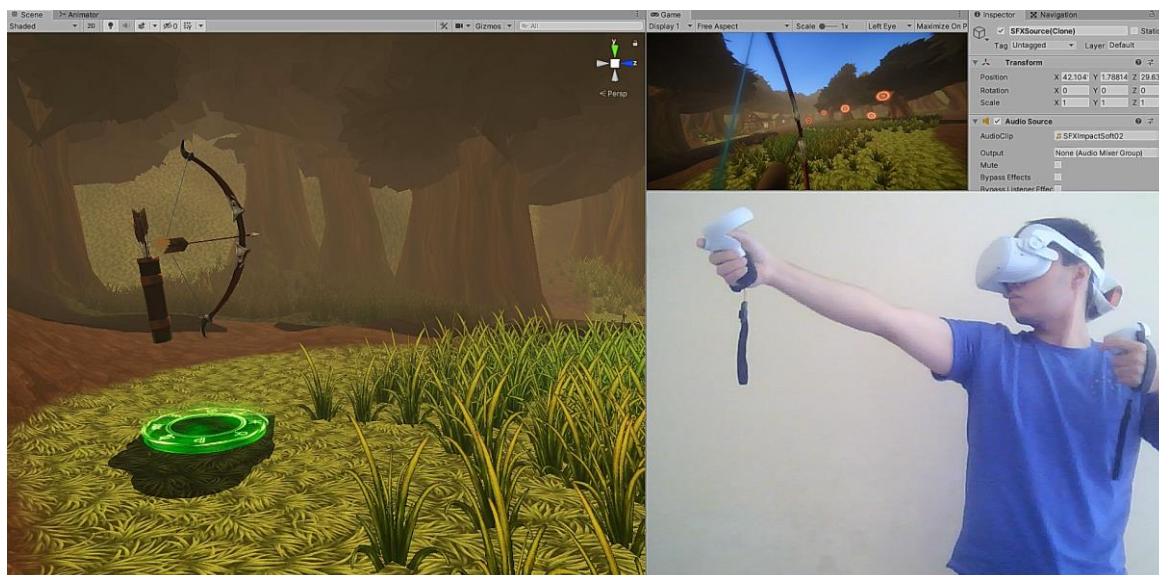
Алтыншы суреттегі қолданба архитектурасының бірінші тармағының, іс-әрекеттегі көрінісі. Ойыншы алдымен қолчаннан жебені алады, содан кейін жебе садаққа бекітіледі, кейін адырна босатылып, жебе садақ кеңістігіндегі позиция бағытында ұшады.



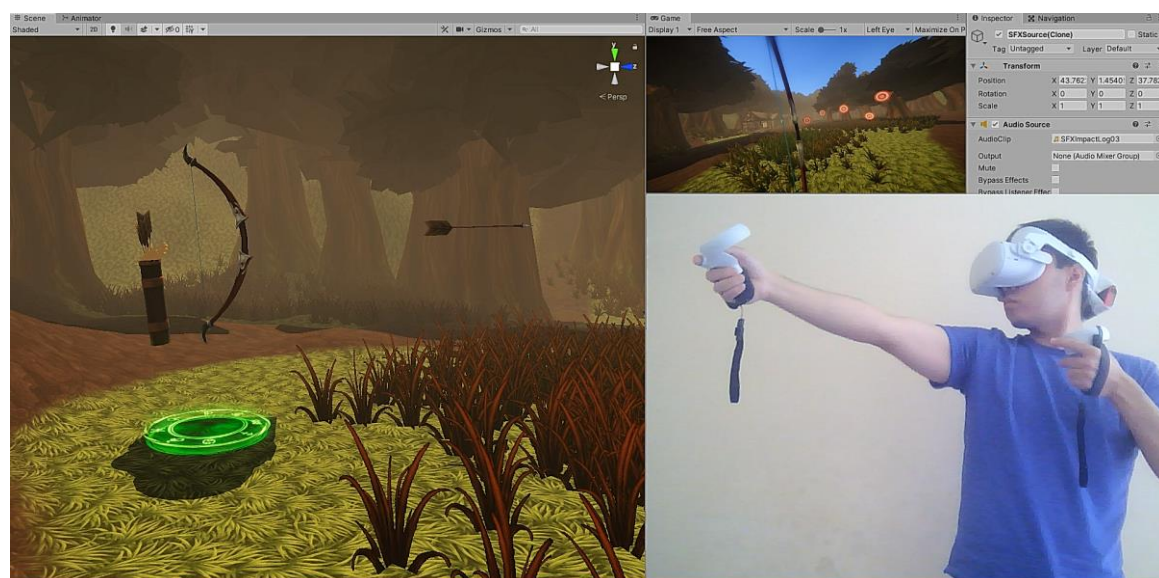
Сурет 19 – Қолчаннан жебе алу



Сурет 20 – Жебені садаққа тіркеу



Сурет 21 – Адырнаны керу



Сурет 22 – Адырнаны жіберу, садақтан ату

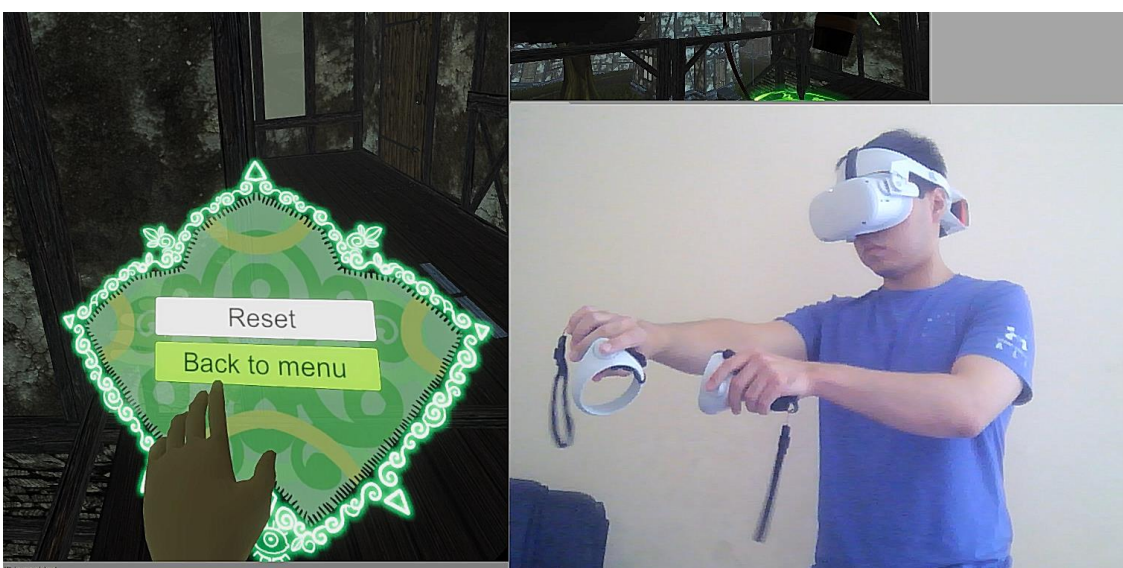
Көріністер арасында жылжу үшін кішкене қалқымалы мәзір (пайдаланушы интерфейсі) қосылды, ол бастың оң білегіңіздегі сағаттарға бағытталған кезде ашылады. Ойыншы орналасқан көріністі қайта қалпына келтіру үшін бірінші тастау "Reset" түймесін басу керек. Екінші батырма ойнатқыш орналасқан көрініске байланысты өзгереді: біріншісінде - "Start Game" ойынын бастау, екіншісінде - "Back to menu" мәзірге немесе бірінші көрініске оралу.



Сурет 23 – Қалқымалы мәзір



Сурет 24 – Ойынды бастау батырмасын таңдау



Сурет 25 – Ойыннан бастапқы бетке көшу батырмасын таңдау

Екінші көрініс - садақшының қамалды өлместіктен қорғауының көрінісі, мұндай ойындар үшін "Tower defence" аталатын жанрының бөлек тармағы бар. Сахна жаулар сіз арқылы өтуге тырысатын раундтардан тұрады. Содан кейін, егер сіз өмір ұпайларың жоғалтпай тірі қалсаңыз ойын жеңіспен аяқталады, әйтпесе барлық денсаулық ұпайларын жоғалтасып ойын жеңіліспен аяқталады.

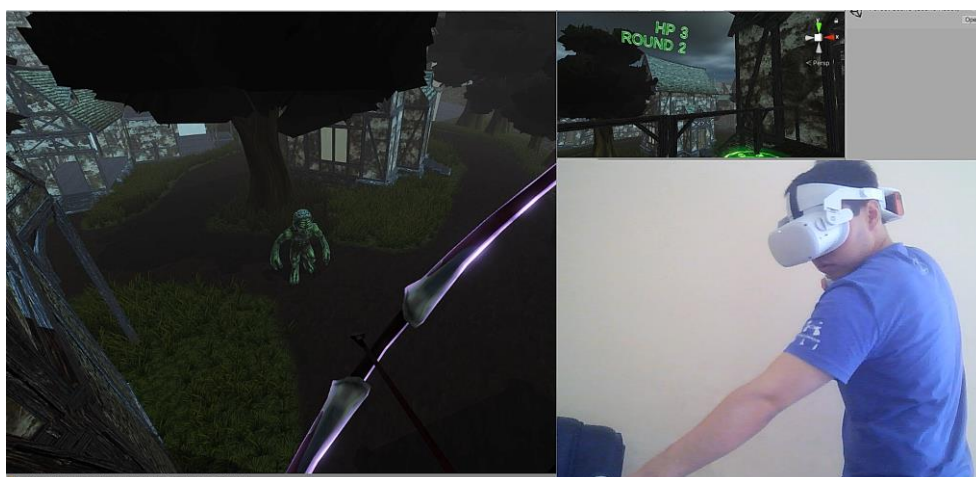


Сурет 27 – Екінші көрініс әлем картасы

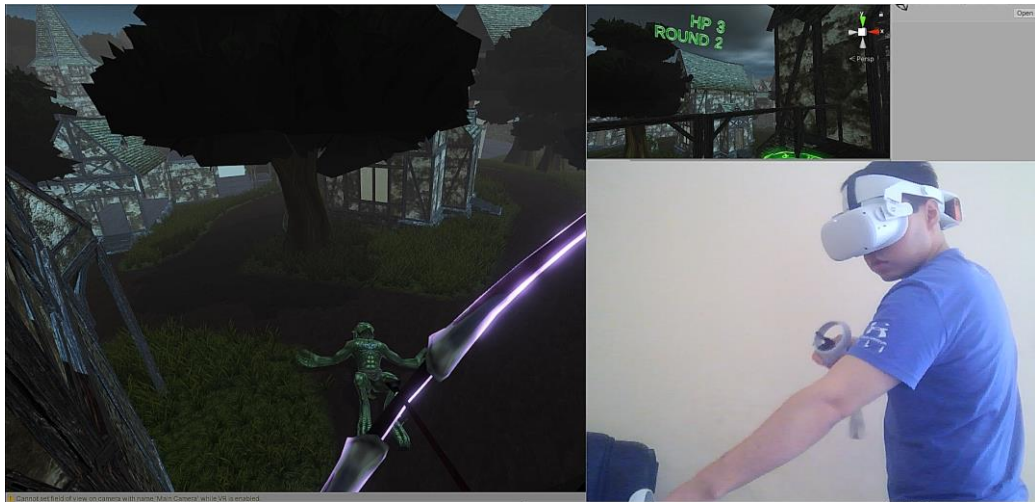
Картадағы жауларды жылжыту үшін жасанды интеллект ойынды әртараптандыру үшін қолданылады. Картада жауларға жылжуға болатын арнайы сценарий бойынша орындар анықталған, олар 28 суретте көк түспен боялған. Қызыл түспен екі координатамен анықталған аймақ белгіленген, олар өз кезегінде картаға қызыл шаршы салады. Бұл аймақта жаулар кездейсоқ пайда болады және жасыл нүкте бағытында жүреді. Қысқа жолды табу, кедергілерді айналып өту барысында ЖИ арқасында жол салынады және жаулар басқарылады.



Сурет 28 – Екінші көрініс картасында, нысандар қозғалуы. Көк - нысандар қозғала алатын алаң; Қызыл - нысандар кездейсоқ пайда болатын аймақ; Жасыл - нысандар жету қажет соңғы нүктесі



Сурет 29 – Нысанды көздеп ату



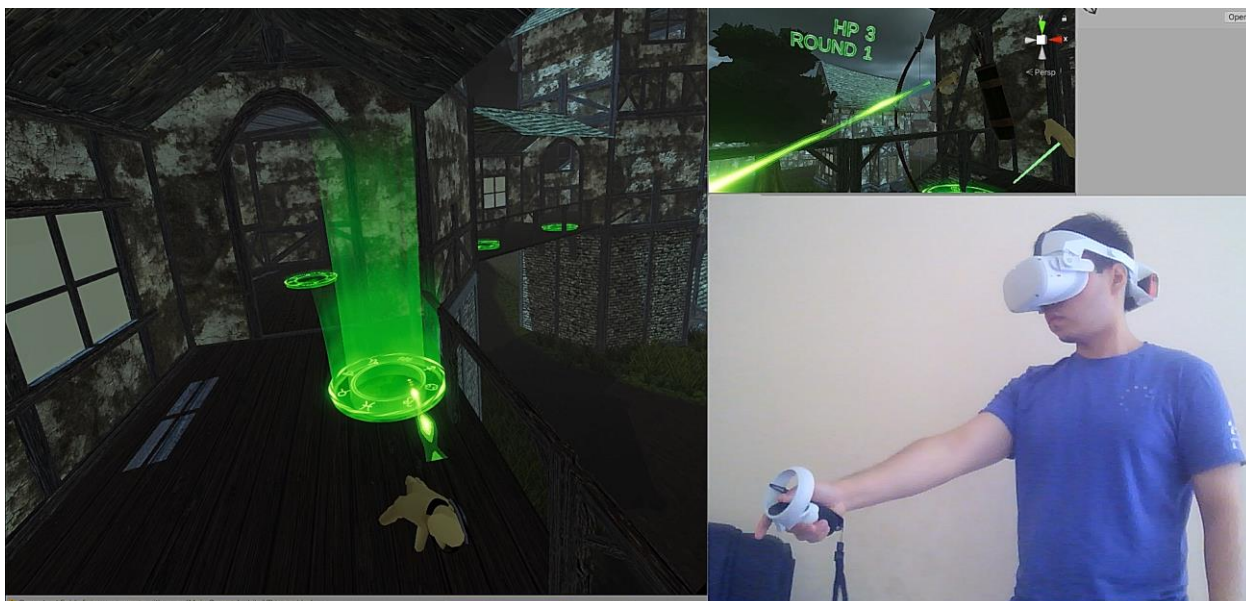
Сурет 30 – Нысанның түсуін және жойылуын тіркеу

Денсаулықты бақылау және раундтарды санау үшін аспанда сипаттамасы бар екі жол түріндегі интерфейс қосылған.



Сурет 31 – Раундтарды және ойыншының денсулығын қадағалау пудаланушы интерфейсi

Сонымен қатар екінші көріністе ойыншы тұрған орның ғимаратты қорғау үшін ауыстыра алады. Мекенің ауыстыру үшін қолың ауысу зәкіріне қарай бағыттап батырманы басу қажет.



Сурет 26 – Орын ауыстыру нүктесін таңдау



Сурет 27 – Контреллер

1 – орын ауыстыру джойстигі; 2 – затты ұстау батырмасы;
3 – іс-әрекет қолдау батырмасы

ҚОРЫТЫНДЫ

Бұл дипломдық жұмыс кез-келген деңгейдегі пайдаланушының жаттығуға, садақтан атуға үйренуің ойын-сауық түрінде келтіру мақсатында жасалды.

Садақтан ату, кездейсоқ жерде жаулардың пайда болуы, жаулардың қозғалысы, раундтар арасында жылжу параметрлерін есептеу және ойын нәтижесін анықтау алгоритмдері жасалды. Жоба барысында бұл әрекеттер қолданбалы архитектураның диаграммасында сипатталған және көрсетілген.

Ойынды дамыту үшін қозғалтқыш ретінде Unity компьютерлік ойындарының платформааралық даму ортасы және C# бағдарламалау тілі таңдалды. Unity-де көптеген платформалар мен жанрларды қамтитын мыңдаған ойындар, қосымшалар, математикалық модельдердің визуализациялары жазылған. Unity-ді ірі әзірлеушілер де, тәуелсіз студиялар да пайдаланады. Сонымен бірге, C# тілі бағдарламашылар арасында өте танымал және оны Microsoft алып компаниясы қолдайды. Демек, бұл фактор осы тілде және қозғалтқышта жазылған қосымшалар ұзақ уақыт бойы өзекті болатынын анық көрсетеді.

Жұмыстың мақсаты нақты әлемнің белгілі бір физикалық мүмкіндіктері болуы керек қосымшаны құру болды. Осы қосымшаға қойылған талаптарға сүйене отырып, жүйенің техникалық және бағдарламалық бөлігі жасалды.

Сондай-ақ, осы қосымшаны құруда негізгі рөлді төрт тармақтан тұратын қолданба архитектурасы ойнады деп санаймын. Бұл кезеңде негізгі идея қаланды және активтер таңдалды, ойын құрудың барлық кейінгі процесі барысында оларға баса назар аударылды.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- [1] Виртуальная реальность в Unity. / Пер. с англ. Рагимов Р. Н. – М.: ДМК Пресс, 2016. – 316 с.: ил.
- [2] [https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA))
- [3] https://ru.wikipedia.org/wiki/C_Sharp
- [4] https://en.wikipedia.org/wiki/Oculus_Quest_2
- [5] <https://metanit.com/sharp/>
- [6] <https://docs.microsoft.com/ru-ru/aspnet/core/mvc/overview?view=aspnetcore-5.0>
- [7] Getting your game on: Using virtual reality to improve real table tennis skills / Stefan Carlo Michalski, Ancret Szpak, Dimitrios Saredakis, Tyler James Ross, Mark Billingham, Tobias Loetscher / September 10, 2019
- [8] A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda / Jaziar Radiantia, Tim A.Majchrzaka, Jennifer Fromm, Isabell Wohlgenannt / April 2020
- [9] Using Virtual Environments to Improve Real-World Motor Skills in Sports: A Systematic Review / Stefan C. Michalski, Ancret Szpak, Tobias Loetscher / September 20, 2019

A ҚОСЫМШАСЫ

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.AI;

public class MainController : MonoBehaviour
{
    public GameObject prefabGoblin;
    private GameObject newEnemy;

    public int roundCount;
    private int round;
    public int enemyCount;
    public Transform SpawnAreaX;
    public Transform SpawnAreaZ;
    private int xPos;
    private int zPos;
    public float waitSec;

    private int killedEnemyCount;
    [SerializeField] private TextMeshProUGUI roundCountText;
    [SerializeField] private TextMeshProUGUI HPText;
    [SerializeField] private Canvas MainCanvas;

    [SerializeField] private TextMeshProUGUI winText;
    [SerializeField] private TextMeshProUGUI loseText;

    public int maxHP;

    void Start()
    {
        StartCoroutine(EnemySpawn());
        killedEnemyCount = 0;
        round = 1;
    }
    void Update()
    {
    }

    IEnumerator EnemySpawn()
    {
```

```

    for (int i = 0; i < enemyCount; i++)
    {
        xPos = (int)Random.Range(SpawnAreaX.position.x,
SpawnAreaZ.position.x);
        zPos = (int)Random.Range(SpawnAreaX.position.z,
SpawnAreaZ.position.z);
        newEnemy = Instantiate(prefabGoblin, new Vector3(xPos,
SpawnAreaX.position.y, zPos), Quaternion.identity);
        newEnemy.SetActive(true);
        yield return new WaitForSeconds(waitSec);
    }
}

```

```

public void KillEnemy()
{
    killedEnemyCount += 1;

    if (killedEnemyCount == enemyCount)
    {
        StartCoroutine(CheckKilledEnemy());
    }
}

```

```

public void PassedEnemy()
{
    maxHP -= 1;
    HPText.SetText(maxHP.ToString());
    KillEnemy();
}

```

```

IEnumerator CheckKilledEnemy()
{
    if (maxHP <= 0)
    {
        MainCanvas.gameObject.SetActive(false);
        round = roundCount;
        loseText.gameObject.SetActive(true);
    }
    else if (round != roundCount)
    {
        killedEnemyCount = 0;
        enemyCount += 1;
        round += 1;
        yield return new WaitForSeconds(8);
    }
}

```



```

        StartCoroutine(EnemySpawn());
    }
    else
    {
        MainCanvas.gameObject.SetActive(false);
        winText.gameObject.SetActive(true);
    }
    roundCountText.SetText(round.ToString());
}
}

```

```

using UnityEngine;
using UnityEngine.AI;
using System.Collections;

```

```

public class EnemyController : MonoBehaviour
{

    public GameObject targetArea;
    private float move = 20;
    private bool stop = false;
    private float blend;
    private float delay = 0;
    public float AddRunSpeed = 1;
    public float AddWalkSpeed = 1;
    private bool hasAniComp = false;
    private NavMeshAgent agent;
    private GameObject navMesh;

    private bool dead = false;

    void Start()
    {
        agent = GetComponent<NavMeshAgent>();
        if (null != GetComponent<Animation>())
        {
            hasAniComp = true;
        }
        agent.SetDestination(targetArea.transform.position);
        GetComponent<Animation>().Play("walk");

        navMesh = GameObject.Find("NavMesh");
    }
}

```

```

void MoveActive()
{
    float speed = 0.0f;
    float add = 0.0f;

    if (hasAniComp == true)
    {
        move *= 1.015F;
        GetComponent<Animation>().Play("walk");
        add = 5 * AddWalkSpeed;
    }
    else
    {
        if (Input.GetKey(KeyCode.UpArrow))
        {
            add = 5 * AddWalkSpeed;
            speed = Time.deltaTime * add;
            transform.Translate(0, 0, speed);
        }
    }
}

public void Dead()
{
    StartCoroutine(Wait());
    navMesh.GetComponent<MainController>().KillEnemy();
    agent.Stop();
    GetComponent<Animation>().CrossFade("dead", 0.1f);
}

private IEnumerator Wait()
{
    yield return new WaitForSeconds(2);
    gameObject.SetActive(false);
}

bool CheckAniClip(string clipname)
{
    if (this.GetComponent<Animation>().GetClip(clipname) == null)
        return false;
    else if (this.GetComponent<Animation>().GetClip(clipname) != null)
        return true;

    return false;
}

```

```

}

void Update()
{
    var targetPosition = targetArea.transform.position;
    targetPosition.y = gameObject.transform.position.y;
    if (!dead && Vector3.Distance(targetPosition, transform.position) < 1.0)
    {
        dead = true;
        agent.Stop();
        navMesh.GetComponent<MainController>().PassedEnemy();
    }

    if (hasAniComp == true)
    {

        if (Input.GetKey(KeyCode.V))
        {
            if (CheckAniClip("damage away") == false) return;

            GetComponent<Animation>().CrossFade("damage away", 0.2f);
            GetComponent<Animation>().CrossFadeQueued("idle01");
        }

        if (Input.GetKey(KeyCode.C))
        {
            if (CheckAniClip("dead away") == false) return;

            GetComponent<Animation>().CrossFade("dead away", 0.2f);
        }

        if (Input.GetKey(KeyCode.E))
        {
            if (CheckAniClip("attack03") == false) return;

            GetComponent<Animation>().CrossFade("attack03", 0.2f);
            GetComponent<Animation>().CrossFadeQueued("idle01");
        }

        if (Input.GetKey(KeyCode.Q))
        {
            if (CheckAniClip("attack01") == false) return;

            GetComponent<Animation>().CrossFade("attack01", 0.2f);
        }
    }
}

```

```

    GetComponent<Animation>().CrossFadeQueued("idle01");
}

if (Input.GetKey(KeyCode.W))
{
    if (CheckAniClip("attack02") == false) return;

    GetComponent<Animation>().CrossFade("attack02", 0.2f);
    GetComponent<Animation>().CrossFadeQueued("idle01");
}

if (Input.GetKey(KeyCode.A))
{
    if (CheckAniClip("drop down") == false) return;

    GetComponent<Animation>().CrossFade("drop down", 0.2f);
}

if (Input.GetKey(KeyCode.Z))
{
    if (CheckAniClip("sit up") == false) return;

    GetComponent<Animation>().CrossFade("sit up", 0.2f);
    GetComponent<Animation>().CrossFadeQueued("idle01");
}

if (Input.GetKey(KeyCode.S))
{
    if (CheckAniClip("damage") == false) return;

    GetComponent<Animation>().CrossFade("damage", 0.1f);
    GetComponent<Animation>().CrossFadeQueued("idle01");
}

if (Input.GetKey(KeyCode.X))
{
    if (CheckAniClip("dead") == false) return;

    GetComponent<Animation>().CrossFade("dead", 0.1f);
}

if (Input.GetKey(KeyCode.D))
{
    if (CheckAniClip("idle02") == false) return;
}

```

```

        GetComponent<Animation>().CrossFade("idle02", 0.1f);
        GetComponent<Animation>().CrossFadeQueued("idle01");
    }

}

if (Input.GetKey(KeyCode.LeftArrow))
{
    transform.Rotate(0.0f, Time.deltaTime * -100.0f, 0.0f);
}

if (Input.GetKey(KeyCode.RightArrow))
{
    transform.Rotate(0.0f, Time.deltaTime * 100.0f, 0.0f);
}

}
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;
public class Arrow : XRGrabInteractable
{
    public float speed = 1000f;
    public Transform tip;
    bool inAir = false;
    Vector3 lastPosition = Vector3.zero;
    private Rigidbody rb;
    public Collider sphereCollider;

    [Header("Particles")]
    public ParticleSystem trailParticle;
    public ParticleSystem hitParticle;
    public TrailRenderer trailRenderer;

    [Header("Sound")]
    public AudioClip launchClip;
    public AudioClip hitClip;

    protected override void Awake()

```

```

{
    base.Awake();
    rb = GetComponent<Rigidbody>();

}
private void FixedUpdate()
{
    if (inAir)
    {
        CheckCollision();
        lastPosition = tip.position;
    }
}

private void CheckCollision()
{
    if (Physics.Linecast(lastPosition, tip.position, out RaycastHit hitInfo))
    {
        if (hitInfo.transform.TryGetComponent(out Rigidbody body))
        {
            if (body.TryGetComponent<EnemyController>(out EnemyController
enemyController))
                enemyController.Dead();
            else if (body.TryGetComponent<Lantern>(out Lantern lantern))
                lantern.TurnOn();
            else if (body.TryGetComponent<Potion>(out Potion potion))
                potion.BreakPotion();
            rb.interpolation = RigidbodyInterpolation.None;
            transform.parent = hitInfo.transform;
            body.AddForce(rb.velocity, ForceMode.Impulse);
        }
        StartCoroutine(Dead());
        Stop();
    }
}

private IEnumerator Dead()
{
    yield return new WaitForSeconds(2);
    gameObject.SetActive(false);
}

private void Stop()
{

```

```

    inAir = false;
    SetPhysics(false);

    ArrowParticles(false);
    ArrowSounds(hitClip, 1.5f, 2, .8f, -2);
}

public void Release(float value)
{
    inAir = true;
    SetPhysics(true);
    MaskAndFire(value);
    StartCoroutine(RotateWithVelocity());

    lastPosition = tip.position;

    ArrowParticles(true);
    ArrowSounds(launchClip, 4.2f + (.6f * value), 4.4f + (.6f * value),
    Mathf.Max(.7f, value), -1);
}

private void SetPhysics(bool usePhysics)
{
    rb.useGravity = usePhysics;
    rb.isKinematic = !usePhysics;
}

private void MaskAndFire(float power)
{
    colliders[0].enabled = false;
    interactionLayerMask = 1 << LayerMask.NameToLayer("Ignore");
    Vector3 force = transform.forward * power * speed;
    rb.AddForce(force, ForceMode.Impulse);
}

private IEnumerator RotateWithVelocity()
{
    yield return new WaitForFixedUpdate();
    while (inAir)
    {
        Quaternion newRotation = Quaternion.LookRotation(rb.velocity,
transform.up);
        transform.rotation = newRotation;
        yield return null;
    }
}

```

```

}

public void OnSelectEnter(XRBaseInteractor interactor)
{
    base.OnSelectEnter(interactor);
}

public new void OnSelectExit(XRBaseInteractor interactor)
{
    base.OnSelectExit(interactor);
}

public void ArrowHaptic(XRBaseInteractor interactor)
{
    if (interactor is HandInteractor hand)
    {
        if (hand.TryGetComponent(out XRController controller))
            HapticManager.Impulse(.7f, .05f, controller.inputDevice);
    }
}

void ArrowParticles(bool release)
{
    if (release)
    {
        trailParticle.Play();
        trailRenderer.emitting = true;
    }
    else
    {
        trailParticle.Stop();
        hitParticle.Play();
        trailRenderer.emitting = false;
    }
}

void ArrowSounds(AudioClip clip, float minPitch, float maxPitch, float volume,
int id)
{
    SFXPlayer.Instance.PlaySFX(clip, transform.position, new
SFXPlayer.PlayParameters()
    {
        Pitch = Random.Range(minPitch, maxPitch),
        Volume = volume,

```



```

        SourceID = id
    });
}

}

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

public class WitchHouseUIHook : WatchScript.IUIHook
{
    public GameObject LeftUILineRenderer;
    public GameObject RightUILineRenderer;

    public override void GetHook(WatchScript watch)
    {
        var ind = SceneManager.GetActiveScene().buildIndex;
        watch.AddButton("Reset", () => { SceneManager.LoadScene(ind); });
        if (ind == 0)
        {
            watch.AddButton("Start Game", () => { SceneManager.LoadScene(1); });
        }
        else
        {
            watch.AddButton("Back to menu", () => { SceneManager.LoadScene(0);
        });
    }

    LeftUILineRenderer.SetActive(false);
    RightUILineRenderer.SetActive(false);

    watch.UILineRenderer = LeftUILineRenderer;
}

}

using UnityEngine;
using UnityEngine.Events;
using UnityEngine.XR.Interaction.Toolkit;

public class Bow : XRGrabInteractable
{
    public Transform notch;

```

```

private PullInteraction pullInteraction;
private LineRenderer lineRenderer;

[Header("Sound")]
public AudioClip grabClip;

[Header("Quiver")]
public Transform quiver;
public Vector2 quiverOffset;

protected override void Awake()
{
    base.Awake();
    pullInteraction = GetComponentInChildren<PullInteraction>();
    lineRenderer = GetComponentInChildren<LineRenderer>();
}

public override void
ProcessInteractable(XRInteractionUpdateOrder.UpdatePhase updatePhase)
{
    base.ProcessInteractable(updatePhase);
    if (updatePhase == XRInteractionUpdateOrder.UpdatePhase.Dynamic)
        if (isSelected)
            UpdateBow(pullInteraction.PullAmount);
}
private void UpdateBow(float value)
{
    Vector3 linePosition = Vector3.forward * Mathf.Lerp(-0.25f, -0.5f, value);
    notch.localPosition = linePosition;
    lineRenderer.SetPosition(1, linePosition);
}

protected override void OnSelectEnter(XRBaseInteractor interactor)
{
    base.OnSelectEnter(interactor);
}

public void OffsetQuiver(XRBaseInteractor interactor)
{
    if (interactor.TryGetComponent(out XRController controller))
    {
        bool right = (controller.inputDevice.role ==
UnityEngine.XR.InputDeviceRole.RightHanded);

```

```

        quiver.localPosition = new Vector3(quiverOffset.x * (right ? -1 : 1),
quiverOffset.y, quiver.localPosition.z);
    }
}

using System;
using UnityEngine;

namespace UnityEngine.XR.Interaction.Toolkit
{
    public class TeleportationAnchor : BaseTeleportationInteractable
    {
        [SerializeField]
        [Tooltip("The transform that represents the teleportation destination")]
        Transform m_TeleportAnchorTransform;
        public Transform teleportAnchorTransform { get { return
m_TeleportAnchorTransform; } set { m_TeleportAnchorTransform = value; } }

        private void OnValidate()
        {
            if (teleportAnchorTransform == null)
                teleportAnchorTransform = transform;
        }

        protected override bool GenerateTeleportRequest(XRBaseInteractor
interactor, RaycastHit raycastHit, ref TeleportRequest teleportRequest)
        {
            if(teleportAnchorTransform == null)
            {
                return false;
            }

            teleportRequest.destinationPosition =
m_TeleportAnchorTransform.position;
            teleportRequest.destinationUpVector = m_TeleportAnchorTransform.up;
            teleportRequest.destinationRotation =
m_TeleportAnchorTransform.rotation;
            teleportRequest.destinationForwardVector =
m_TeleportAnchorTransform.forward;
            return true;
        }
        private void OnDrawGizmos()
        {

```

```

        Gizmos.color = Color.blue;

GizmoHelpers.DrawWireCubeOriented(m_TeleportAnchorTransform.position,
m_TeleportAnchorTransform.rotation, 1.0f);

        GizmoHelpers.DrawAxisArrows(m_TeleportAnchorTransform, 1.0f);
    }
}
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

public class PrefabSpawner : XRSocketInteractor
{
    [SerializeField] GameObject prefab = default;
    private Vector3 attachOffset = Vector3.zero;
    public Arrow currentArrow;

    [Header("Sound")]
    public AudioClip grabClip;

    protected override void Awake()
    {
        base.Awake();
        CreateAndSelectPrefab();
        SetAttachOffset();
    }
    protected override void OnSelectExit(XRBaseInteractable interactable)
    {
        base.OnSelectExit(interactable);
        CreateAndSelectPrefab();
    }

    void CreateAndSelectPrefab()
    {
        Arrow interactable = CreatePrefab();
        SelectPrefab(interactable);
    }

    Arrow CreatePrefab()

```

```

    {
        currentArrow = Instantiate(prefab, transform.position - attachOffset,
transform.rotation).GetComponent<Arrow>();
        return currentArrow;
    }
void SelectPrefab(Arrow interactable)
{
    OnSelectEnter(interactable);
    interactable.OnSelectEnter(this);
}

void SetAttachOffset()
{
    if (selectTarget is XRGrabInteractable interactable)
    {
        attachOffset = interactable.attachTransform.localPosition;
    }
}

public void ForceDeinteract(XRBaseInteractable interactable)
{
    OnSelectExit(interactable);
}
}

```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ғылыми жетекшінің пікірі

Дипломдық жұмыс

Жағабаев Аңсар Махмутұлы

5B070300 – Ақпараттық жүйелер

Тақырыбы: Unity қозғалтқышында, Oculus Quest виртуалды шындық көзілдірігіне арналған жоба жасау

Бұл дипломдық жұмыс өзінің логикалық құрылымымен ерекшеленген. Түсіндірме жобаның құрамы кіріспеден, 3 бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Менің пікірімше, диплом жобалаушы алдына қойылған тапсырманы толығымен орындады және кейінгі технологияларын меңгергендігін көрсетті.

Жалпы дипломдық жоба профессионалдық деңгейде орындалған. Түсіндірме жазба сауатты бейнеленген, жоба бойынша барлық қажетті ақпараттар бар. Кемшілік ретінде кейбір шағын стилистикалық қателерді атап кетуге болады.

Жоғарыда айтылғандарға байланысты, «Unity қозғалтқышында, Oculus Quest виртуалды шындық көзілдірігіне арналған жоба жасау» дипломдық жұмысы 5B070300 – «Ақпараттық жүйелер» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және дипломдық жұмыс қорғауға жіберіле алады, ал оның авторы Жағабаев Аңсар Махмутұлы бакалавр академиялық дәрежесін алуға лайықты деп есептеймін.

Ғылыми жетекші

Техн. ғыл. магистрі, лектор



Аристомбаева М.Т.

«29» 05 2021 ж.

Кафедра меңгерушісінің Ұқсастық есепті Талдау хаттамасы

Кафедра меңгерушісі жұмысқа қатысты плагиатты анықтау және алдын алу жүйесімен құрылған ұқсастықтың толық есебімен танысқанын мәлімдейді:

Автор: Жағабаев Аңсар Махмутұлы

Атауы: Unity қозғалтқышында, Oculus Quest виртуалды шындық көзілдірігіне арналған жоба жасау

Үйлестіруші: Аристомбаева Меруерт Турлубекқызы

Ұқсастық коэффициенті 1: 0.48

Ұқсастық коэффициенті 2: 0.00

Дабыл: 1.31

Ұқсастық есебін талдағаннан кейін кафедра меңгерушісі мынаны айтады:

жұмыста табылған қарыздар адал және плагиат белгілері жоқ. Осыған байланысты мен жұмысты өзім мойындаймын және оны қорғауға рұқсат етемін;

жұмыста табылған қарыздарда плагиат белгілері жоқ, бірақ олардың шамадан тыс мөлшері жұмыстың мәніне және оның авторының тәуелсіздігінің болмауына күмән тудырады. Осыған байланысты қарыз алуды шектеу мақсатында жұмыс қайта өңделуі керек;

жұмыста табылған қарыздар жосықсыз болып табылады және плагиат белгілері бар немесе жосықсыз қарыз алуды жасыру әрекеттерін көрсететін мәтінді қасақана бұрмалау бар. Осыған байланысты жұмысты қорғауға жібермеймін.

Негіздеме:

Плагиат туралы есепті және магистранттың жұмысын талдағаннан кейін қарыз алу Адал және плагиат белгілері жоқ екендігі анықталды, сондықтан олар негізінен белгілі терминдерді, сондай-ақ компанияның қайталанатын атауларын қолданумен байланысты.

«28» мамыр 2021 ж.

Күні



Сейлова Н.А., кафедра
меңгерушісінің қолы

Ғылыми жетекшінің Ұқсастық есепті Талдау хаттамасы

Мен жұмысқа қатысты плагиатты анықтау және алдын-алу жүйесі жасаған ұқсастықтың толық есебімен танысқанымды мәлімдеймін:

Автор: Жағабаев Аңсар Махмутұлы

Атауы: Unity қозғалтқышында, Oculus Quest виртуалды шындық көзілдірігіне арналған жоба жасау

Үйлестіруші: Аристомбаева Меруерт Турлубекқызы

Ұқсастық коэффициенті 1: 0.48

Ұқсастық коэффициенті 2: 0.00

Дабыл: 1.31

Ұқсастық есебін талдағаннан кейін мен мынаны айтамын:

жұмыста табылған қарыздар адал және плагиат белгілері жоқ. Осыған байланысты мен жұмысты өзім мойындаймын және оны қорғауға рұқсат етемін;

жұмыста табылған қарыздарда плагиат белгілері жоқ, бірақ олардың шамадан тыс мөлшері жұмыстың мәніне және оның авторының тәуелсіздігінің болмауына күмән тудырады. Осыған байланысты қарыз алуды шектеу мақсатында жұмыс қайта өңделуі керек;

жұмыста табылған қарыздар жосықсыз болып табылады және плагиат белгілері бар немесе жосықсыз қарыз алуды жасыру әрекеттерін көрсететін мәтінді қасақана бұрмалау бар. Осыған байланысты жұмысты қорғауға жібермеймін.

Негіздеме:

Жұмыстағы қарыз алу Адал және плагиат белгілері жоқ, жалпы қабылданған техникалық терминологияны қолданумен, сондай-ақ компания атауын қолданумен түсіндіріледі.

«28» мамыр 2021 ж.

Күні



Аристомбаева М.Т.
Ғылыми жетекші қолы