

**ЎЗБЕКИСТОН АЛОҚА, АХБОРОТЛАШТИРИШ ВА  
ТЕЛЕКОММУНИКАЦИЯ ТЕХНОЛОГИЯЛАРИ ДАВЛАТ  
ҚЎМИТАСИ  
ТОШКЕНТАХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**

*Қўлёзма ҳуқуқида*

УДК

**ЖОЛДАСОВА КАМИЛА ЖОЛЛИБАЕВНА**

**Виртуал ва аддитив воқелик тизимларининг таснифи ва уларни  
қуриш усулларининг таҳлили**

**5A330202- “Ахборот ва мультимедиа технологиялари”  
(тармоқлар бўйича)**

**Магистр  
академик даражасини олиш учун ёзилган  
диссертация**

**Илмий раҳбар:  
ф-м.ф.н. Қабилжанова Ф.А.**

| <b>МУНДАРИЖА</b> |                                                                         |           |
|------------------|-------------------------------------------------------------------------|-----------|
|                  | <b>КИРИШ.....</b>                                                       | <b>3</b>  |
| <b>I БОБ.</b>    | <b>ВИРТУАЛ ВА АДДИТИВ ВОҚЕЛИК РИВОЖИ.....</b>                           | <b>8</b>  |
| 1.               | Виртуал ва аддитив воқелик тарихи .....                                 | 8         |
| 2.               | Олдинги виртуал ва аддитив воқелик технологиялари.....                  | 13        |
| 3.               | Виртуал ва аддитив воқелик тизимининг афзалликлари ва камчиликлари..... | 17        |
|                  | <b>I боб буйича хулоса.....</b>                                         | <b>21</b> |
| <b>II БОБ.</b>   | <b>ВИРТУАЛ ВА АДДИТИВ ВОҚЕЛИКНИ ЯРАТИШ.....</b>                         | <b>22</b> |
| 1.               | Виртуал воқеликни моделлаштириш тили (VRML).....                        | 22        |
| 2.               | O3D – виртуал борликни яратиш платформаси.....                          | 30        |
| 3.               | Дастурий таъминотни ишлаб чиқиш учун фреймворкни танлаш.....            | 38        |
|                  | <b>II боб буйича хулоса.....</b>                                        | <b>45</b> |
| <b>III БОБ.</b>  | <b>ДАСТУРИЙ ТАЪМИНОТНИНГ ТАСВИРЛАНИШИ.....</b>                          | <b>46</b> |
| 1.               | Виртуал объектлар билан ўзаро ҳаракатлар.....                           | 46        |
| 2.               | Дастурий таъминотнинг архитектураси.....                                | 53        |
| 3.               | Фойдаланувчига йўриқнома .....                                          | 55        |
|                  | <b>III боб буйича хулоса.....</b>                                       | <b>61</b> |
|                  | <b>ХУЛОСА.....</b>                                                      | <b>62</b> |
|                  | <b>Фойдаланилган адабиётлар .....</b>                                   | <b>64</b> |
|                  | <b>ИЛОВАЛАР.....</b>                                                    | <b>69</b> |

## КИРИШ

**Мавзунинг долзарблиги.** Ахборот–коммуникация тизимларида маълумот алмашинувини самарали амалга оширишни ташкил этиш бугунги ривожланган жамиятимизда катта аҳамият касб этади. Ахборот технологияларининг жадал ривожланиб бориши, замонавий алоқа тармоқларининг ривожланишига олиб келади.

Ўзбекистон Республикаси Президенти И.А. Каримов ва Ўзбекистон Республикаси Вазирлар Маҳкамасининг қатор фармон ва қарорларида Республикада ахборот технологияларини ривожлантиришнинг аниқ йўналишлари белгилаб берилиб, соҳа мутахассисларига фаоллик кўрсатиш учун шарт-шароитлар яратилиб берилмоқда[1-6]. Бундай фармон ва қарорларга қуйидагиларни киритиш мумкин:

–“Компьютерлаштиришни янада ривожлантириш ва ахборот – коммуникация технологияларини жорий этиш тўғрисида”Ўзбекистон Республикаси Президентининг фармони (30 май 2002 й.);

–Ўзбекистон Республикасининг “Ахборотлаштириш тўғрисида”ги қонуни (11 декабр 2003 й.);

–Ўзбекистон Республикасининг “Алоқа, ахборотлаштириш ва телекоммуникация технологиялари давлат қўмитасини ташкил этиш тўғрисида” фармони (16 октябр 2012й.);

–“Замонавий ахборот–коммуникация технологияларини янада жорий этиш ва ривожлантириш чора-тадбирлари тўғрисида”Ўзбекистон Республикаси Президентининг қарори (21 март 2012 й.);

–Ўзбекистон Республикасининг“Миллий ахборот – коммуникация тизимини янада ривожлантириш чора – тадбирлари тўғрисида” ги қонуни (27 июн 2013й.);

–Ўзбекистон Республикасининг “Ахборот – коммуникация технологиялари соҳасида кадрлар тайёрлаш тизимини янада

такомиллаштириш чора – тадбирлари тўғрисида”ги қонуни (26 март 2013й.) ва бошқалар.

Маълумотларга параллел ишлов бериш жараёнида маълумот хажмининг ошиб бориши янада кўпроқ маълумотларни таҳлил қилишга ва қайта ишлашга олиб келади. Буни бажаришда бизга объект ҳаракатларининг айрим тавсифномаларини бошқа объект, масалан, компьютер ёрдамида акс эттириш технологияси ёрдам беради. Бу технология компьютерли моделлаштириш дейилади ва визуаллаштириш уни амалга оширишни таъминлайди.

Визуаллаштириш – бу маълумотларнинг ментал моделини ташкил қилиш бўлиб, унинг ёрдамида маълумотларнинг ички структураси ҳақида тасаввур қилиш таъминланади. Визуаллаштириш ёрдамида объектнинг абстракт моделини компьютерда яратиш виртуал воқелик ёрдамида бажарилади.

Виртуал воқелик – бу виртуал, яъни ҳақиқатан мавжуд бўлмаган ёки бўлиб, ундан бошқача қабул қилинадиган муҳит. “Виртуал воқелик” тушунчаси компьютер воситалари ёрдамида яратилган дунёни билдиради. У ҳақиқатан мавжуд бўлмайди, бироқ компьютер инсоннинг кўриш, эшитиш ва бошқа ҳиссиёт органларига таъсир қилиб, ушбу дунёдан фойдаланиш иллюзиясини келтириб чиқаради. Бундан ташқари, инсон ушбу дунёда рўй бераётган воқеаларга ўз таъсирини ўтказиши мумкинлиги ҳиссини кучайтиради.

Виртуал воқеликдан фойдаланишнинг оддий мисоли бу – компьютер ўйинидир. Фойдаланувчи атрофида компьютерлар томонидан яратилган воқелик ҳиссини берувчи тасаввур ва товушлар пайдо бўлади. Фойдаланувчи сунъий дунё билан унинг ҳаракатлари ва туйғуларини ҳамда аудиовизуал эффектларини боғловчи шлем ва қўлқоп каби турли сенсорлар орқали мулоқотда бўлади. Виртуал воқелик соҳасидаги келгуси

тадқиқотлар кузатилаётган нарсаларнинг ҳаққонийлиги туйғусини кучайтиришга қаратилган.

Виртуал воқелик технологияси – бу ахборот технологияларининг тури. Информатика ва кибернетика воситалари ёрдамида инсонлар онгида виртуал воқеликни шакллантиришни таъминлайди. Виртуал воқелик технологияси доимий равишда ривожланиб бормоқда. Бу технологиянинг энг дастлабкиси овозсиз кино бўлса, охиригилари – стерео, видео ва виртуал воқелик шлемларидир [1].

**Тадқиқот объекти ва предмети.** Виртуал ва аддитив воқелик тизимлари – тадқиқот объекти ҳисобланади. Виртуал ва аддитив воқелик тизимларини қуриш усуллари таҳлили ва уларнинг таснифи – тадқиқот предмети ҳисобланади.

**Магистрлик диссертациясининг мақсади.** Виртуал ва аддитив воқелик тизимларини қуришнинг мавжуд усулларини тадқиқ қилиш ва виртуал ва аддитив воқеликни моделлаштириш учун дастурий таъминот ишлаб чиқиш.

**Тадқиқот масалалари:**

- виртуал ва аддитив воқелик тизимларини ўрганиш;
- виртуал ва аддитив воқелик тизимларини қуриш усулларини тадқиқ қилиш;
- виртуал ва аддитив воқелик тизимларини қиёсий таҳлил қилиш;
- виртуал ва аддитив воқелик тизимлари таснифини тадқиқ қилиш;
- виртуал ва аддитив воқеликни моделлаштириш учун дастурий таъминот ишлаб чиқиш.

**Мавзунинг ўрганилганлик даражаси.** Виртуал ва аддитив воқеликни ўрганишга бир қатор олимларнинг ишлари бағишланган. Жумладан, В. Л. Авербух, А. Ю. Байдалин, П. А. Васёв, Д. Р. Исмагилов, А. Ю. Казанцев, Д. В. Манаков, А. Н. Шинкевич ва бошқалар. Республикамизда бу масалалар билан олимлардан академиклар Т. Ф

Бекмуратов, М. М. Камилов М. М, шунингдек, профессор Мусаев М. М. ва уларнинг шогирдлари шуғулланишади [7-10].

**Тадқиқот усуллари.** Виртуал ва аддитив воқеликни моделлаштиришнинг эмпирик ва математик усуллари, уни қуриш алгоритмлари ва методлари.

**Ишнинг амалий аҳамияти.** Виртуал ва аддитив воқелик тизимларини ўрганишга бағишланган махсус курсларда фойдаланиш мумкин.

**Илмий янгилиги.** Виртуал ва аддитив воқелик тизимлари қиёсий таҳлил қилинган, уни моделлаштириш учун дастурий таъминот ишлаб чиқилган.

**Ишнинг апробацияси.** Асосий натижалари қуйидаги анжуманларда муҳокама қилинган:

- Joldasova K.J., Alisherov F.A., Kabiljanova F.A.—“Study of user experience in virtual environments”, “XXI аср – интеллектуал авлод асри” Тошкент шаҳри ва вилояти ҳудудий илмий – амалий конференцияси. 16 – 17 май 2014й. 198-2016.

- Кабилжанова Ф.А., Жолдасова К.Ж. “Возможности применения виртуальной реальности”.— Техник ва ижтимоий – иқтисодий фанлар соҳаларининг муҳим масалалари. Республика Олий ўқув юртлараро илмий ишлар тўплами. Ўқисм. Тошкент, 2014й. 82-84б.

- Кабилжанова Ф.А., Жолдасова К.Ж. “Сравнительный анализ сред виртуальной реальности”.— Техник ва ижтимоий – иқтисодий фанлар соҳаларининг муҳим масалалари. Республика Олий ўқув юртлараро илмий ишлар тўплами. Ўқисм. Тошкент, 2014й. 84-86б.

**Натижалари.** Диссертация мавзуси бўйича 3та мақола нашр қилинган.

**Ишнинг ҳажми ва структураси.** Магистрлик диссертацияси кириш, учта боб, хулоса, фойдаланилган адабиётлар рўйхати ва иловадан иборат.

Биринчи бобда виртуал ва аддитив воқеликнинг тарихи, илғор технологиялари, афзалликлари ва камчиликлари, ривожланиши ҳақида сўз юритилади.

Иккинчи бобда виртуал ва аддитив воқелик тизимларининг таснифи, тизимлар таҳлили келтирилган.

Учунчи бобда виртуал ва аддитив воқеликни моделлаштириш тили, уни яратиш ва яратилган виртуал воқеликнинг таҳлили келтирилган.

Хулоса қисмида магистрлик диссертацияси бўйича қилинган асосий хулосалар берилган.

# **I БОБ. ВИРТУАЛ ВА АДДИТИВ ВОКЕЛИКНИНГ РИВОЖЛАНИШИ**

## **1. -Виртуал ва аддитив воқеликнинг тарихи**

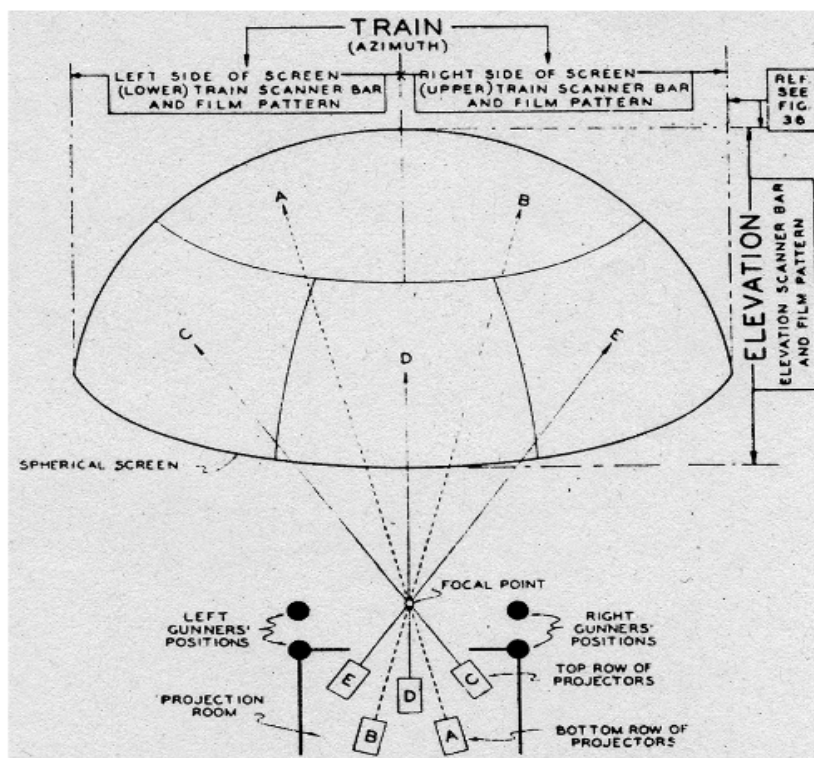
Виртуал воқеликнинг тарихи дастлаб Link Trainer имитацион тренажердан бошланган. Ушбу тренажер 1929 йил Эдвин Линк томонидан ҳарбий бошқарма учун яратилган бу виртуал воқелик тизимининг биринчи прототипи бўлиб ҳисобланади. Тренажер ўқувчиларни ўқитиш вақтида учиш жараёнини имитация қилувчи восита бўлган. Линк тренажери ҳаво пуфловчи орган асбобининг қисми ва кабинали фюзеляждан иборат бўлган. Ҳаво ҳаракати ёрдамида кабина худди реал учиш вақтидагидек ҳаракат қилган. Учиш тренажери уч ўлчовли фазодаги ҳаракатни синхрон равишда кўрсатган [7].

1950 йил бошларида Линк тренажерлар яратишда компьютер технологияларидан фойдалана бошлади. Шундан сўнг авиацион тренажерларининг янги авлоди яратилди. Линк тренажерларидан Шаттл учувчиларини ўргатишда Аполло астронавтларини машқ қилдиришда фойдаланилди.

1939 йил Нью-Йоркдаги бутун жаҳон кўргазмасида Витарама тизими, замонавий панорама кинотеатрларининг кўриниши намойиш қилинди. Архитектор Ральф Уолкер ва инженер Фред Уоллер 11та проектор билан панорамали экранда тасвирни ҳосил қилувчи системани яратишди. Ҳар бир проектор тасвирнинг фақат бир қисmini намойиш қилар эди. Аммо проекторлардан тушадиган тасвирни, экранда махсус жойлаштириш натижасида ягона тасвир ҳосил қилинади. Панорамали экран томошабин атрофидаги барча кўринадиган соҳани тасвир билан тўлдирилган қилиб лойиҳалаштирилганди. Шундай қилиб, фильм намойиш қилинганда томошабин ўзини фильм ичидагидек ҳис қиларди. Иккинчи жаҳон уруши вақтида бу тизим асосида Фред Уоллер Waller



gunnery trainer (Уоллер ўқ отиш тренажери) деб номланган машқ симуляторини яратди. Бу тренажер виртуал воқеликнинг дастлабки кўриниши ҳисобланади.

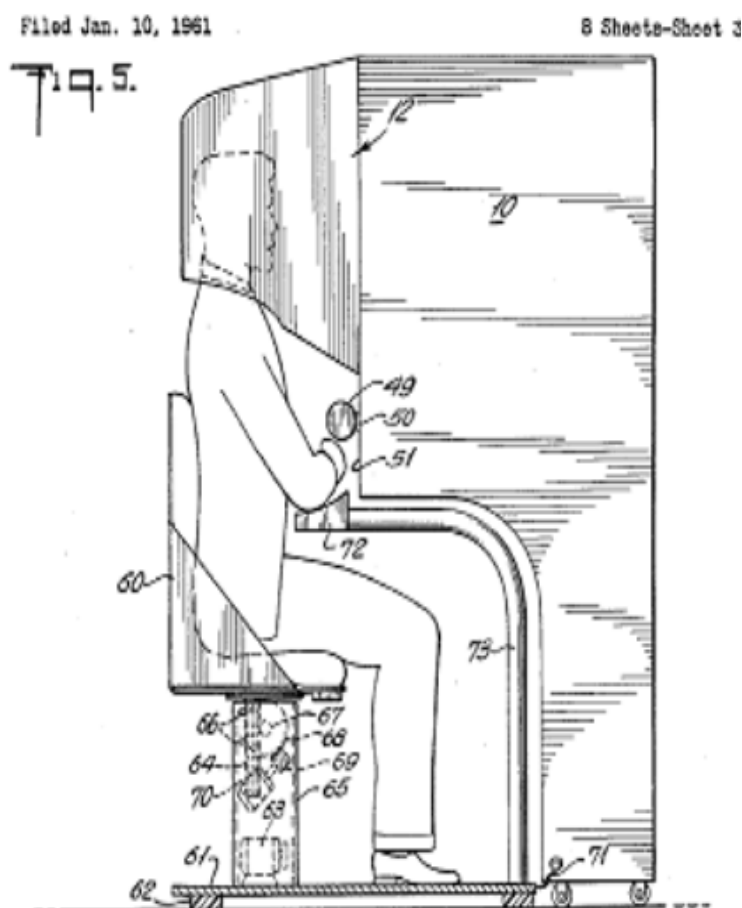


Расм-1. Vitarama тизими кўриниши.

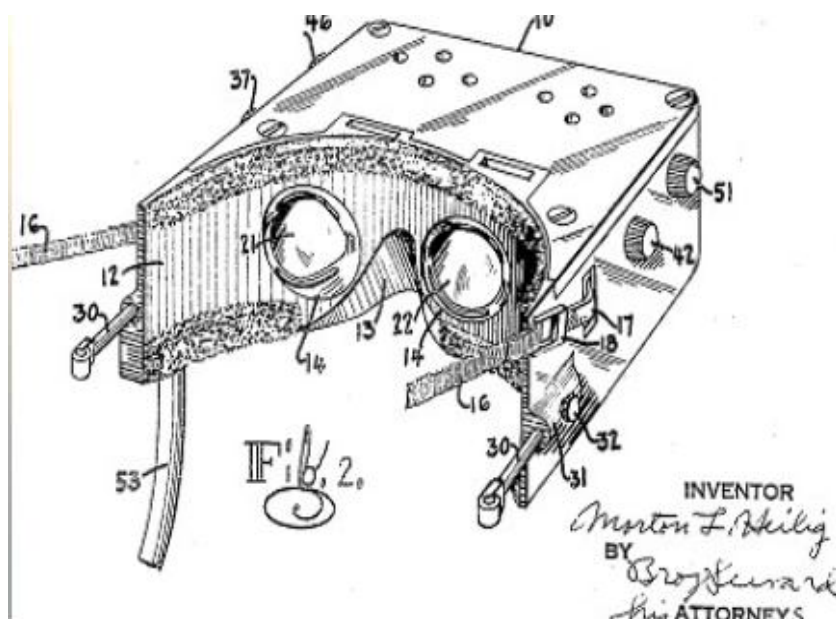
1955 йилда ихтирочи ва кино арбоби Мортон Хейлинг “Театр тажрибаси” китобида, кино жараёнига нафақат кўриш ва эшитиш, балки бошқа сезги аъзоларини ҳам жалб этишга таклиф қилди. 1957 йилда Сенсорама тажриба театри яратилди. Бунда стереотовушли 3D фильм хид, ўтирадиган жойнинг вибрацияси ва шамол шиддати билан намоиш қилинди. Томошабин махсус ўтирадиган жойда бошини махсус камерага тикади ва унга стереовидео роликлар намоиш қилинади. Масалан, у кўча бўйлаб мотоциклда саёҳат қилиши мумкин. Бунда унга бир вақтда 3та камера билан олинган уч ўлчовли фильм намоиш қилинади. Тасвир билан бир вақтда стереотовуш эшиттирилади. Ўтирадиган жойнинг вибрацияси йўлнинг ўнқир-чўнқирларини сездиради ва махсус вентиляторлар ёрдамида шамол сезилади. Фильм сюжетиға мос равишда махсус хид

бериш системалари турли хидларни таратиб туради. Ушбу қурилма виртуал воқеликка тўлиқ кириб кетиш имконини берувчи биринчи қурилма бўлди [8].

Кейинчалик Хейлинг бу ғояни залдаги томошабинлар иштирок этадиган қилиб ривожлантирди. Сенсорамани яратиш вақтида Хейлингда яна бир ғоя туғилди— бу ғоя бошга кийиладиган дисплей эди. Шу тариқа 1961 йилда виртуал воқелик шлемининг биринчи кўриниши —Telesphere Mask—бошга кийиладиган дисплей пайдо бўлди. Бу қурилма стерео эффектни таъминловчи иккита кенг экранли дисплей, стереотовуш берувчи икита динамикдан иборат шлемдир.



Расм-2. Sensorama тажриба театри кўриниши



Расм-3. Telesphere Mask бошга кийиладиган дисплей

Бу барча қурилмалар механик мосламалар ёрдамида ишга туширилиб, фойдаланувчи фақат кузатувчи бўлиб иштирок этади. Компьютерларнинг пайдо бўлиши сунъий фазовий дунёни яратиш имконини берди. 1959 йилда биринчи DAC-1 (Design Augmented by Computers), номли General Motors ва IBM компаниялари тамонидан яратилган график тизими пайдо бўлди. Бу тизим ёрдамида автомобиль кузовининг уч ўлчовли моделини ҳар томонлама кўриш мумкин бўлди.

1963 йилда Айвен Сазерленд график интерфейсининг биринчи кўриниши—Sketchpad дастурини ишлаб чиқди. Мохиятига кўра бу дастур (САПР) - автоматлаштирилган ва лойҳалаштирилган биринчи кўринишидир. Дастур ёрдамида чизиклар чизиб, уларни турли фигуралар кўринишида бирлаштириш мумкин. Ҳосил қилинган фигураларни ягона ҳолда кўчириш, нусхасини олиш, айлантириш ва масштаблаштириш мумкин. Бундай ёндашув объектга йўналтирилган дастурлашга асос бўлди.

Визуаллаштиришдаги ютуқлар ва инсон — машина интерфейси соҳасидаги тараққиёт компьютер технологиялари ёрдамида сунъий воқеликни яратиш имконини берувчи виртуал воқелик тизимларининг пайдо бўлишига олиб келди. 1965 йилда Айвен Сазерленд The Ultimate

Display номли эссе ёзди. Унда фойдаланувчига экранда берилган объектларнинг шакли ва ҳолатини худди реал ҳолатдагидек ўзгартириб бошқаришга имкон берувчи компьютеризимини таърифлаб берди.

Сазерленд концепцияси қўйдагилардан иборат:

- Шлем-дисплей орқали ихтиёрий кузатувчига ҳақиқий бўлиб кўринадиган, уч ўлчовли товуш ва тактли қўзғатувчилар билан тўлдирилган виртуал дунё.
- Дунёни реал вақтда акслантирувчи ва яратувчи компьютер.
- Фойдаланувчилар учун виртуал объектларни, худди ҳақиқий ҳолдагидек бошқариш имконияти.

Сазерленд ўша вақтда унинг концепциясини амалга оширувчи технология йўқлигини тушунарди ва у уларнинг баъзиларини амалга оширувчи содда қурилма яратишга ҳаракат қилди. 1966 йилда мудофаа вазирлигининг қўллаб – қувватлаши билан, у биринчи виртуал воқелик шлемини ишлаб чиқишни бошлади. 1968 йилда эса Айвен Сазерленд бошга кийиладиган дисплейни яратди. Унда тасвир компьютер ёрдамида ҳосил қилинадиган эди. Бу қурилма оғир бўлиб у шифтга осиб қўйиларди. Шу сабабли уни “Дамоклов меч” (The Sword of Damocles) деб аташган. Компьютер тасвирни шакллантириб экранда “стерео” режимида акслантирилган. Тасвир билан бир вақтда стерео товуш ҳам эшиттирилган. Шунингдек, шлемда фойдаланувчи бошининг ҳолатини кузатувчи қурилмалар ўрнатилган бўлиб, улар ёрдамида фойдаланувчи бошининг бурилишига мос равишда дисплей экранидаги тасвир ўзгарган. Бундай шлемнинг биринчи кўриниши фойдаланувчи олдида осиб қўйилган куб бўлган. Шундай қилиб, фойдаланувчини сунъий дунёга киритиш имкони пайдо бўлди. Бунда у четдан кузатувчи эмас, балки, узи ҳам жараёнда фаол иштирок этган. Бу кашфиёт виртуал воқеликни ривожлантиришга катта туртки бўлган [9].

Виртуал воқелик тизимининг энг биринчиси деб Aspen MovieMap (Аспен шахрининг кино картаси) ҳисобланган. У 1977 йилда Массачусет технология Институтда яратилган бўлиб, Аспен шаҳрида автомобил ҳаракатини кўрсатган. Фойдаланувчи шаҳар кўчаларини автомобиль ҳайдовчиси бўлиб кўради. Бунинг учун автомобилда ўрнатилган 4та камера ёрдамида шаҳар кўчалари тасвирга олинган. Камералар автомобиль атрофини тўла камраб оладиган қилиб ўрнатилган. Суратга олиш ҳар ўн фут масофада бажарилган. Олинган тасвирлар асосида фильм монтаж қилинган ва лазерли дискка ёзилган. Лазерли дисклардан фойдаланиш фильмнинг ихтиёрий кадрига эркин ўтиш имконини берди. Шундай қилиб, фойдаланувчи шаҳар бўйлаб ихтиёрий йўналишни танлаши мумкин бўлди. Бунда ягона чегаралаш кўчанинг ўртасида туриб, 10 фут масофага юриш мумкинлиги эди.

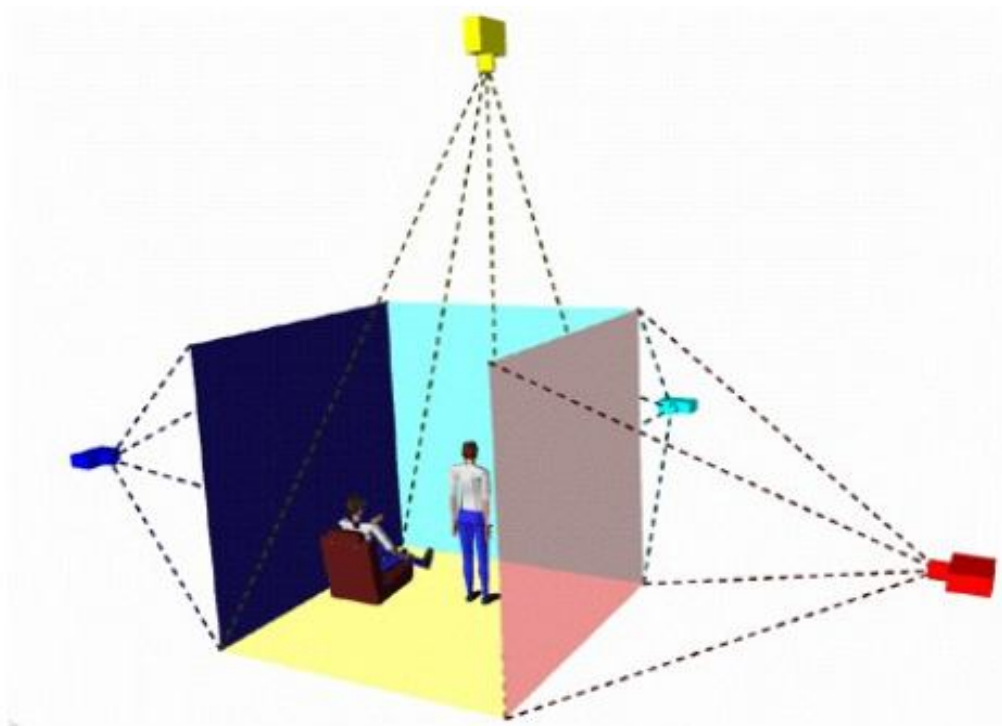
Аввалги Сенсорама системасидан фарқли равишда, фойдаланувчи нафақат виртуал дунёнинг кузатувчиси, балки унинг иштирокчисига айланди. У энди шаҳар кўчаларида бемалол ҳаракат қилиши, ёзги, қишки жой тасвирларини танлаши мумкин бўлди. Aspen Movie Map ҳозирги оммабоп Google Map's street view хизматининг дастлабки кўриниши дейиш мумкин. Интерактивлик ғоясини америкалик рассом Мирон Крюгер ривожлантирди. 1983 йилда у “Видеостол”, улардан кейин “Видеомакон” ғоясини таклиф қилди. “Видеостол” қўл, бармоқлар ва бошқа объектлар ҳаракатини кузатувчи системадан иборат бўлган. Ундан ташқари у жуда кўп хилдаги ҳаракатлардан таъсирланган. “Видеомакон” эса компьютердаги тасвир билан объект (Одам) тасвирини бирлаштирувчи системадан иборат бўлган. Шундай қилиб, сунъий дунё билан одамнинг ўзаро таъсири янги поғонага кўтарилди.

1984 йилдан бошлаб VPL (Visual Programming Language) Research, Inc. компанияси виртуал воқелик қурилмаларини ишлаб чиқариш билан шуғулланди. Бу компания ходимлари томонидан виртуал воқеликнинг

биринчи примитив системаси лойихалаштирилди. Бу виртуал воқелик шлеми VPL HMD ва қўлқоплари VPL DataGlovedан иборат эди. Шунингдек, компания бренди остида биринчи виртуал воқелик костюми VPL DataSuitчиқарилди. VPL DataSuit ёрдамида тана аъзоларининг ҳаракатларини рақамли кадрларда ифодалаш мумкин бўлди. Тананинг фазодаги ҳолати эса 4та қўшимча Polhemus позициондатчиклар ёрдамида ўлчанади.

## **2. Олдинги виртуал ва аддитив воқийликтехнологиялари**

Ҳозирги кунда дунёда виртуал воқеликнинг 500дан ортиқ катта маштабли қурилмалари мавжуд. Улар фанва техниканинг турли соҳаларида виртуал таълимдан тортиб то лойихалашгача ишлатилади. Фойдаланувчини виртуал муҳитда тўлақамраб оладиганвиртуал воқелик системалари эътиборга лойиқдир. Фойдаланувчининг виртуал муҳитга тўлиқ кириши кўпгина техник факторларга боғлиқ. Бу технология одамнинг сезги органлари виртуал муҳитни тўлиқ қабул қилишга мажбур қилиши керак.Шунинг учун тўлиқ қамраб олувчи система ҳажмли тасвирни шакллантиришга имкон берувчи ва фойдаланувчи кўра оладиган соҳани эгалловчи панорамали стереоскоп экран, стерео товушни эшиттирувчи акустик системани, сезиш хиссини қайд этувчи қурилмаларниўз ичига олади. Ундан ташқари фойдаланувчи ташқи муҳит билан табиий, интуитив усулда мулоқот қила олиш имконига эга бўлиш керак. Виртуал муҳит фойдаланувчи ҳаракатларига таъсир кўрсата олиши керак. Бунинг учун трекинг тизимидан фойдаланилади.



Расм-4.CAVE тизими кўриниши

CAVE тизими фойдаланувчи атрофида жойлашган 4та экрандан иборат хона кўринишида(Расм-4).3та ён томондаги экранларда тасвирни шакллантириш, тескари проекция усули билан амалга оширилади. Проекторлар тизимдан ташқарида жойлаштирилади ва экраннинг орқасига тасвирни туширади. Полдаги экранда тасвир юқорида жойлашган проектор ёрдамида тўғридан – тўғри туширилади. Барча экранлар бир-бири билан  $90^0$ ли бурчак остида туташади, шу сабабли, экранлар туташган жойда тасвирни текислаш қийинчилик туғдиради. CAVE тизими тўла виртуал воқелик хиссини яратади. Трекинг қурилмаси “виртуал муҳит”даги одамнинг ҳаракатларини, виртуал дунёдаги тасвирга мос равишда ўзгартириш учун, фиксирлаб беради. Шундай қилиб, тадқиқотчи объектни, унинг атрофида ёки унинг виртуал образи ичида ҳаракатланиб ўрганиши мумкин.

StarCAVE тизими CAVE типли виртуал муҳит тизимларининг учинчиавлодли ҳисобланади. У Калифорния телекоммуникациялар ва ахборот технологиялари институтида қурилган (Calit2).

StarCAVE ҳар томонида учтадан экрани бор бешбурчакли кўринишидаги хонадан иборат. Юқори ва қуйи экранлар хона ичига  $15^0$  оғдирилган. Шундай қилиб, тизим 17та проектли экрандан фойдаланади: 15та экран хона деворларида, биттаси полда ва биттаси шифтда. Хонанинг бундай шакли экранлар туташган жойидаги тўғри бурчакларни йўқотишга ва ягона тасвирни шакллантиришга ёрдам беради. Стерео эффектга эришиш учун ҳар бир экрандаги тасвирни иккита проектор шакллантиради. Натижада 68млн.пиксель сиғимга эга ягона тасвирни ҳосил қилувчи 34 та проектордан фойдаланилади [10].

iCone тизими юқорида келтирилган тизимларга нисбатан анча қулай ҳисобланади. Бу тизимда ишлатиладиган коник экран тасвирда тўғри бурчакларнинг бўлмаслигини таъминлайди. Шу сабабли тасвирда бир текисликдан иккинчисига кескин ўтишлар бўлмайди. i-CONE – бу тўғри чоксиз проекция принципида ишловчи проектли тизимдир.

Тизим экрани тўнқарилган кесик конус шаклига эга, бу унинг номида ҳам ўз аксини топган i-CONE (инглизча, cone — конус). Цилиндрли экрандан фарқли, конусли экраннинг вертикал оғиши  $7^0$ – $10^0$  ни ташкил қилади. Экраннинг шундай шакли туфайли, юқорида жойлашган проектор, унга тўғри бурчакка яқин бурчак остида нур туширади ва экрандан аксланувчи ёруғлик кузатувчи томонга йўналтирилмайди. Конусли экраннинг баландлиги 2.8 м, юқори ва қуйи қирраларининг эгилиш радиуслари мос равишда 2.96 м. ва 3.3 м. i-CONE нинг бошқа тавсифлари CAVE типидagi тизимларга ўхшаш бўлади.

i-CONE тизимининг асосий камчилиги вертикалига кўриш майдонининг чегараланганлигидир. Бу муаммони ҳал қилишда VisionStation ва iDome (инглизча dome—гумбаз) экран тизимлари ишлатилган. iDome тизимининг экрани диаметри 3 метр бўлган кесилган ярим сфера шаклида қилинган. Бундай иншоот горизонталига  $180^0$  ва вертикалига  $135^0$  кўриш майдонига эга бўлиб, инсоннинг кўриш майдонини



тўлик қоплайди. Сферали сиртга тасвирни лойиҳалаш, оддий тизимлардагидек, тўғридан-тўғри эмас, балки эгри ойналар тизими ёрдамида бажарилади. Шу сабабли, тизим билан ишлашда проекторларни созлаш асосий қийинчиликни ташкил қилади. 1993 йилда Фраунгофер институтида Вольфганг Крюгер томонидан Responsive Workbench тизими яратилди.

Responsive Workbench ўзаро тўғри бурчак остида жойлашган иккита экрандан иборат. Проекторлар ва катта ойналардан иборат тизим экранларда тасвирни ҳосил қилади. Ойналардан фойдаланиш проектор ва экранни компакт жойлаштириш имконини беради. Фойдаланувчининг жойлашиши Polhemus фирмасининг электромагнит қурилмаси ёрдамида аниқланади. Бу тизим “иш столи”ни эслатади. Виртуалобъектларнинг проекцияси унга туширилади. Стереопроектор кузатувчига объектларни ҳажмли қилиб кўриш имконини беради, электромагнит трекинг тизими эса, улар билан ўзаро ҳаракат қилишга имкон беради. Responsive Workbench тизими архитектор, дизайнер, хирург ва бошқаларга иш жойи сифатида тўғри келади.

Ҳозирги вақтда Фраунгофер медиакоммуникацияларинститути олимлари L-Shape тизимини яратишди. Ўзининг тузилишига кўра у катталаштирилган Responsive Workbench ни эслатади ва шу тизимнинг такомиллаштирилган авлоди ҳисобланади. Виртуал воқеликнинг барча юқорида келтирилган тизимлари фақат битта фойдаланувчи билан ишлайди. L-Shape нинг улардан фарқи, у коллектив билан ишлашга имкон беради. Бир вақтнинг ўзида иккита фойдаланувчи виртуал дунё объектлари билан ҳаракатланиши мумкин. Бунинг учун тизим экранига битта эмас, балки иккита стереоскопик тасвир туширилади. Бундан ташқари ҳар хил стереоскопик тасвирларни тушириш мумкин.

### **3. Виртуал ва аддитив воқеликнинг афзалликлари ва камчиликлари**

Виртуал воқелик тизимлари ахборотни икки ўлчовли тасвирлашдан кейинги визуализация тизимлари ривожланишининг навбатдаги босқичини ифодалайди. Шу сабабли улар кўп томонлама қўлланишга эга. Виртуал воқеликдан фаол фойдаланиладиган соҳаларга қуйидагилар киради:

- Лойҳалаш, саноат дизайни, прототиплар яратиш;
- Тренажерлар, симуляторлар;
- Томоша, аттракционлар;
- Маркетинг, реклама;
- Масофавий бошқарув;
- Таёргарлик кўриш ва қарор қабул қилишни қўллаш марказлари, ситуацион хоналар;
- Технологик жараёнларни бошқариш;
- Медицина;
- Таълим;
- Архитектура ва дизайн.

Аддитив воқелик — бу дастлаб бир-бири билан боғлиқ бўлмаган соҳаларни: инсон атрофидаги реал объектлар ва виртуал дунёни экранда бирлаштиришдир. Бу интерактив технология фойдаланувчи учун компьютердаги махсус 2D ва 3D объектларни видеокамерадан туширилаётган тасвир устидан қўйиш имконини беради ва шу тариқа, воқелик тўлдирилади. Аддитив воқелик асосида оптик трекинг тизими ётади [11].

Тизим ишлаши учун қуйидаги компонентлар зарур:

- Меткалар — махсус тасвирлар, компьютер моделлари учун визуалидентификаторлар;

- Камера—меткаларни ҳақиқий дунёда «кўради» ва компьютерга видеосигналларни юборади;

- Дастурий таъминот —қабул қилинган сигналларни қайта ишлайди ва виртуал моделларни ҳақиқий объектларнинг тасвири билан бирлаштиради.

#### **Аддитив воқеликнинг афзалликлари:**

- Аддитив воқелик меткаси турли шакл ва ўлчамдаги ихтиёрийтасвир бўлиши мумкин;

- 3D моделлаштириш дастурларида яратилган ҳар хил моделлардан фойдаланиш мумкин;

- технология имкониятлари билан энг мураккаб сценарийларни амалга ошириш мумкин;

- бир неча ўнлаб «тирик 3D меткалардан» бир вақтда фойдаланиш мумкин (бу дегани, бир нечта одам бир вақтда ишлаши мумкин);

- 3D иловаларни интернет орқали кўпайтириш таннарни минимал, лойиҳаларни эса ҳам онлайн, ҳам оффлайн режимида кўриш мумкин;

- Аддитив воқелик инсталляцияси осон масштабланади (ноутбуклардан то видеоредакторга, моно форматдан то стереоскопик форматгача);

- Меткалар бир-бирига ўзаро таъсир қилиши мумкин, улардаги 3D объектлар эса, дунёвий физика қонунларига бўйсунishi мумкин;

- Метка лойиҳаларини матн ва товушлар билан тўлдириш мумкин, шунингдек, унда ичма-ич меню яратиш мумкин;

- Аддитив воқелик диққатингизни жалб қилади ва мураккаб нарсаларни содда визуал тилда тушунтиради;

- Фойдаланувчида интерактив, диққатга сазовор таассурот қолдиради;

- Мижозлар орасида брендни қабул қилишга кўмаклашади;

- Харидорга харид ҳақида қарор қабул қилишга ёрдамлашади;

- Шиддат билан ривожланаётган мобил бозорига кириш имконини беради;

- альтернатив медиаканалларга нисбатан тежамкор;
- фойдаланувчи ҳаракатлари синчиклаб таҳлил қилинади.

Juniper Research маълумотларига кўра, аддитив воқеликнинг кенг ёйилишига қуйидагилар тўсқинлик қилади:

- жамоатчиликнинг беҳабарлиги;
- янги фойдаланувчи интерфейсларининг яратилиши;
- мобил аддитив воқелик AR га эришиш қийинлиги;
- ахлоқий муаммолар;
- хавфсизлик;
- технологик чеклашлар;
- фойдаланувчи учун кутилган натижаларга эришиш;
- сотиш учун бозорни қидириш;
- истеъмолчиларга афзалликларини тушунтириш кераклиги;
- мобил интернетга уланиш зарурлиги.

Аддитив воқелик иловаларини корхоналарга киритиш учун ҳал қилиш зарур бўлган бир қатор ижтимоий, иқтисодий ва техник масалалар бор. Аддитив воқелик дастурларини яратиш энди ривожланмоқда, шунинг учун ундан фойдаланиш ҳақида, унинг афзалликлари ҳақида кенг жамоа яхши билмайди. Аддитив воқелик қўшимчаларини ишлаб чиқарувчи брендлар ўз аудиториясини унинг барча имкониятларидан фойдаланишга ўргатишлари керак. Шунингдек, аддитив воқелик техник таъминотга ҳам жуда боғлиқ. Унга сифатли камера ва маълумот узатиш қурилмалари керак [12].

Фойдаланувчи учун технологиялар қимматини таъминловчи интерфейсни яратиш ҳам асосий муаммолардан бири ҳисобланади. Аддитив воқелик ташкилотларнинг ички ва ташқи алоқалари учун йўл

очади. Ташкилотларнинг аддитив воқеликдан турли соҳаларда фойдаланишлари учун бир неча усуллари қуйида берилган:

- *Туризм ва саёҳат.* Аддитив воқелик ёрдамида, GPS га асосланиб, туристларга ҳаракат маршрутларини кўрсатиш мумкин. Таржимон-иловалар хорижий тилда ёзилган ёзувларни таржима қилади. Диққатга сазовор жойларни кўриш вақтида, шу жой ҳақида тўлиқ маълумот экранга чиқарилади. Туристлар тарихий жойлар, музейларда айланиб юриб, смартфон ёки планшет экранида шу жойга тегишли факт ва рақамларни ўқийдилар.

- *Соглиқни сақлаш.* Аддитив воқелик дастурлари тиббиёт талабаларига хирург усулларини виртуал беморларда қўллашга ёрдам беради. Ёрдамчи қўлланмаларнинг 3D–визуаллаштиришдан касалликни аниқлаш ва ташхис қўйишда фойдаланилади. Рентген, МРТ ва шунга ўхшаш тиббиёт қурилмаларига ҳам аддитив воқеликни қўшиш мумкин.

- *Ишлаб чиқариш.* Аддитив воқелик маҳсулотни лойиҳалаштириш жараёнида моделлаштириш муҳитини яратишда ёрдам беради. Бундай муҳит ходимларни ўқитишда ҳам ишлатилиши мумкин. Масалан, катта машиналарни ишлаб чиқаришда, ундаги нуқсонларни аниқлаш учун актуал маълумотлар билан бирга аддитив воқелик тасвирларидан фойдаланилади.

- *Таълим.* Аддитив воқелик ёрдамида университет биноси ичида ҳаракатланиш мумкин. Мактаб ўқувчилари эса аддитив воқеликнинг 3D–моделларида табиий фанларни ва математик тушунчаларни ўрганишда фойдаланишлари мумкин. Электрон дарсликларни ўқишда, аддитив воқеликка мослашган смартфонларда очиладиган видеороликлар, ҳамда қўшимча маълумотлар сақланган маркерлардан фойдаланиш мумкин.

- *Савдо-сотиқ.* Аддитив воқелик дастурлари маҳсулотларни кўриш ва созлаш ишларини таъминлаш мумкин. Масалан, кийим-кечак магазинларида mijozларга кийимни виртуал кийиб кўрсатиш мумкин.

Ундан ташқари, маҳсулот ҳақида қўшимча маълумот берувчи қўшимча аддитив воқелик материалларини ҳам ишлаб чиқиш мумкин.

- *Спорт ва кўнгул очар тадбирлар.* Спорт тадбирларида аддитив воқелик ёрдамида ўйин зонасига рекламани киритиш мумкин. Аддитив воқелик образлари ва маркерлари ўйин стратегияси ва усулларини тушунтириш учун ишлатилиши мумкин. Автомобиль пойгалари ёки кураш ҳаракатларини виртуал кўришда аддитив воқелик ўйинлари яхши тажриба бўлади. Бу технологияларни кино соҳасида қўллаш, томошабинда катта таассурот қолдиради. Бунда томошабин ўзини кино ичида юргандек ҳис қилади.

### **Биринчи боб бўйича хулосалар**

Биринчи боб бўйича асосий натижалар қуйидагилардан иборат:

- Виртуал ва аддитив воқелик тарихи ўрганиб чиқилди.
- Виртуал ва аддитив воқеликнинг илғор технологиялари ўрганилди.
- Виртуал ва аддитив воқеликнинг афзалликлари ва камчиликлари таҳлил қилинди.

## **II БОБ. ВИРТУАЛ ВА АДДИТИВ ВОҚЕЛИКНИ ЯРАТИШ**

### **1. Виртуал воқеликни моделлаштириш тили (VRML)**

Информацияни Internet да тарқатишнинг навбатдаги катта кадамларидан бири, виртуал воқеликни моделлаштириш тилидан (Virtual Reality Modeling Language, VRML) фойдаланиш бўлиши мумкин. HTML нинг гиперматнли йўлланмалари ва улар билан ишлайдиган браузерлар Internetнинг ахборот ресурслари билан ўзаро ҳаракатларга киришиш учун икки ўлчовли интерфейсни яратади [13].

HTML ва улар билан ишлайдиган браузерлар Internetнинг ахборот ресурслари билан ўзаро ҳаракатларга киришиш учун икки ўлчовли интерфейсни яратади. VRML WWW да уч ўлчамли кенгликни яратиш ва ахборотни янада ёрқинроқ тақдим қилиш йўли билан бу ясси оламни кенгайтиришга қаратилган. VRML 2.0 версиясининг киритилиши бу тилда товушдан, интерфаол ва ҳаракатланувчи объектлардан фойдаланиш имконини беради.

VRML ёрдамида виртуал оламни яратиш жараёни бир нечта босқичларни ўз ичига олади, у VRML нинг алоҳида объектларини ишлаб чиқишдан бошланиб, уларни жойлаштириш ва бутун виртуал сюжетни йиғиш билан тугайди. Сўнгра VRML 2.0 ёки Java тиллари ёрдамида бу оламга жон бахш этиш мумкин. Мазкур бобда VRML объектларини яратиш асосларидан бошлаб уларни жонлантиришгача бўлган барча босқичлар муҳокама қилинади. Халқаро ўргимчак тўрининг қайси қисмидан керакли ахборотни ва дастурий таъминотни олиш тўғрисидаги маълумотлар, шунингдек виртуал оламларни моҳирона яратишга кўмаклашадиган амалий тавсиялар берилади.

Виртуал воқеликни дастурлаштириш тили воқеликни уч ўлчовли тақсимланган интерфаол ўхшатишни ишлаб чиқиш учун мўлжалланган.

Содда қилиб айтадиган бўлсак, VRML ёрдамида киберкенгликнинг гиштчалари яратилади [14].

Халқаро ўргимчак тўри (WWW) гиперматнни (HTML) белгилаш тилини ўз ичига олади, у SGML стандартининг (StandardGeneralMarkup Language —Белгилашнинг стандарт умумлаштирилган тили) ривожланишига сабаб бўлган. Ҳар иккала тил матнни форматлашнинг (ўлчамга келтиришнинг) икки ўлчовли воситалари сифатида ишлаб чиқилган. Марк Д. Песк (Mark D.Pesce), Питер Кеннард (PeterKennard) ва Энтони Парией (Anthony S. Parisi) 1994 йилнинг май ойида WWW бўйича биринчи Халқаро Конференцияда "Киберкенглик" ("Cyberspace") деб номланган маърузани тақдим қилганлар, унда одамнинг уч ўлчовли оламда яшаши ва ишлаши, унинг кўзи уч ўлчовли предметларни кўришга мослашганлиги таъкидланган, WWW га учинчи ўлчовнинг қўшилиши ахборотни тақдим қилишни янада яхшироқ ташкил қилиш имконини беради. Улар виртуал киберкенгликни ишлаб чиқиш воситасини виртуал воқеликни белгилаш тили(VirtualRealityMarkupLanguage) деб аташни таклиф қилганлар. Конференция иштирокчилари бу ғояни қўллаб – қувватлаганлар ва зудлик билан янги стандартнинг маълумотлари учун форматни қидиришга киришганлар. Тез орадаХалқаро ўргимчак тўрининг амалдаги ясси матнли табиати билан келгусидаги уч ўлчовли виртуал воқелик ўртасидаги фарқни таъкидлаб кўрсатиш учун тилнинг номланишидаги Markup (белгилаш) сўзи Modeling (моделлаштириш) сўзи билан алмаштирилган.

VRML стандартини яратиш учун асос сифатида SiliconGraphics компаниясининг OpenInventor дастурий пакети қабул қилинган. Бу уч ўлчовли график объектларни тезкорлик билан ишлаб чиқиш учун мўлжалланган объектга мўлжалланган воситадир(C++). OpenInventor пакети ёрдамида кўнгилочар ўйинлар саноатида фойдаланиладиган KeystoneInterchangeFormatниваX3H3 3D ANSI/ISO мегафайлларнинг



спецификациясини ҳам ўз ичига оладиган бир нечта стандартлар яратилган.

VRML спецификацияларини ишлаб чиқишда учта масала асосий мақсад қилиб қўйилган:

- Платформага боғлиқ бўлмас;
- Кенгая олиш қобилияти;
- Паст тезликли уланишларда ишлай олиш қобилияти.

*VRML объектлари.* Виртуал воқелик VRML блокларидан таркиб топади, улар VRML оламлари (VRML worlds) деб аталади. Тилнинг спецификацияси ўзида элементлар (nodes) деб аталувчи ва сфера, куб ёки цилиндр типигаги турли-туман содда объектларни, шунингдек, чўққилар (vertices) ва томонлар (faces) жамланмасидан таркиб топган объектларни яратиш учун мўлжалланган командалар жамланмасини тақдим қилади.

Томонлар — бу объектни чегаралаб турувчи ясси юзалардир, чўққиларбу — ушбу текисликларнинг кесишиш бурчакларининг чўққиларидир. Масалан, куб олтита квадрат томонлардан таркиб топади, уларнинг ҳар бири тўртта чўққига эга бўлади. Бутун кубни белгилаш учун саккизта чўққи талаб қилинади, чунки ҳар бир чўққи учта томон учун умумий бўлиб ҳисобланади.

VRML тили содда объектларни комбинациялаш йўли билан кўпроқ мураккаб бўлган объектларни ҳам яратиш имконини беради. Бу шажаравий толдир — дунёга келтирилган ёки қизлик (child) объектларота-она (parent) объектларнинг хусусиятларини мерос қилиб олади. Агар мураккаб объект, масалан, одам танасининг модели яратилган бўлса, у ҳолда бутун тана учун аниқланган ҳар қандай хусусият (масалан ранг) объектнинг барча таркиб топтирувчи содда қисмлари учун, масалан, бош, оёқ, қўл ва ҳоказолар учун ўз кучини сақлаб қолади.

*VRML оламлари.* Яхлит қилиб йиғилган VRML объектларининг жамланмаси VRML олами деб аталади. Турли объектларнинг жойлашган

ўрни ва ўзаро муносабатларини аниқлаш учун VRML координаталар системаси ёрдамида уларнинг нисбий ўлчамлари ва позицияларини кўрсатиш талаб қилинади. Бунинг устига тил сизнинг оламингиздаги ёритилганликни, шунингдек олдиндан ўрнатилган пейзажларни ўрнатиш имконини беради [15].

Тилнинг 1.0 версияси статик бўлган. Фақатгина томошабиннинг, унинг виртуал кенглик бўйлаб ҳаракатланишида, жойлашган жойи ўзгарган. VRML 2.0 динамик оламларни яратиш имконини беради.

Энди объектлар мустақил равишда ҳаракатланишлари мумкин. Бундан ташқари уч ўлчовли товуш ҳамроҳлигини қўшиш мумкин, яъни товуш тингловчининг товуш манбаига нисбатан жойлашган ўрнига қараб ўзгариши мумкин. Объектнинг ўзини қандай тутишини аниқлаш имконияти VRML нинг яна бир янги хусусияти бўлиб ҳисобланади. Ўзини тутиш (behavior) — бу объектнинг бошқа объектлар, томошабин билан муносабатлари, ёки бошқа параметрлар, масалан, вақтга боғлиқ бўлган характеристикасидир. Ўзини тутиш Java тилида скриптлар ёки апплетлар билан тасвирланади. Масалан “аквариум” деворига яқинлашганда виртуал “балиқ” кўрқиб кетиши ва сузиб кетиши мумкин [7].

Ҳозирги кунда, VRML 2.0 ни якуний деталлаштириш босқичида жуда кам сонли браузерлар ва қўшимча уладиган дастурий модуллар бу стандартни қўллаб-қувватлайди. Ҳозирги вақтда Netscape Live3D модулидан фойдаланиш энг яхши натижа беради. Ушбу модул NetscapeNavigator браузерининг стандарт қисми бўлиб ҳисобланиши, шунингдек, MicrosoftInternetExplorer билан ишлаши туфайли виртуал оламларни Live3D форматида кўриш қобилиятига эга бўлган потенциал аудитория тобора кенгайиб бормоқда.

VRML нинг тўлақонли оламини Internet бўйлаб узатиш кўп вақтни олади, бу аудиторияни амалда “ғаройиб” виртуал таассуротларни қидирадиган кам сонли томошабинлар даражасигача қисқартириб қўяди.

Виртуал оламлар уларни куриш учун жуда катта вақт сарфини талаб қиладиган ўта мураккаб оламлар бўлиши мумкин.

Бироқ Web-саҳифаларга махсус эффектларни киритиш VRML нинг жуда яхши ва узоқ вақт юклашни талаб қилмайдиган қўлланилиши бўлиб ҳисобланади. Live3D модул энг оммавийлашган браузер билан биргаликда тарқалиши ва оммавийлиги бўйича иккинчи ўринда турадиган браузер билан ҳам ишлаши туфайли HTML-саҳифаларга VRML нинг унчалик катта бўлмаган сахна кўринишларини киритиш мумкин бўлади. Шу нуқтаи-назардан VRML тили турли-туман трюкларни киритишда яхши восита ҳисобланади [16].

*VRML нинг базавий синтаксиси.* VRML файллари — булар одатдаги матнли ASCII – файллардир (гарчи улар кўпинча Internet бўйлаб узатишни соддалаштириш учун GNU утилитлари билан сиқилган бўлсада), бу шуни билдирадики, улар ҳар қандай матн редактори воситасида яратилиши мумкин. Аввал VRML нинг кодлар даражасидаги базавий элементлари билан яқинроқ танишибчиқамиз.

Листингда VRML нинг NetscapeNavigator браузери ойнасида қизил конус ҳосил қиладиган, оқ фонда жойлашган содда файлига мисол келтирилади.

```

#(VRML V1.0 ascii
Separator { Info {
  string "Special Edition, Using HTML, 3rd Edition" } DEF
BackgroundColor Info {
  string "1 1 1" } Separator {
  Material {
    diffuseCoior 1 0 0 # the color red
    Cone {} } }
```

VRML да изохлар # символдан кейин листингнинг ҳар қандай жойида жойлашиши мумкин (масалан, thecolorred сўзига ўхшаб). Коднинг

биринчи қатори ҳам изох бўлиб ҳисобланади, бироқ у мажбурий бўлиб ҳисобланади ва браузерга файлнинг типини ва VRML нинг версиясини кўрсатади.

Ушбу ҳолатда биринчи қатор шуни кўрсатадики, файл VRML нинг 1.0 спецификациясига бўйсунди ва унда ASCII символларининг стандарт жамланмасидан фойдаланилган, VRML 2.0 нинг файллари одатда #VRML V2.0 utf8 қатори билан бошланади, бу шуни билдирадики, улар VRML 2-0 спецификацияси бўйича қурилган ва символларнинг ASCII га асосланадиган халқаро жамланмасидан фойдаланилади [17].

Файлни қуриш формаси етарлича ихтиёрий — бунда ораликлар, ёки табуляция ва қаторларни кўчиришдан, қайси бири керак бўлса, фойдаланиш мумкин. Масалан, кўриб чиқиладиган файл жуда ихчам тарзда қуйидагича ёзилади:

```
#VRML V1.0 asciiSeparator{Info{string "Special Edition, Using
HTML,3rd Edition")}
DEF BackgroundColorInfo{string "1 1 1") Separator
{Material{diffuseColor 1 0 0}
Cone{}}}
```

Дастурдаги изохлар дастурни тушунишни анчагина соддалаштиради.

*Элементлар ва майдонлар*

C, C++ ёки Java тиллари каби VRML тилида ҳам фигурали қавсларга боғланган маълумот блоклари киритилади. VRML файллари элементларнинг тасвирланишидан таркиб топади (nodes), улар қуйидаги кўринишга эга бўлади:

```
NodeType{ configurationinformation }
```

Элементнинг типи (NodeType) VRML нинг спецификацияси билан аниқланади.

Юқорида келтирилган мисолда элементларнинг тўртта типидан фойдаланилади: Info, Separator, Material ва Sphere.

Майдонлар ёрдамида элементларни конфитгурациялаш. Конфигурация тўғрисидаги фигурали кавсларда жойлашган маълумот майдонлардан (fields) таркиб топади. Масалан, info элементлари олдинги мисолда string типдаги майдонларга эга, Material элементи эса — diffuseColor типдаги майдонга эга. Одатда ҳар бир майдон ном ва қийматга эга бўлади. diffuseColor майдон учта сондан ташкил топган 1 0 0 қийматга эга, улар sphere нинг элементи учун рангни кўрсатади (одатдаги тартибда: қизил, яшил, кўк).

Майдонларнинг қийматлари алоҳида сонлар, сонлар гуруҳлари, символлар қаторлари, захираланган сўзлар (командалар), мантиқий қиймат ва ҳоказолар билан ифодаланиши мумкин. Баъзи бир майдонлар бир нечта қийматга эга бўлиши мумкин; Бундай ҳолда улар утирлар билан ажратилади ва квадрат кавсларга жойлаштирилади.

Масалан, diffuseColor майдонга Material элементнинг рангининг учта қийматини қўшиб ёзиш мумкин:

*Material( diffuseColor [1 0 0,0.5 0.5 0,0 1 0] }*

Шу тариқа объектнинг турли томонларининг рангини аниқлаш мумкин.

Элементларнинг номланиши. Ҳар қандай элементга, кейинчалик унга мурожаат қилиш учун, ном бериш мумкин. Бу худди листингда кўрсатилгандай DEF префикси ёрдамида бажарилади:

*DEF BackgroundColorInfo{ string "1 0 0" }*

Бу ерда BackgroundColor элементнинг фоннинг рангини аниқлаш учун фойдаланиладиган махсус номи бўлиб ҳисобланади. Умумий ҳолатда ҳар қандай элементга ҳар қандай номни бериш мумкин бўлади.

Фоннинг рангини юқорида кўрсатилган аниқлаш VRML 1.0 стандарти билан қўллаб-қувватланмайди — бу VRML нинг баъзи бир браузерлар қабул қиладиган кенгайишдир. Агар бу кенгайишни қабул қилмайдиган браузердан фойдаланилса, берилган элемент инкор қилинади,

фон эса браузердааввал ўрнатилган (по умолчанию) рангга эга бўлади (қоидага кўра, қора). Ушбу бобнинг мисолларида фон рангининг бундай тарзда танланишининг сабаби шундаки, VRML 1-0 спецификациясида фон рангини белгилаш усули расмий равишда кўрсатилмаган; бироқ бу мисоллар босма формада тўғри қайта тикланиши учун муҳимдир [18].

*VRML олами объектларининг шажараси.* VRML оламига VRML нинг содда объектларининг шажараси сифатида қараш мумкин. Бу тилда Separator элементида объектни ўз ичига олган контейнер сифатида фойдаланилади. SeparatOr контейнери ичида баъзи бир элементлар геометрияни белгилайди (VRML объектларининг чўққилари ва томонлари), бошқаларидан эса содда объектларни мураккаброқ объектларга гуруҳлаштириш учун фойдаланилади. Шу тариқа VRML да объектларнинг шажараси яратилади. separator элементи бу элементнинг объектларининг хусусиятларини тасвирлаш учун дунёга келтирилган элементларни ўз ичига олиши мумкин.

*VRML нинг содда объектлари.* VRML 1.0 спецификацияси улар ёрдамида энг содда геометрик объектларни қуриш мумкин бўладиган саккизта элементни ўз ичига олади. VRML 1.0 нинг барча объектлари битта ёки бир нечта ўхшаш элементлардан қурилади. Уларнинг таркибига каноник геометрик жисмларни тасвирлайдиган тўртта элемент киради: битта элемент ASCII – матнни тасвирлаш учун ва учта элемент — ихтиёрий икки ёки уч ўлчовли объектларни яратиш учун.

VRML конусдан ташқари яна учта элементар объектни: куб, цилиндр ва сферани қуриш имконини беради. Гарчи сфера, конус ва цилиндр типдаги элементлар эгри юзаларни билдирса-да, VRML нинг файлини браузер томонидан изоҳлашда бундай турдаги барча объектлар тесселяция (tessellation — шахмат тахтаси кўринишидаги мозаикани яратиш) деб аталувчи жараён ёрдамида чўққилар ва томонларга конверцияланади. Бу қандай ишлашини кўриш учун эгри юзага эга бўлган мисоллардан бирини

VRML браузерига юклаб, томоннинг (текисликнинг) фигурасини таркиб топтирувчиларни кўриш учун ясси соялар режими ёкилади (Live3D)да VRML сахнасида сичқончанинг ўнг тугмачасини босиб, Lights (Ёруғлик) FlatShading (Ясси соя) ни танланади [20].

## **2. O3D – виртуал борлиқни яратиш платформаси**

O3D бу интерактив учўлчовли веб қўшимча яратиш имконини берадиган браузерда акс эттирилган дастлабки очик коди билан JavaScript API бўлиб ҳисобланади. Масалан, 3D уйинини, маҳсулотнинг тақдимотини, 3D кўринишини ёки умуман виртуал дунёни яратиш мумкин ва фойдаланувчилар сайтга кириб қўшимчаларни ишга туширмасдан кўшиш ва турли хил ҳаракатларни бажариш имконини беради [21].

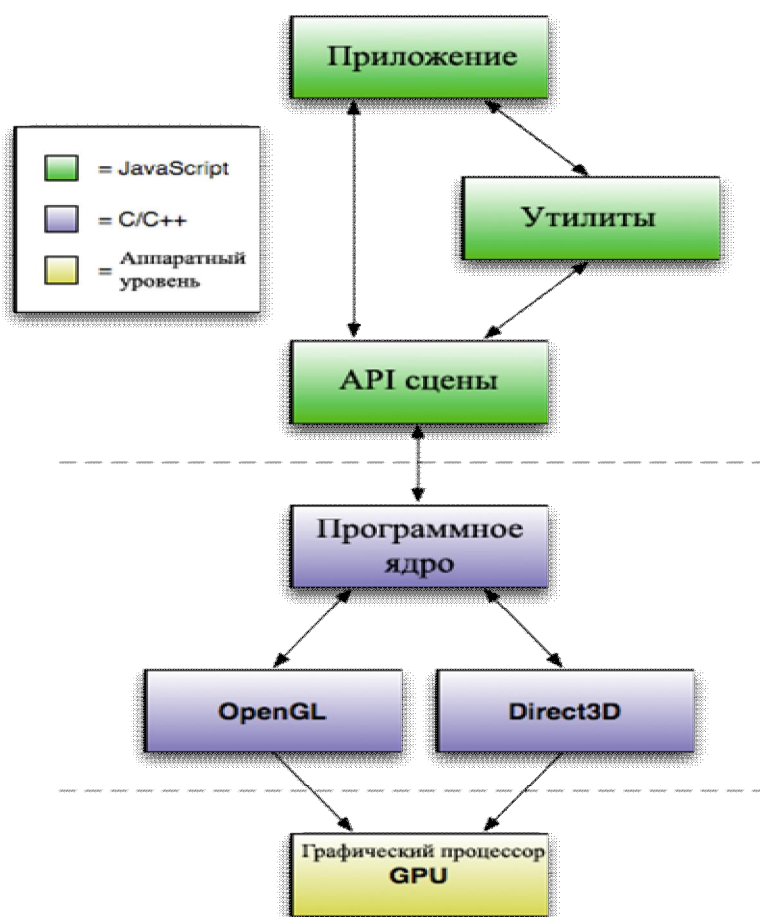
O3D базибир ўзгачаликлари билан харидорнинг веб қўшимчаси бўлимини кенгайтиради. Уларни ушбу даражаларда олиб қарасак бўлади:

- Тизим: O3D браузер учун плагин бўлиб, уни урнатгандан сўнг браузерни 3Dқўшимчаси акс эттира олади ва у билан ишлашга имкон яратиб беради. Асосийси у Windows, Macintosh, ва Linux (TBP) платформаларида ҳам мумкин.

- Тўлдириш: Бугунда веб HTML, расм ва видео файллар билан тўла. Ишлаб чиқарувчилар учун ҳужжатлаштириш конвертерини қандай яратиш ва ҳар хил 3D тўлдиришлар йўқловчиси ҳақида маълумотдан иборат. O3D мисол сифатида COLLADA конвертеридан туради, унинг ёрдамида COLLADA форматидаги файлларни импорт қилиш мумкин, масалан, эркин ва очик стандарт 3D элементлари учун таниқли SketchUp, 3ds Max, ва Maya 3D редакторларини кўтаради. Уни эркин қўллаш ёки конвертерларни ва йўқловчиларни бошқа форматлар учун ёзиш мумкин.

- Дастлабки код: 3D графика билан ишлаш учун O3D JavaScript-да ёзилган ва API – дан иборат. У стандарт методлардан ва JavaScript воқеяларини қайта ишлашдан фойдаланади.

O3D API JavaScript ва 3D графикаси билан(хеч бўлмаса юзаки) таниш веб ишлаб чиқарувчилар учун йўналтирилган. Албатта, Firefox, Safari, Google Chrome, ва Internet Explorer каби браузерлар фойдаланувчилари билан ўз ижодини бўлишишни хоҳлаганлар, O3D уйинларини ишлаб чиқарувчилар учун ҳам йуналтирилган. Сабаби O3D қўшимчалари браузер учун плагин сингари ишга туширилади, фойдаланувчилар катта ҳажмдаги қўшимчаларни тортиб олиш, уларни шахсий компьютерларга ўрнатиш ва алоҳида қўшимча сифатида ишга тушириш шарт эмас. O3D API тўғридан –тўғри GPU дастурлаштириш, ЯВУ швейдери ёрдамида унумдорликни кучайтиради, бу эса дастурлаштириш рендеринги олдида катта ютуғидир.



Расм-5. O3D архитектураси кўриниши



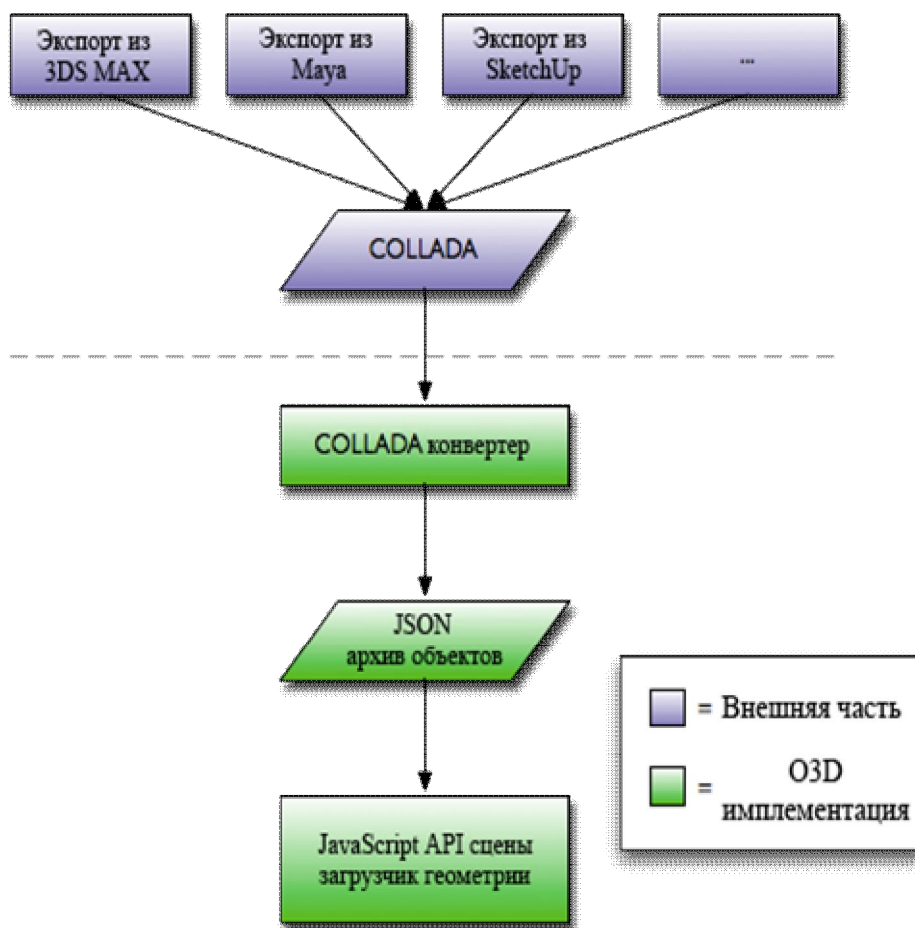
*Архитектураси.* Юқорида кўрсатилган чизилмада O3D элементларининг оддийлашган тузилиши берилган ва улар аппаратлар ёрдамида амалга ошириладиган бирикмалар билан ўзаро хизмати кўрсатилган.

Яшил ранг билан O3D асосий компонентлари белгиланган:

- JavaScript O3D қўшимчасининг ўзи;
- Ишлаб чиқаришда сизга ёрдам берадиган кўпқўлланиладиган код элементлари JavaScript утилитлари;
- График кўринишининг API – си, яъни иловадан фойдаланадиган синфлардан ва функциялардан иборат бўлган JavaScript API.

JavaScript O3D қўшимчаси коди бутунлай браузер орқали юкланадиган HTML ҳужжати таркибида тўлиқ жойлашган. O3D – да ишлаб чиқарув учун керак бўладигани фақатгина браузерга ўрнатилган плагин ва JavaScript кодини ёзиш учун матнли редакторлар лозим. Архитектуралик чизмада кўрсатилганидек, O3D компьютердаги аппаратлар ёрдамида амалга ошириладиган бирикмалар (GPU видео процессори назарда тутилган), OpenGL ёки Direct3D кутубхонаси билан ўзаро ҳаракатдадир.

*Импорт қилинадиган элементлар.* O3D библиотекаси уч ўлчовли объектларнинг тасвирланишини иловадан тўғридан –тўғри кўрсатиб беради (нуқталар массиви, чизиқлар массиви, полигонлар массиви, полигонлар тасмалари ва полигонлар ишончи). Масалан, O3DCOLLADA фарматидаги объектларнинг импортини намойиши қилиш учун оддий конвертерни назарда тутди. Навбатдаги чизмада кўрсатилгандек COLLADA конвертери Autodesk 3ds Max, Maya, ва Google SketchUp каби таниқли 3D редакторларда яратилган объектларни импорт қилади [22]:



Расм-6. Collada API кўринишида импорт қилинган объектлар

O3D – да API кўринишлар графикларини яратиш, трансформациялаш ва рендеринглаш учун қўлланилади. Графикларни трансформациялаш учун 3D “дунё” қўшимчасининг асосий маълумотини курувчи жойлашувлар, объектларнинг ўлчами, материаллар ва швейдерлар хакида маълумотлар ишлатилади. Графикни рендеринглашда фойдаланувчининг икки ўлчовли ойнасида 3D дунёсининг акс эттирилиши билан масала барқарор этилади. Рендеринг жараёни кўрсатилган элементлар учун масъулдир:

- У фойдаланувчи учун қайси объектлар кўриниши керак ва аксинча эмаслигини ҳисоблайди;
- Объектлар кўринишларини ва уларнинг аниқ чизмалари учун керакли трансформацияларни ишлаб чиқаришни ҳисоблайди;

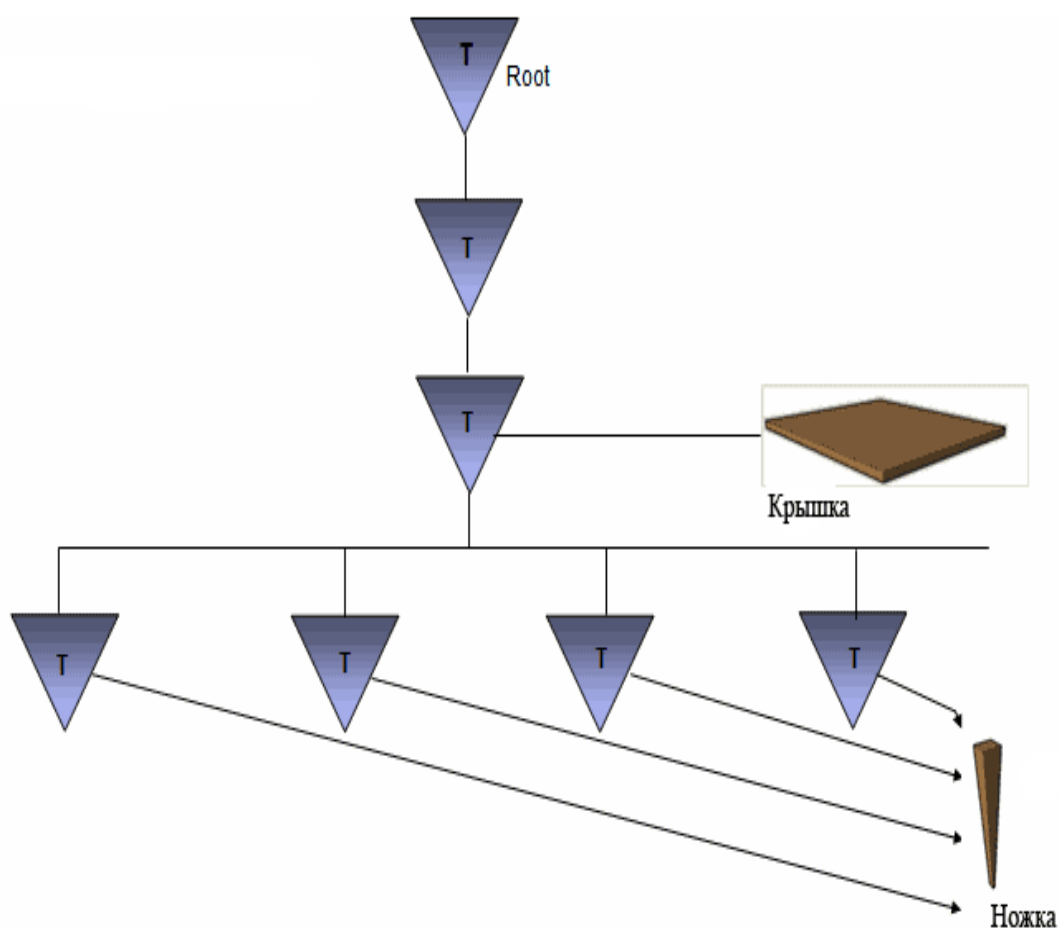
- Шаффофлик, 3D дунёсининг турларининг махсус эффектларининг амалга оширилиши ва HUD элементлари учун жавоб беради.

Аввал, қўшимчанинг 3D дунёси графигининг трансформацияси учун механизмлар ва акс эттирилган элементларнинг асосий рендерингини яратади. Кейинчалик, рендеринглашда 3D таркибини акс эттиришда керак бўладиган O3D график трансформация ҳақида маълумотлар йиғади. Бундай ёндошув, «immediate – mode йули»(аниқ чизмалар методи бевосита чақирув) билан тасдиқлайди, бунда ҳар бир аниқ чизмалар гуруҳи алоҳида чақирилади ва амалга оширилади.

*Трансформациялар.* Трансформациялар уч ўлчовли бўшлиқда объектлар ўлчами ва жойлашуви берилган матрицалар билан белгиланади. Трансформация графиги бу ота–она/меросхўр иерархияда таркиб топган трансформация коллекциясининг кетмакетлигидир. Ҳар бир қўшимчанинг график трансформацияси иерархия дарахтининг тепасида илдизга оид ота–оналик боғичидан(root), кейин эса унга тармоқланиш натижасида ҳосил бўлган трансформациялар шохлари қўшилиб борилади. Юқорида айтиб ўтилганидек, иерархия дарахти ва пастки шохдаги тармоқланиш натижасида ҳосил бўлган трансформацияси учун график трансформацияда кумулятив эффект мавжуд [23].

*Шакиллар.* Трансформация унга бириктирилган бир ёки бир неча шакллардан туриши мумкин. Шакл жойлашган ва ўзининг ўлчамларига эга бир бутун ҳолдаги геометрик кенгликнинг бир бўлаги сифатида белгиланади. Шакл ўз навбатида ҳар бирига турли материал(маълумот) мос келадиган примитивлардан(оддий) ташкил топган бўлади. Шакллар мустақил ва хусусий локал координат тизим билан жойлашган трансформация билан белгиланади. Масалан, агар 4 оёқли стол тузишни истасак, шубҳасиз бир оёғининг моделини ишлаб чиқамиз ва уни 4та турли позицияда жойлаштирамиз. Ундан кейин столнинг бош қисмини

тайёрлаймиз ва уни оёқлар устига жойлаштиришни истаймиз ва натижада, столни кўчирганда ҳар бир элементини ҳаракатга келтиришни хохламаймиз албатта шунинг учунуларни бир бутунликка тўплашни истаймиз. Стол учун график трансформациялари тахминан шундай кўринишга эга бўлади:

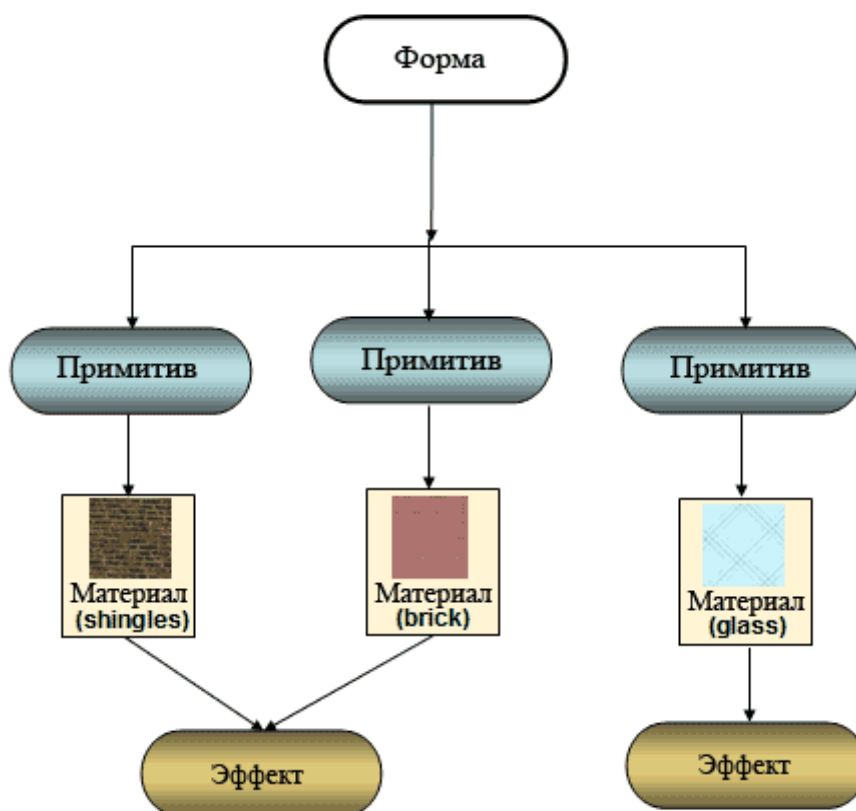


Расм-7. Стол трансформациясининг график кўриниши

*Материаллар.* Ҳар бир примитив материални аниқлаб берадиган параметрлардан иборат. Материал кўп примитивларда қўлланилиши мумкин. Агар 3D редакторидан қандайда бир моделни импорт қилаётган бўлсак, примитивлар автомат таризда керакли материални сурайди. Примитив материални аниқловчи параметрни ўз ичига олади. Материал

эса амалга оширилаётган эффектни аниқлайдиган параметрни ўз ичига олади.

*Эффектлар.* Материал оширилаётган эффектни белгиловчи ва керакли ҳолатни белгиловчи параметрдан ташкил топган. Эффект эса ўз навбатида керакли шакл ҳосил қилиш учун қандай қилиб пикселларни кенгайтиришни тепадаги шейдерларни ва фрагментли (пикселли) шейдерларни ўз ичига олади. Тарқалиш ва нурнинг майинлиги, товланишлар ва ҳоказолар каби материаллар параметри ҳам эффектлар билан қўлланилади.



Расм-8. Эффектлар ва примитивлар графиги

*График трансформацияларни яратиш.* О3D ёрдамида сиз икки йулнинг бирисини танлаб график трансформацияларини яратишингиз мумкин:

1. Ҳар хил 3D редакторлардан объектларни импорт қилишда, SketchUp, 3ds Max, ва Maya каби редакторлар COLLADA форматидаги объектларни экспорт қила олади ва O3D учун эса ўз навбатида шу форматдаги конвертер мавжуд. Google 3D Warehouse тақдим этган моделлар айнан шу форматда сақланади. (Ушбу конвертерни кўриш принципини қўллаш орқалибемалол ўзингизнинг экспорт/импорт қилувчингизни қандайда бир формат учун ёзишингиз мумкин). Ушбу сайтнинг намуна бўлимида 3D кўринишининг импорт қилинган таркибининг мисолларини топиш мумкин. Ишлаб чиқарувчилар учун хужжатлаштириш қисмида объектларнинг импорти тўғрисида кўпроқ маълумот олиш мумкин.

2. Объектларнинг жойлашуви координациясидан, умумийлигидан, ранги ва эффектидан таркиб топган ёзувдан(scratch) график трансформацияларни тузиш, сўнгра берилган қўшимчанинг уч ўлчовли кенлигида объектларни қандай қилиб жойлаштириш кераклигини аниқлаш мумкин. Ушбу усул сайтдаги кўплаган мисолларни кўришда қўлланилади.

Шунингдек айрим моделларни импорт қилиш, базилари учун эса зарур маълумотли ёзувни қўллаш мумкин. Кўп ҳолларда O3D қўшимчаси аниқ ифодасини берган кўриниш ва проекцияли матрицаларини қўллаш орқали кўринишнинг аниқ чизгиларини чизади. Шу билан бирга O3D қуйиладиган белгилар, шаффофлик, ялтираш, кўп миқдордаги кўриниш ва ҳоказолардан иборат аниқ чизгини берадиган тўлиқ механизмни амалга оширишни таъминлайди.

*Хотирани бошқариш.* O3D қўшимча объектини яратмоқчи бўлсак, у автоматик таризда махсус пак хотирасига қўшиб қуйилади ва тасодифий юқотилишдан ҳимоя қилинади. Ҳар доим пакдаги ҳисоблар бирликда ҳисобланса, бу объект бошқа объектга юборади. Ҳар бир объектаги фақат битта далилни пак ўзида сақлайди. Функция `pack.removeObject` пактаги фақат битта далилни ўчиради, лекин функция `pack.destroy` пакдаги барча

далилларни ўчиради. Агар бу функцияларнинг қайсидири ишга туширилса, объекттаги оқирги далиллар хотирадан ҳам ўчирилади.

### **3. Дастурий таъминотни ишлаб чиқиш учун фреймворкни танлаш**

Ишлаб чиқишни бошлашдан олдин Фреймворкни (кутубхоналар жамланмасини) танлашни амалга ошириш зарур бўлади.

Альтернативалар (муқобилликлар) шарҳи

OpenCV – компьютернинг кўриш алгоритмларининг, тасвирларга ишлов бериш ва очиқ калитга эга бўлган умумий белгиланишдаги сонли алгоритмларнинг кутубхонасидир. C/C++ да амалга оширилган, шунингдек, Python, Java, Ruby, Matlab, Lua ва бошқа тиллар учун ишлаб чиқилмоқда. Академик ва тижорат мақсадларида бемалол фойдаланилиши мумкин — BSD лицензияси шартларида тарқатилади [24].

Vuforia SDK – бу мобил қурилмалар учун мўлжалланган дастурий таъминот бўлиб, аддитив воқелик иловаларини яратиш имконини беради. У компьютер кўриш технологиясидан ясси тасвирлар ва содда 3D-объектларни реал вақт режимида таниб олиш ва кузатиб бориш учун фойдаланади. Тасвирларни бундай қайд қилиш имконияти уларга мобил қурилманинг камераси орқали қаралганда 3D–моделлар каби виртуал объектларнинг реал оламда жойлашиши ва йўналтирилишини аниқлаш имконини беради. Виртуал объектнинг жойлашиши ва йўналиши реал вақтда кузатиб борилади, шу сабабли томошабиннинг объектга бўлган нуқтаи-назари уларнинг тасвирга бўлган нуқтаи-назарига тегишли бўлади, чунки виртуал объект оламнинг реал сахнасининг бир қисми бўлиб ҳисобланади.

Vuforia SDK турли туман 2D ва 3D нишонли типларни, маркерсиз нишонларни ҳам ўз ичига олган ҳолда, ушлаб туради. SDK нинг қўшимча имкониятлари виртуал тугмачалар, мақсадли танлашни бажариш

тасвирлари ёрдамида окклюзияни маҳаллий аниқлашларни, шунингдек бажариш пайтида мақсадли жамланмаларни дастурий тарзда яратиш ва ўзгартириш имкониятини ўз ичига олади.

Vuforia C++, Java, Objective – C ва Net тиллар учун API ни таъминлайди. Unity ўйин движогининг кенгайиши мавжуд. Шу тариқа SDK IOS ва Android тилларни она тили сифатида ушлаб туради ва шу билан биргаликда аддитив воқеликнинг иловаларини Unity да ишлаб чиқиш имконини беради, улар ҳар иккала платформага осон кўчириб ўтказилади. Шу сабабли Vuforia дан фойдаланиш билан ишлаб чиқилган иловаларни мобил қурилмаларнинг кенг спектри билан, iPhone (4/4S), iPad, Android ва ОС Android 2.2 версияси ёки юқорироқ ҳамда FPU га эга бўлган ARMv6 ёки 7 процессор бошқаруви остидаги телефонлар ва планшетларни ҳам ўз ичига олган ҳолда, бирлаштириш мумкин.

ARToolkit – бу аддитив воқеликка эга бўлган иловаларни яратиш учун компьютер кузатиб бориши кутубхонасидир. Бунинг учун у видеокузатувнинг имкониятларидан, камеранинг реал вақт режимида квадрат физик маркерга нисбатан реал жойлашиши ва йўналишини ҳисоблашдан фойдаланади. Камеранинг реал ҳолати маълум бўлганда виртуал камера худди ўша нуктада жойлашиши мумкин ва 3D модель реал маркерга қўйилади. Шу тариқа ARToolKit тўлдитрилган воқеликда иккита муаммони ҳал қилади: нуктаи-назарни ва объектларнинг виртуал ўзаро ҳаракатларини кузатиб бориш [25].

ARToolKit дастлаб фан ва технологиялар институтининг HirokazuKato Нара томонидан 1999 йилда ишлаб чиқилган ва Вашингтондаги НИТ университетининг лабораториясида чиқарилган. Ҳозирги кунда у SourceForge га жойлаштирилган очиқ бошланғич лойиҳа сифатида фаолият кўрсатади. ARToolKit дан жуда кенг фойдаланилади (2004 йилдан бошлаб кутубхонага 160000 мартадан ошиқ мурожаат қилинган).



Metaio SDK – аддитив воқеликнинг мобил иловаларини яратиш учун тайёр кутубхона. Янада аниқроқ ишлаш учун OpenGL ва SLAM услублардан фойдаланади.

String – iOS қурилмаларга мўлжалланган мобил иловаларни яратиш учун кутубхона.

жадвал–1. Альтернативалар ва критериялар

| Критерия<br>Альтер-<br>нативалар | Кросспла<br>тформаси              | Хужжатларнинг<br>ва<br>адабиётларнинг<br>мавжудлиги | Турли<br>дастурларнинг<br>қўллаши            | Қўшимча<br>сервисларни<br>танлаш (баҳо) | Лицензия<br>турлари                   |
|----------------------------------|-----------------------------------|-----------------------------------------------------|----------------------------------------------|-----------------------------------------|---------------------------------------|
| OpenCV                           | iOS, PC,<br>Android,<br>Linux     | Кўп                                                 | C/C++, Python,<br>Java, Ruby,<br>Matlab, Lua | Аъло                                    | Boost<br>Software<br>License          |
| Vuforia                          | iOS,<br>Android,<br>Unity         | Етарлича                                            | C++, Java,<br>Objective-C,<br>.Net           | Аъло                                    | Free+<br>Commerci<br>al SDK<br>option |
| ARToolKit                        | iOS,<br>Android,<br>Unity         | Кўп                                                 | Java,<br>Objective-C                         | Кам                                     | Free+<br>Commerci<br>al SDK<br>option |
| Metaio<br>SDK                    | iOS,<br>Android,<br>Unity,<br>WEB | Кам                                                 | Java,<br>Objective-C,<br>.Net                | Аъло                                    | Free+<br>Commerci<br>al SDK<br>option |
| String                           | iOS,<br>Unity                     | Кам                                                 | Objective-C,<br>.Net                         | Жуда кам                                | Free+<br>Commerci<br>al SDK<br>option |

Кўпроқ муҳим бўлган критерияларни (мезонларни) ҳисобга олиш билан альтернативаларни шарҳлаш шуни кўрсатадики, Vuforia тизимини ишлаб чиқиш учун энг оптимал Фреймворк бўлиб ҳисобланади. Vuforia – фойдаланиш текин бўлган кутубхона, у доимо такомиллашиб боради, натив иловалар билан ишлаш имконини ҳам, Unity махсус ўйин движоги ёрдамида кроссплатформали иловаларни яратиш имконини ҳам беради.

Тасвирларни қайд қилиш имконияти ишлаб чиқувчиларга 3D-моделларва медиаконтентлар каби виртуал объектларни мобил қурилмаларнинг камералари орқали қарашда реал образлар билан боғлаш

асосида жойлаштириш ва йўналтириш имконини беради. Виртуал объект реал образда шундай йўналтириладики, бунда томошабиннинг нуқтаиназари бош эффект – виртуал объект реал оламнинг бир қисми эканлигини хис қилиш учун уларга бир хил муносабатда бўлсин.

Vuforia нишонларининг турли-туман 2D ва 3D – типларинимаркерсиз ImageTargetларини, Multi-Target уч ўлчовли нишонларини, шунингдек, объектларни сахнада таниб олиш учун уларни ажратадиган репер маркерларини ҳам ўз ичига олган ҳолда ушлаб туради. Қўшимча функциялар «Виртуалтугмачалар» («VirtualButtons») деб аталувчи тугмачалар ёрдамида тўсиқларни аниқлашни, нишонларни детектирлаш ва ўз-ўзидан дифицирланадиган код доирасида нишонларни дастурий тарзда яратиш ва реконфигурациялаш имкониятини ўз ичига олади [26].

Vuforia –C++, Java, Objective-C, ва .Net тилларида иловаларни дастурлаштириш интерфейсларини Unity ўйин движоги билан интеграциялаш орқали тақдим қилади. Шу тариқа SDK iOS ва Android учун натив AR-иловаларни ишлаб чиқишни ушлаб туради ва шу билан бир вақтнинг ўзида Unity да ишлаб чиқишни ҳам кўзда тутди, унинг натижалари ҳар иккала платформага осон кўчирилиши мумкин. Vuforia платформасида яратилган аддитив воқеликнинг иловаларини қурилмаларнинг кенг спектри билан, iPhone, iPad ни, 2.2 версия ва процессорли Android смартфонлар ва планшетларни, ҳисоблашларни сузувчи утир билан амалга ошириш имконини берадиган ARMv6 ёки 7 архитектурасидан бошлаб ўз ичига олган ҳолда, бирлаштириш мумкин.

C++ тилида ёзилган ва ўзининг интерфейсларида нишонни аниқлашдан (target) тортиб тасвири рендеринглашгача бўлган ҳамма нарсани яширадиган QCAR кутубхонаси платформанинг ядроси бўлиб ҳисобланади.

Платформа кутубхонадан ташқари қуйидагиларни ҳам ўз ичига олади: ишлаб чиқувчилар учун iOS ва Android Vuforia SDK; TargetManager — нишонларни яратиш ва бошқариш учун тизим; шунингдек уларга нишонларни сақлаш ва яна бошқа функцияларни олиб чиқиш мумкин бўлган web сервислар жамланмаси (VuforiaWebServices). Платформа ўта динамик тарзда ривожланмоқда ва унга тез-тез янги қисмлар қўшилади.

*Targets—нишонлар.* Реал оламга бирор нарса қўшиш учун бу оламдаги қандайдир бир объектга “ёпишиб олиш” зарур бўлади. Нишонлар айнан шу мақсад учун хизмат қилади. Бошқача қилиб айтганда, нишон – бу қандайдир бир реал объект бўлиб, уни билган ҳолда бизнинг иловамиз виртуал объектларни керакли жойларга керакли пропорцияда қўйиб чиқиши мумкин. Vuforia нишонларнинг жуда катта жамланмасини тақдим қилади:



Расм-9. Нишонларнинг кўриниши

*Imagetargets* – нишонларнинг ўзида одатдаги тасвирни, масалан журналнинг муқовасини, фотосуратни ёки янги фильмнинг афишасини тақдим қиладиган базавий туридир. Тасвир ўзига хос икки ўлчовли штрих-код ролини ўтайди, фақат унда оқ-қора сохалар бўлмайди. У бўйича биз айнан қайси тасвир камера объективига тушганлигини, шунингдек, унинг фазода жойлашиши ва масштабини аниқлашимиз мумкин бўлади. Шунини айтиш керакки, ҳар қандай тасвир ҳам нишон яратиш учун тўғри келавермайди. Контраст деталлари кўп бўлган тасвирлар яхши нишон бўла

олади. Нишонларни кейинчалик таниб олиш учун таянч матрица айнан ана шу деталлардан курилади.

*Simple 3D targets (CubeandCuboid)* – бу тўғри бурчакли параллелепипед кўринишидаги (кубни ҳам ўз ичига олган ҳолда) нишонлардир. Масалан, куруқ нонуштадан бўшаган идишлар, гугурт кутиси ёки эндигина сотиб олинган стол ўйини шундай нишон бўлиб хизмат қилиши мумкин. Ҳар қандай қути каби бу нишон ҳам олтита текисликдан таркиб топади ва уни яратиш учун бизга уларнинг ҳар бири учун олтита тасвир керак бўлади.

*Cylindertargets* – нишонларнинг бу тури, номига қарамасдан, асосларнинг диаметрларини бериш имкониятига эга бўлган қирқилган конусни ўзида тақдим қилади. Албатта, агар бир хил диаметрлар танланадиган бўлса, у ҳолда цилиндр олинади, бироқ бу хусусий ҳолат бўлиб ҳисобланади. Бундай нишонни яратиш учун бизга фақатгина асосларнинг диаметрлари ва баландликни кўрсатиш эмас, балки яна учта тасвирни қўшиш – иккита асоснинг ҳар бирига биттадан ва ён юза учун яна биттани қўшиш керак бўлади.

*Framemarkers* – бу махсус тайёрланган рамка кўринишидаги нишон бўлиб, кўпроқ штрих-кодга ўхшаб кетади. Бундай рамкага ҳар қандай тасвирни жойлаштириш мумкин. Нишонларнинг бу тури тасвир етарлича деталлаштирилмаган ва ундан бирор-бир тушунарли `imagetarget` яратилмаган ҳолларда қўл келади.

*Text (wordtargets)*. Vuforia кутубхонасига яна матнларни таниб олиш ҳам киритилган, шу сабабли ҳар қандай сўз ёки уларнинг бирикмаси нишон бўлиб ҳисобланиши мумкин. Ҳозирги кунда фақатгина лотин тили ушлаб турилади, лекин хорижий сўзларни ўрганиш ҳам мумкин.

Илова яратиш учун зарур бўладиган нишонларнинг сонига боғлиқ равишда уларни ёки `DeviceDatabase` да сақлаш, уларга ҳоҳлаган пайтда кириш ва уларни бевосита қурилманинг ўзида таниб олиш ёки бу

юкламанинг бир қисмини CloudDatabases га– нишонларни сақлаш ва курилмадан юбориладиган маълумотлар асосида уларни аниқлаш учун мўлжалланган VuforiaWebServices жамланмасидан таркиб топган сервисга ўтказиш мумкин. Ҳар иккала ёндашув ҳам ўзининг плюслари ва минусларига эга, бироқ илованинг мақсадлари аниқ бўлгандан кейин танлаш унчалик қийин бўлмайди [27].

Булардан ташқари нишонлар камерадан олинган картинкадан фойдаланиш билан бевосита илованинг ўзидан ҳам яратилиши мумкин.

Шуни қайд қилиш лозимки, QCAR нишоннинг тўлиғича кадрга тушишини талаб қилмайди. Уни таниб олиш ва ундан фойдаланиш учун унинг қисмлари етарли бўлади. Бу жуда муҳимдир ва илованинг жилосини турли бурчаклар остида кўриб чиқиш имконини беради. Augmentedreality дан фойдаланишнинг барча мисоллари ва улардаги барча нарсалар Vuforia билан фойдаланилиши мумкин:

*Статик ва динамик 3D моделлар.* Биз ҳаммамиз учун севимли бўлган чойнакдан тортиб то мураккаб анимациягача бўлган ҳар қандай 3D объектни қўшишимиз мумкин. Vuforia–Unity билан интеграциялаш имкониятини фаол тарзда қўллаб қувватлайди, бу бизнинг имкониятларимизни янада оширади. Бунда бизни фақатгина фантазия ва смартфонлар/планшетларнинг қуввати чегаралаб туради.

*Виртуал тугмачалар* – улар билан ўзаро ҳаракатларга киришишда реал предметлар билан иш кўришга тўғри келади. Бу нишондаги тугмача бўлиши мумкин, уни босиш, масалан, иш столидан ер-ҳаво ракетасини учиради, уни фақатгина смартфон экрани орқали кўриш мумкин.

*Видео ватасвирлар*, улар бизнинг нишонимизни алмаштириш ёки уни янада жонлироқ қилиш имконини беради.

*Фонни ўзгартириш*, ҳаммасидан ташқари, камерадан олинган тасвирнинг ўзини ҳам ўзгартириш мумкин. Бошқача қилиб айтганда биз

фақат картинкани тўлдириш билан чекланмасдан, унинг ўзи билан ҳам операциялар бажаришимиз мумкин.

### **Иккинчи боб бўйича хулосалар**

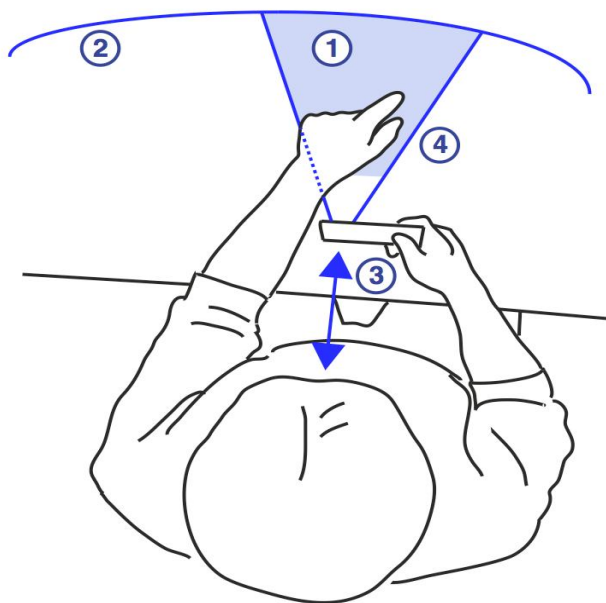
Иккинчи боб бўйича асосий натижалар қуйидагилардан иборат:

- Виртуал воқеликни моделлаштириш тилининг ўзига хос томонлари ўрганилди.
- Виртуал воқеликни яратиш учун O3D платформаси таҳлил қилинди.
- Дастурий таъминот ишлаб чиқиш учун керакли турли фреймворклар қиёсий таққосланди.
- Аддитив воқелик яратиш учун фреймворк аниқлаштирилди.

### III БОБ. ДАСТУРИЙ Истурий таъминотнинг тасвирланиши

#### 1. Виртуал объектлар билан ўзаро ҳаракатлар

Мобил AR ларда ишораларни таниб олишнинг умумий имкониятини баҳолаш учун сенсор экрани орқали стандарт ўзаро ҳаракатлар ва мобил қурилманинг қандай ҳаракатланиши унга боғлиқ бўладиган ўзаро ҳаракатлар типини солиштирамиз. Мақсад виртуал 3D объектнинг силжиш, масштаблаш ва айланиш каби асосий аффин ўзгартиришлари учун махсус ишоралардан фойдаланиш имкониятини тадқиқ қилиш ҳисобланади. Бу концепциянинг қулайлиги бир нечта чекловчи омиллар (факторлар) туфайли танқидий бўлиши мумкин [28].



Расм-10. Мобил аддитив воқеликдан фойдаланишда фаол фазовий соҳа

Биринчидан, асбоб-ускуналарга қўйиладиган чеклашлар мавжуд. Камеранинг кўриш майдони (FOV) қўл ёки бармоқни кузатиб бориш мумкин бўладиган соҳани белгилайди. Масалан, телефоннинг камераси мос равишда тахминан  $55^\circ$  ва  $42^\circ$  горизонтал ва вертикал кўриш бурчагига эга. Камераларнинг имконияти ҳам ўз таъсирини ўтказди. Қуйи

имкониятишораларни, айниқса бармоқ камерага яқин бўлганда, аниқ кузатиб боришни амалга ошириш имконини бермайди.

Иккинчидан, одамнинг биомеханикаси билан боғлиқ чеклашлар мавжуд. Одамнинг қўли қўл етадиган иш соҳасига эга бўлади, у барча нуқталарга тирсакда танланган таянч нуқтасидан туриб эришиш мумкин бўладиган ҳажм сифатида таърифланиши мумкин. Бу чеклаш бошқа характеристикалар туфайли ҳам вужудга келади, масалан, одам қўлининг узунлиги: у эркакларда ўртача 0,5968 м ни ва аёлларда ўртача 0,5541 м ни ташкил қилади.

Бундан ташқари, телефон кўздан маълум бир масофада жойлашиши лозим. Энг яхши кўришнинг энг кичик масофаси (Least Distance of Distinctive Vision, LDDV) нормал кўриш қобилиятига эга бўлган одам бирор-бир нарсага қийналмасдан қараши мумкин бўлган минимал масофани характерлайди, бу масофа одатда 25см бўлади. Барча муаммоларни эътиборга олган ҳолда, биз мобил AR илова билан ўзаро ҳаракатларга киришиш учун фойдаланилиши мумкин бўлган етарлича чекланган соҳага эга бўламиз, бу 10-расмда кўрсатилган. Шу нарса аниқланганки, тасвирдаги (1) соҳанинг ўлчами 15 см дан 25 см гача бўлади [29].

*Ўзаро ҳаракатларнинг асосий концепциялари.* Ўзаро ҳаракатларнинг кўйилган масалани ҳал қилиши мумкин бўлган бир нечта турларини кўриб чиқамиз. Содда бўлиши учун қуршаб турувчи физикавий муҳит билан бевосита боғланмаган виртуал объектнинг ўзаро ҳаракатларини кўриб чиқамиз.

#### 1) Сенсор экрани ёрдамида ўзаро ҳаракатлар

Стандарт сенсор экрани асосида ўзаро ҳаракатларни кўриб чиқиш учун учта масала қуйидаги тарзда кўриб чиқилган: нишонни танлаш —у бўйича сенсор экранда оддий чақириш билан бажарилади. Бундай танлаш контекстли меню деб аталади. Менюнинг пунктларидан бири объектни



“силжиш режими” га ўтказди, унда фойдаланувчи экран бўйлаб судраш билан объектни силжитиши мумкин. Агар қурилма алоҳида силжиса ва объект силжимаса, у ҳолда у реал оламга нисбатан ўз позициясида қолади. Трансляциялаш режимининг тугалланиши мос келувчи белгига босиш билан бажарилади ва натижада объект реал оламга нисбатан охирги ҳолатида қотириб қўйилади.

Фойдаланишнинг қулайлиги нуктаи-назаридан ушбу ёндошув оддий ва интуитив жиҳатдан тушунарлидир, чунки у сенсор экрани билан ўзаро ҳаракатларга мос келади. Менюни амалга ошириш ҳам ишончли ва аниқ бўлиб ҳисобланади, чунки интерфейсининг дизайни тўлиқ назорат қилинади. Виртуал объектни танлашдаги аниқлик – айниқса, у етарлича кичик бўлса, бошқа объектга жуда яқин жойлашган бўлса ёки бир объект бошқасини қоплаб турса – ўзаро ҳаракатларнинг мазкур типининг заиф томонлари бўлиб ҳисобланади. Фойдаланувчи қурилмани вертикал ҳолатда ушлаб туришига ва уни реал оламдаги маълум бир нуқтага йўналтиришига тўғри келадиган ҳолларда фойдаланувчининг сенсор экрани билан ўзаро ҳаракатларга киришишида муаммолар вужудга келиши мумкин. Бу, айниқса, объект бошланғич тасвирда кўринмайдиган позицияга, масалан фойдаланувчининг орқасида жойлашган позицияга силжитилиши лозим бўлган вазиятларда қийин бўлади [30].

## 2) Қурилмага асосланган концепция

Ўзаро ҳаракатларнинг иккинчи тури қурилманинг манипуляциялар учун жойлашиши ва йўналишидан фойдаланади (киритилган акселерометр ва компасдан олинган маълумотлар ёрдамида аниқланади). Бундай ҳолда махсус “мўлжал” экран марказида акс эттирилади ва объектни танлаш учун фойдаланилади. Уни объектда маълум вақт давомида ушлаб туриш билан объектни танлаш содир бўлади ва меню чақирилади. Меню шундай тарзда ишлайдики, “мўлжал” ни ёзувлардан бирига силжитади ва унчалик катта бўлмаган вақт оралиғида уни қимирлатмасдан ушлаб туради.

Силжиш режимида объект экран марказида “мўлжал” остига қотириб қўйилади, қурилманинг ўзи эса бу пайтда айлана бўйлаб силжийди. У сенсор экранида чиқиллатиш билан керакли жойга жойлаштирилиши мумкин. Бу ҳаракат шунингдек силжиш режимини тугаллайди ва тизимни нормал ўзаро ҳаракатларга қайтаради.

Объектлар ва меню элементларини танлашда қурилмага асосланган ўзаро ҳаракатлар сенсор экранни ёрдамидаги ўзаро ҳаракатларга қараганда кўпроқ вақтни олади, чунки фойдаланувчи уларни тўғридан-тўғри танлай олмайди ва у индикатор тўлгунча кутиб туришга мажбур бўлади. Аниқлик нуқтаи-назаридан бу концепция объектларни аниқроқ танлаш имконини беради, чунки “мўлжал” бармоққа нисбатан етарлича кичик бўлган нишонларни янада аниқроқ кўрсатиши мумкин. Шунга қарамасдан, қурилмани узоқ вақт мобайнида битта ҳолатда ушлаб туриш муаммо туғдириши мумкин. Бундай ёндошувда объектлар билан ишлаш интуитив ва осон бўлиб туюлиши мумкин, чунки фойдаланувчи қурилмани шунчаки силжитиши зарур бўлади. Шунга қарамасдан сенсор экранига босиш билан объектни мақсадли позицияга силжитиш фойдаланувчининг бармоғи текканда қурилманинг силкиниши мумкинлиги туфайли баъзи бир ноаниқликларга олиб келиши мумкин.

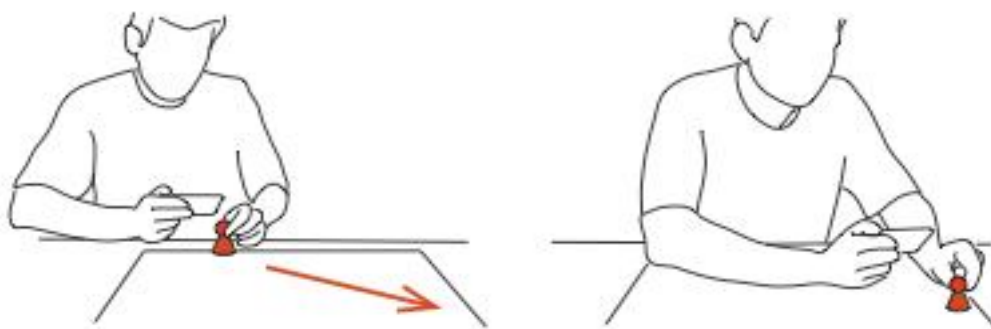
### 3) Ишораларга асосланган концепция

Сенсор экранни асосидаги ўзаро ҳаракатлар интуитив бўлиб туюлиши мумкин, чунки у замонавий мобил қурилмалар билан мунтазам ўзаро ҳаракатларга мос келади. Бироқ бу фақатгина уч ўлчовли аддитив воқеликни сенсор экранида 2D проекция ёрдамида масофадан туриб бошқариш имконини беради. Агар фойдаланувчининг қўли қурилма олдида бўлганда (яъни у реал вақтда экранда пайдо бўлганда) унинг ишораларини кузата бошласак, у ҳолда бармоқнинг учидан объектлар билан тўғридан-тўғри ўзаро ҳаракатларга киришиш учун, яъни уларни танлаш ва манипуляциялаш учун фойдаланилиши мумкин. Идеал ҳолатда

бармоқни барча учта идеал ҳолатда кузатиб бориш ва шундан келиб чиққан ҳолда 3D да объектлар билан тўлиқ манипуляциялаш имкониятини таъминлаш мумкин. Тасвир шовқинининг трекинг алгоритмига кучли таъсир кўрсатишидан қочиш учун ишончли кузатиб бориш маркеридан фойдаланиш мумкин. Обьектни танлаш унинг реал оламдаги ҳолатига бармоқни йўналтириш билан амалга оширилади. Силжиш пайтида объектни “орқасидан итариш” мумкин.

4) Қуршаб турувчи физик муҳит билан интеграллашган виртуал объектлар билан ўзаро ҳаракатлар

Аддитив воқелик тизимлари учун ишоралардан фойдаланиш потенциални тадқиқ қилишни давом эттириш учун реал қуршаб турувчи физик муҳит билан боғланган виртуал объектлар билан манипуляцияларни кўриб чиқамиз. Бош вазифа – бу табиий ва қўйилган мақсадларга эришиш учун фойдали бўлиб ҳисобланадиган, мумкин бўлган ўзаро ҳаракатларни аниқлашдан иборат бўлади.



Расм-11. Виртуал объектнинг силжиши

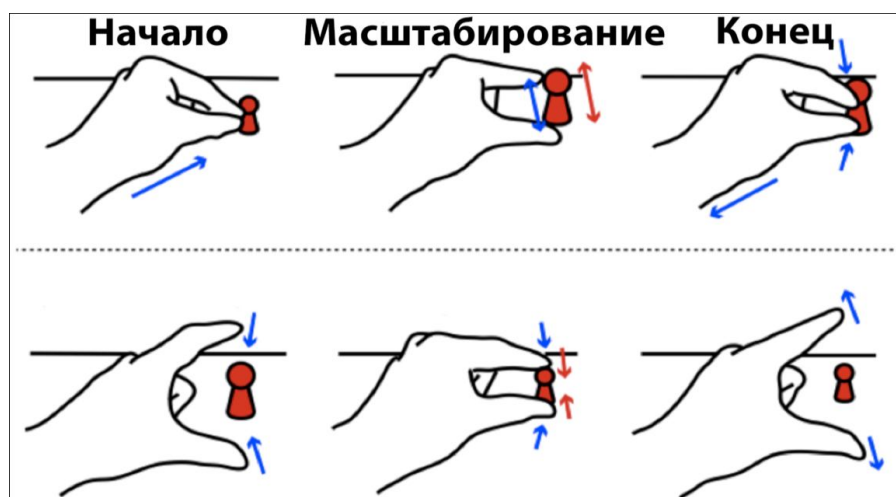
### **Ўзаро ҳаракатларнинг турлари**

Силжитиш. Одамлар реал ҳаётда объектларни силжитганда ўзаро ҳаракатларнинг иккита типини кузатиш мумкин: ёки объектни керакли томонга қаратиб силжитадилар, ёки иккита бармоқ билан олиб зарур бўлган жойга жойлаштирадилар. Силжитишда, бармоқ унинг конкрет томонларидан бирига текканда, объект силжийди. Шундан келиб чиққан

ҳолда, бундай ўзаро ҳаракатлар учун битта бармоқдан фойдаланилади. Иккинчи вариант – ушлаб олиш – иккита бармоқ билан амалга оширилади. Таниб олишни яхшилаш учун бармоқларга махсус маркер ёпиштирилади. Ҳар иккала маркер объект чегараларига етарлича яқинлашиши билан виртуал объект “қўлга тушади”. Энди у ҳар қандай жойга силжитилиши ва бармоқларни бир биридан узоқлаштириш билан ўша жойга жойлаштирилиши мумкин.

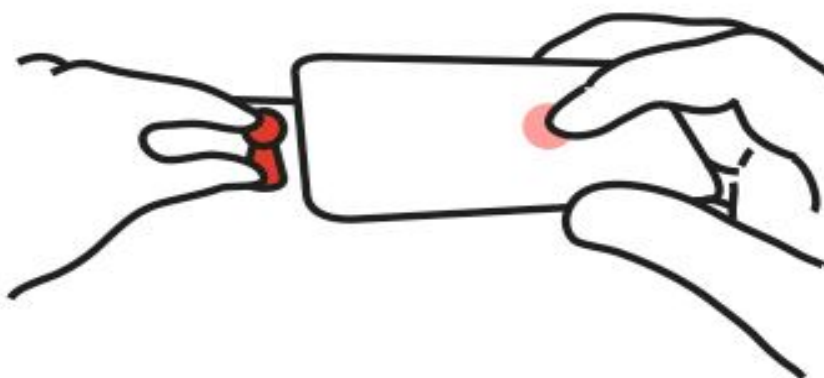
Масштабланиш. Масштабланиш – бу фақатгина виртуал объектлар билан қилиниши мумкин бўлган ўзаро ҳаракатдир. Шундан келиб чиққан ҳолда ҳеч қандай табиий эквивалент мавжуд эмас. Шунга қарамадан, масштабланиш ишораси *pinch-to-zoom* ишора туфайли сенсор экранларида етарлича кенг тарқалган бўлиб ҳисобланади: иккита бармоқнинг марказдан ёки бир-бирига яқинлашувчи ҳаракати мос равишда ўсишга ёки камайишга олиб келади. Шу билан биргаликда бу AR да виртуал объектларни масштабланиш учун табиий ишорага ўхшаб туюлади, бироқ бунда сенсор экранидан фарқли равишда битта катта муаммо бор, биз ишора қачон масштабланиш ишораси бўлиб ҳисобланишини, қачон эса ҳисобланмаслигини автоматик тарзда аниқлай олмаймиз. Бу муаммони ҳал қилиш учун қуйидагича ечимлар бўлиши мумкин.

Улардан биринчисида ишораларнинг иккита типи фарқланади: объектга иккита бирлаштирилган бармоқлар билан тегиниш, сўнгра эса улар ўртасидаги масофани катталаштириш, бу объектнинг катталашувига олиб келади. Объектни икки томондан ушлаб олиш ва иккита бармоқ орасидаги масофани камайтириш объектнинг кичрайишига олиб келади. Ҳар иккала ҳолатда ҳам операциянинг тугалланиши бармоқни объектнинг йўналишидан қарама-қарши томонга силжитиш билан амалга оширилади. Бунда кичрайиш ва катталашуви – иккита алоҳида ўзаро ҳаракатлардир.



Расм-12. Масштаблaш

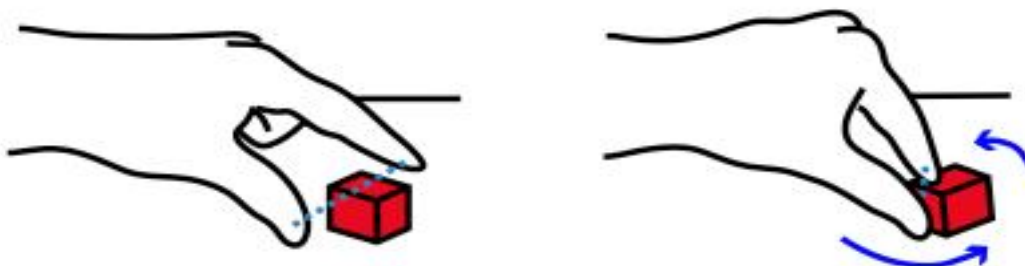
Муаммони ҳал қилишнинг иккинчи варианты иккита бармоқ орасидаги масофани каттартириш ёки кичрайтириш билан объектнинг ўлчамини узлуксиз ўзгартириш имконини беради. Бунинг учун фойдаланувчи масштаблaшнинг бошланиш momentiда сенсор экранининг ҳар қандай жойини босиши лозим бўлади. Бу ўзаро ҳаракат амалда масштаблaш учун кўпроқ қайишқоқ ва табиий бўлиши билан бир қаторда у фойдаланувчи учун қулай бўладими-йўқми – буниси аниқ эмас.



Рас- 13. Масштаблaш учун иккита интерфейсни бирлаштириш

Айлантириш. Объектни айлантириш одатда объектни иккита бармоқ билан ушлаб олиш ва уни ўқ атрофида буриш билан амалга оширилади. Бу – олдин объектни ушлаб олиш (худди силжитишдаги каби), сўнгра эса уни

иккита маркерни бирлаштириб турадиган бошланғич ва охирги чизиклар ўртасидаги бурчакнинг ўзгаришига боғлиқ равишда айлантириш йўли билан амалга оширилиши мумкин.



Расм-14. Виртуал объектни айлантириш

Айлантириш ишорасининг иккинчи версиясида, аввал, объектга қўл теккизиш билан, объектни танлаш амалга оширилади, сўнгра эса кўрсаткич бармоқнинг айланма ҳаракати ёрдамида объектни буриш амалга оширилади. Сенсор экранинингхтиёрий жойга босишдан айлантиришни тўхтатиш ва объектни маълум бир йўналишда қолдириш учун фойдаланилади.

## 2. Дастурий таъминотнинг архитектураси

Кўриб чиқиладиган тизимнинг фаолият кўрсатиши учун қуйидаги модуллар зарур бўлади:

1) Камера таркибий қисми кадрларни ҳар бир кўриш қайд қилинишини ва самарали тарзда трекерга узатилишини кафолатлайди. Ишлаб чиқувчи ёзиб олишни бошлаш ва тўхтатиш учун фақатгина камерани ишга туширади. Камеранинг кадри аппаратли-боғлиқ бўлган форматга автоматик тарзда конвертацияланади ва тасвирнинг керакли ўлчамини беради.

2) Тасвир конвертери. Форматлар конвертери кадрни камера форматидан (масалан, YUV12) OpenGL ES рендеринглаш (масалан, RGB565) ва кузатиб бориш учун яроқли бўлган форматга ўзгартиради. Бу ўзгартириш камерада тасвирни ўзгартирилган кадрлар тўпламида турли хил ўлчамларда кичрайтиришни ҳам ўз ичига олади.

3) Трекер. Трекернинг таркибий қисми реал олам объектларини видеокамера доирасида аниқлаш ва кузатиб бориш учун компьютернинг кўриш алгоритмларини ўз ичига олади. Камерада олинган тасвир асосида турли хил алгоритмлар янги мўлжалларни, ёки маркерлар ва виртуал тугмачаларни аниқлайди. Натижалар объект ҳолатида сақланади, ундан фонли видеони визуаллаштиришда фойдаланилади ва уларга илова кодидан кириш мумкин бўлади. Трекер бир пайтнинг ўзида бир нечта маълумотлар жамланмаларини юклаши ва уларни фаоллаштириши мумкин.

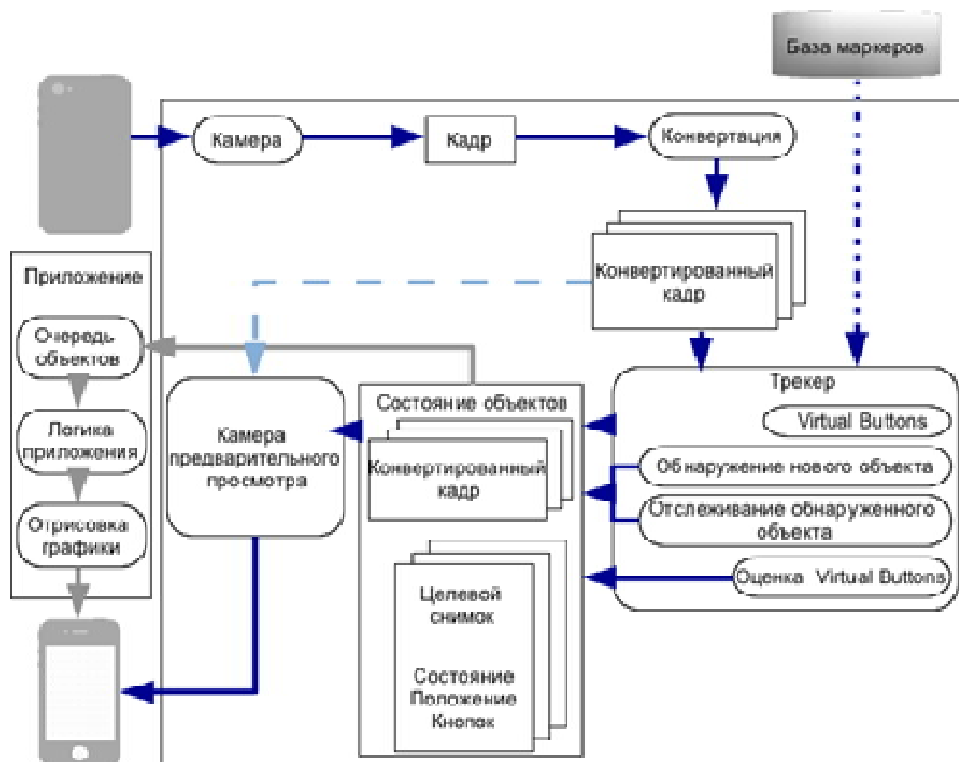
4) Фонни тасвирлаш. Бу визуаллаштириш модули объектда сақланаётган тасвирни яратади. Видеони фон режимида визуаллаштириш унумдорлиги конкрет қурилма учун оптималлаштирилган.

5) Илова коди. Кодда юқорида санаб ўтилган барча таркибий қисмлар киритилган бўлиши ва унга шарт бажарилган бўлиши лозим. Ҳар бир ишлов берилган кадр учун объект янгиланади ва чизиш услуги чақирилади. Бунинг учун қуйидагилар зарур бўлади:

- янгидан аниқланган мўлжаллар, маркерлар ёки бу элементларнинг янгиланган ҳолатлари учун объектларни аниқлаш;
- илованинг логикасини янги кириш маълумотлари билан янгилаш;
- аддитив воқелик қатламини чизиш.

6) Маркерлар базаси. Қурилмада маркерларнинг маълумотлар базаси онлайн Target Manager ёрдамида яратилади. У конфигурациянинг махсус XML файлини ва кузатиб бориш учун маълумотлар базасини ўз ичига оладиган иккилик файлини яратади. Бу файллар иловани ишлаб

чиқувчи томонидан иловани ўрнатиш пакетига қўшилади ва улардан бажариш пайтида фойдаланилади. 15-расмда прототип архитектураси келтирилган.



Расм—15. Тизим прототипининг архитектураси

### 3. Фойдаланувчига йўриқнома

Диссертация ишида аддитив воқеликни визуаллаштириш учун иккита дастурий илова яратилган [31].

Java дастурлаш технологияси Java дастурлаш тилига асосланган бўлиб, Java намунавий дастурлаш тили сифатида қабул қилинади. Java дастурлаш тили базасида Java иловалар, JavaScript функциялари, Java апплетлар, Java Beans, маълумотлар базаси билан ишлаш (JDBS), GUI, сервлетлар, JSP, интерактив динамик Web саҳифалар ва мижоз сервер иловалар яратилиши мумкин.



Java содда, етарли даражада турлаштирилган дастурлаш тили бўлиб, унинг ёрдамида объектга йўналтирилган конструкцияларни қийинчиликсиз ва мамнуният билан яратиш мумкин. Бундан ташқари Java интернет билан ҳамкор бўлиб, кўплаб Web лойиҳалар унинг улишига тўғри келади. Java апплетлар тўғрисида кўпчилик эшитган бўлсада, Java ёрдамида тўла қонли иловалар, компоненталар ёки сервлетлар яратилиши мумкинлигини кўпчилик билишмайди. Ҳозирги замон тараққиёт босқичида Internet Banking ва SMS Banking каби қўл телефонлари базасидаги банк хизмат турлари кенг оммалашмоқда.

Бундан ташқари қўл телефонлар эгалари зерикиб қолмасликлари учун кўплаб ҳар хил кунгил очар ўйинлар яратилган.

Бу ишларни қўл телефонлари учун яратилган дастур таъминоти амалга оширади. Бундай дастурий таъминотининг каттагина қисми Java дастурлаш воситалари ёрдамида яратилмоқда.

Web серверларда CGI сценарийларнинг самарали ворислари сифатида сервлетлар ишлатилмоқда. Сервлетларнинг кенгайтирилгани Java сервер саҳифалари (JSP) ҳисобланади. Сервлетлар ва JSP ларни ишлатишдан мақсад динамик Web саҳифаларни яратиш ва бошқаришдан иборат. JSP технология битта ҳужжатда ҳам Java кодларни ҳам HTML кодларни аралаш ҳолда ишлатиш имконини беради.

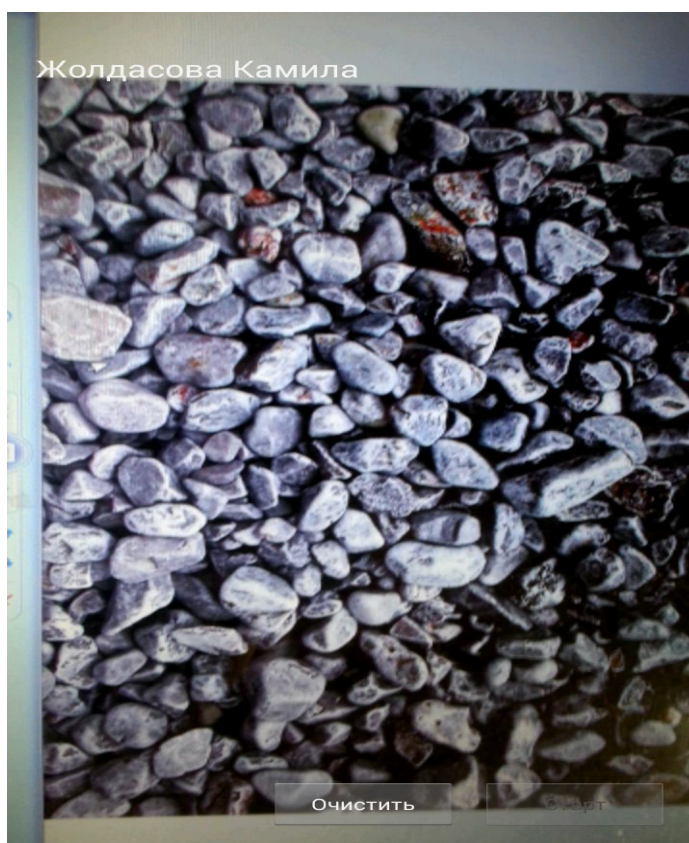
JavaScript объектга йўналтирилган дастурлаш тили бўлиб кучсиз типлаштирилган. У жуда қисқа вақт ичида Web сценарийлар яратишнинг асосий тили бўлиб қолди. JavaScript бевосита HTML ҳужжат ичида ёзилгани учун Web саҳифанинг функционал имкониятларини кескин оширади. JavaScript ни ишлатиш орқали дастурчи статик Web саҳифага ҳар хил динамик эффектлар бериши, мижоз сервер иловаларнинг мижоз қисмини яратиши, киритилаётган маълумотларнинг тўғрилигини киритилаётган жойнинг ўзида серверга жўнатмасдан текшириши ва бошқа ишларни амалга ошириши мумкин.

Юқорида келтирилганлар Java дастурлаш тилига жиддий эътибор бериш ва уни ўрганишга астойдил киришиш зарурлигини асослайди.

Vuforia – C++, Java, Objective-C, ва .Net тилларида иловаларни дастурлаштириш интерфейсларини Unity ўйин движоги билан интеграциялаш орқали тақдим қилади. Шу тариқа SDK iOS ва Android учун натив AR-иловаларни ишлаб чиқишни ушлаб туради ва шу билан бир вақтнинг ўзида Unity да ишлаб чиқишни ҳам кўзда тутди, унинг натижалари ҳар иккала платформага осон кўчирилиши мумкин.

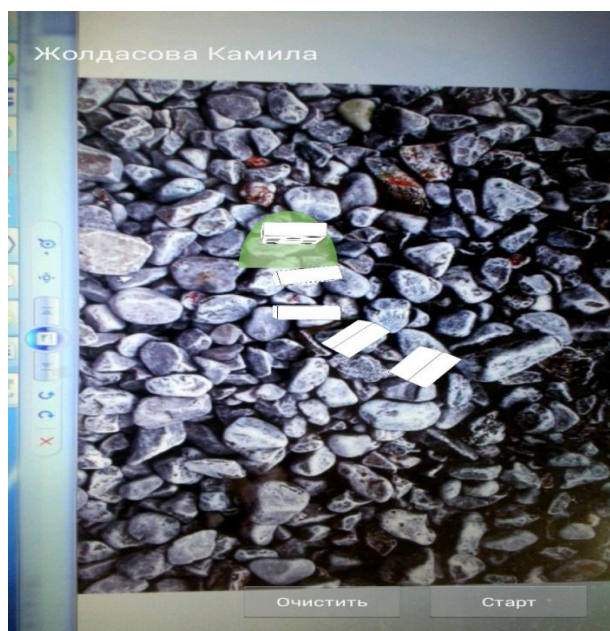
Шу дастурлар асосида икки хил иловалар яратилган.

Биринчиси –«Домино» иловаси.«Домино» иловаси аддитив воқеликни визуаллаштириш учун амалга оширилган. Дастурни ишга туширишда телефон камерасининг шарҳини чиқарадиган ойна чиқади.



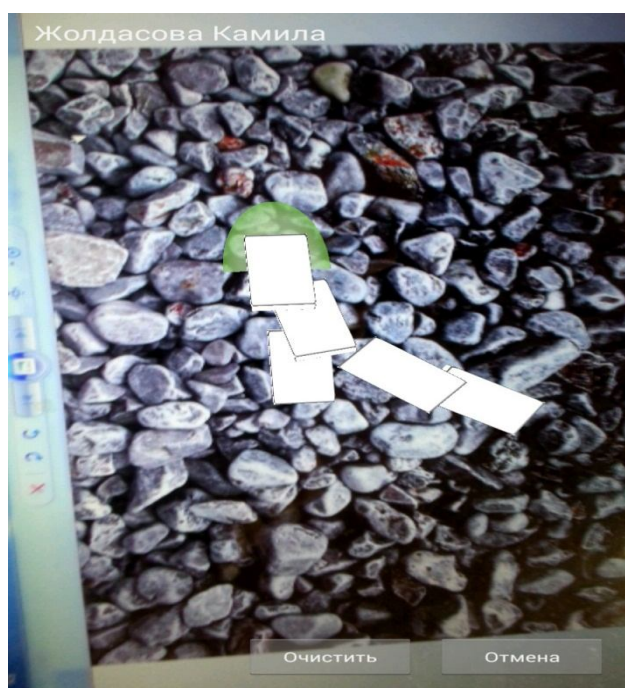
Расм-16. «Домино» иловасининг бош ойнаси

Маркерга телефон камерасини йўналтиришда биз экранга бармоқларни тегизиш билан объектларни (доминони) қўшишимиз мумкин.



Расм-17. «Домино» объектларини қўшиш

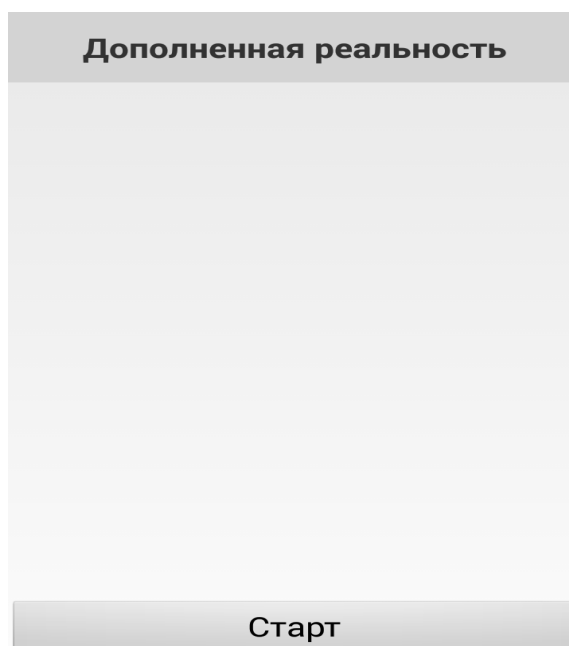
«Старт» тугмачасини босишда объектлар бир-бирининг устига қулайди. (Домино принципи). «Тозалаш» тугмачаси экранни объектлардан тозалаш учун хизмат қилади.



Расм-18. Объектларнинг қулаши

Иккинчи дастурий илова – бу “Чайник” иловасидир. “Чайник” иловаси аддитив воқеликни визуаллаштириш учун хизмат қилади. Бу

иловада маркерларнинг 3 та туридан фойдаланиш мумкин (chips, stones, tarmac).



Расм-19. “Чайник” иловасининг бош ойнаси

«Старт» тугмачасини босганимизда илованинг бош ойнаси очилади. Бунда маркернинг уч хили пайдо бўлади.







Расм-20. “Чайник” обьектини учта маркерда рендеринглаш

Бундан сўнг телефон орқали “Чайник” иловасини ренгдерингланади ва шу билан аддитив воқелик амалга оширилади.

“Чайник” ва “Домино” аддитив воқелик мобил иловалари яратилиб, дастурий таъминот архитектураси қурилди ва дастурий таъминотдан фойдаланиш бўйича кўрсатмалар ишлаб чиқилди.

### **Учинчи боб бўйича хулосалар**

Учинчи бобдаги асосий натижалар қуйидагилардан иборат:

- Виртуал объектларнинг ўзаро таъсири тадқиқ қилинди.
- Аддитив воқелик учун мобил дастурий иловалар яратилди.
- Дастурий таъминотдан фойдаланиш учун кўрсатмалар ишлаб чиқилди.

## ХУЛОСА

Магистрлик ишини бажариш вақтида аддитив воқелик технологиялари кўриб чиқилди, фойдаланувчининг виртуал объектлар билан ўзаро ҳаракатлари таҳлил қилинди. Таҳлил қилиш натижасида уларнинг афзалликлари ва камчиликлари кўрсатилди.

Аддитив воқелик — бу дастлаб бир — бири билан боғлиқ бўлмаган соҳаларни: инсон атрофидаги реал объектлар ва виртуал дунёни экранда бирлаштиришдир. Бу интерактив технология фойдаланувчи учун компьютердаги махсус 2D ва 3D объектларни видеокамерадан туширилаётган тасвир устидан қўйиш имконини беради ва шу тариқа, воқелик тўлдирилади. Аддитив воқелик асосида оптик трекинг тизими ётади.

Тизим ишлаши учун қуйидаги компонентлар зарур:

- Меткалар — махсус тасвирлар, компьютер моделлари учун визуал идентификаторлар;
- Камера— меткаларни ҳақиқий дунёда «кўради» ва компьютерга видеосигналларни юборади;
- Дастурий таъминот — қабул қилинган сигналларни қайта ишлайди ва виртуал моделларни ҳақиқий объектларнинг тасвири билан бирлаштиради.

Шунингдек ишда виртуал воқеликни моделлаштириш тилининг хусусиятлари тадқиқ қилинди, виртуал воқеликни яратиш учун O3D-платформаси таҳлил қилинди, дастурий таъминотни яратиш учун турли фреймворклар таҳлил қилинди. Таҳлил натижасида аддитив воқеликни яратиш учун фреймворк танлаб олинди.

Vuforia — фойдаланиш текин бўлган кутубхона, у доимо такомиллашиб боради, натив иловалар билан ишлаш имконини ҳам, Unity махсус ўйин движоги ёрдамида кроссплатформали иловаларни яратиш имконини ҳам беради.

Тадқиқот ишида виртуал объектлар билан ишлаш ўрганилди, ҳаракатларни реал вақтда танийдиган аддитив воқелик системасининг прототипи ишлаб чиқилди. “Чайник” ва “Домино” аддитив воқелик мобил иловалари яратилди, дастурий таъминот архитектураси қурилди ва дастурий таъминотдан фойдаланиш бўйича кўрсатмалар берилди.



## **Фойдаланилган адабиётлар рўйхати:**

### **Норматив-ҳуқуқий ҳужжатлар:**

1. Ўзбекистон Республикаси Президенти асари «Ўзбекистон XXI аср бўсағасида. Хавфсизликка таҳдид, барқарорлик шартлари ва тарраққиёт кафолатлари» И.А. Каримов 1997 й.
2. Ўзбекистон Республикаси Президенти асари «Юксак маънавият — енгилмас куч» И.А. Каримов 2008 й.
3. Ўзбекистон Республикаси Президенти асари «Ўзбекистон мустақилликка эришиш оstonасида» И.А. Каримов 2011 й.
4. Ўзбекистон Республикаси Президенти қарори «Замонавий ахборот — коммуникация технологияларини янада жорий этиш ва ривожлантириш чора — тадбирлари тўғрисида», Тошкент ш., 21 март 2012 й.
5. Ўзбекистон Республикаси Президенти қонуни “Миллий ахборот — коммуникация тизимини янада ривожлантириш чора — тадбирлари тўғрисида”, Тошкент ш., 27 июн 2013й.
6. Ўзбекистон Республикаси Президенти қонуни “Ахборот — коммуникация технологиялари соҳасида кадрлар тайёрлаш тизимини янада такомиллаштириш чора — тадбирлари тўғрисида”, Тошкент ш. 26 март 2013й.

### **Ўқув қўлланма ва адабиётлар:**

7. Авербух В. Л. К теории компьютерной визуализации //Вычислительные технологии Т. 10, N 4, 2005 , стр 21-51.
8. Averbukh V. L. The Specialized Systems of Scientific Visualization // AIP Conference Proceedings (ZABABAKHIN SCIENTIFIC TALKS - 2005: International Conference on High Energy Density Physics). August 3, 2006. Volume 849, pp. 481-486.
9. Авербух В. Л., Байдалин А. Ю., Васёв П. А., Горбашевский Д. Ю.,

- Исмагилов Д. Р., Казанцев А. Ю., Манаков Д. В., Шинкевич А. Н. Проблемы разработки средств визуализации для супервычислений // Параллельные вычислительные технологии / Труды научной конференции. Челябинск. Издательство ЮУрГУ, 2007. Том 2.
10. Манаков Д. В. Анализ параллельных визуальных технологий // Вычислительные технологии. Том 12, N 1, 2007, Стр. 45-60.
  11. AverbukhV., BakhterevM., BaydalinA., IsmagilovD., TrushenkovaP. Interfaceand Visualization Metaphors // J. Jacko (Ed.): Human-Computer Interaction, PartII, HCI 2007, LNCS 4551, Springer-Verlag Berlin Heidelberg 2007, pp. 13-22.
  12. AverbukhV. L., BakhterevM. O., BaydalinA. Yu., GorbachevskiyD. Yu., IsmagilovD. R., KazantsevA. Yu., NebogatikovaP. V., PopovaA. V., VasevP. A. SearchingandAnalysisofInterfaceandVisualizationMetaphors // Human-Computer Interaction, NewDevelopments. / EditedbyKikuoAsai. Chapter 3, Vienna, In-teh. ISBN 978-953-7619-14-5, pp. 49-84.
  13. Howard Rheingold Virtual Reality: The Revolutionary Technology of Computer-Generated Artificial Worlds - and How It Promises to Transform Society Simon & Schuster (August 15, 2012) ISBN 0671778978 416 pages
  14. Introduction to Virtual Reality John Vince 2008, 150 pages
  15. Бабенко В.С. Виртуальная реальность: Толковый словарь терминов. –СПб.: ГУАП, 2006. – 86 с.
  16. Sutherland, I. E. "The Ultimate Display" // Proceedings of IFIP 65. – 1965. –vol 2. – pp. 506-508.

#### **Илмий журналдаги мақолалар:**

17. Joldasova K.J., Alisherov F.A.,Kabiljanova F.A.– “Study of user experience in virtual environments”. // “XXI аср – интеллектуал авлод асри” Тошкент

шаҳри ва вилояти ҳудудий илмий – амалий конференцияси. 16 – 17 май 2014й.

18. Кабилжанова Ф.А., Жолдасова К.Ж. Возможности применения виртуальной реальности. —// Техник ва ижтимоий-иқтисодий фанлар соҳаларининг муҳим масалалари. Республика Олий ўқув юртлараро илмий ишлар тўплами. Пқисм. Тошкент, 2014. 82-84б.
19. Кабилжанова Ф.А., Жолдасова К.Ж. Сравнительный анализ сред виртуальной реальности. —// Техник ва ижтимоий-иқтисодий фанлар соҳаларининг муҳим масалалари. Республика Олий ўқув юртлараро илмий ишлар тўплами. Пқисм. Тошкент, 2014. 84-86б.
20. Камилов М.М., Фазылов Ш.Х., Мирзаев Н.М. (Институт математики и информационных технологий АН РУз, Москва, 2010, Международный симпозиум «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ») Построение алгоритмов вычисления оценок в условиях коррелированности признаков.
21. Камилов М.М., Маматов Н.С., Худайбердиев М.Х. (Институт математики и информационных технологий АН РУз, Москва, 2010, Международный симпозиум «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ») Формирование информативных признаков объектов распознавания на основе однородного критерия фишеровского типа  $k$ -го порядка.
22. Авербух В. Л., Байдалин А. Ю., Бахтерев М. О., Васёв П. А., Зырянов А. В., Казанцев А. Ю., Манаков Д. В. К обоснованию проекта визуализационной компоненты виртуального испытательного стенда // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа 29 марта — 2 апреля). [Электронный ресурс] Челябинск. Издательский центр ЮУрГУ, 2010, с. 378-386.
23. Авербух В. Л., Васев П. А., Горбашевский Д. Ю., Казанцев А. Ю., Попова А. В., Щербинин А. А. Манипуляция объектами в системах компьютерной визуализации // Тезисы 10-го Международного семинара "Супервычисления и Математическое моделирование", РФЯЦ-ВНИИЭФ,

Саров, 2008, с. 17-18.

24. Авербух Н. В., Щербинин А. А. Феномены присутствия и вовлечённости в средах виртуальной реальности // Дружининские чтения: Материалы 9-й Всерос. науч.-практ. конф., г. Сочи, 6-8 мая. / Федеральное агентство по образованию РФ; Соч. гос. ун-т туризма и курорт. дела; Соц.-пед. ф-т; Студ. психолог. о-во СПФ СГУТИиКД; Под ред. И. Б. Шуванова, О. А. Михаленко, А. А. Никифоровой, С. С. Новиковой. - Сочи: СГУТиКД, 2010. Стр. 79-82.
25. Авербух Н. В., Щербинин А. А. Феномен «присутствия» в виртуальной реальности в контексте интеллектуальной деятельности человека // Седьмая международная научно-методическая конференция "Новые образовательные технологии в вузе, (НОТВ - 2010)" Сборник материалов, Часть 2, Екатеринбург, 2010. Стр. 155-157.
26. Авербух В. Л., Авербух Н. В., Щербинин А. А.. Исследование работы пользователей систем визуализации, построенных на базе сред виртуальной реальности // Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта - 3 апреля 2009 г.), УДК 004.75 Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта - 3 апреля 2009 г.). - Челябинск: Изд. ЮУрГУ, 2009. УДК 004.75, ISBN 978-5-696-03854-4. (Электронное издание). Стр. 338-342.
27. Sadowski W. Jr., Stanney K. Measuring and Managing Presence in Virtual Environments. In Handbook of Virtual Environments Technology: Design, implementation and applications. Edited by: Stanney KM. Mahwah, NJ, Lawrence Erlbaum Associates; 2002. Chapter 45. pp. 791-806.
28. Witmer B. G., Singer M. J. Measuring Presence in Virtual Environments: A Presence Questionnaire. Presence, Vol. 7, No. 3, June 1998, 225–240.

**Интернет сайтлар:**

29. <http://gov.uz> (Правительственный портал Республики Узбекистан)
30. <http://ru.wikipedia.org> (Википедия-интернет энциклопедия)
31. <http://google.com> (Поисковая система компании Google)

```
import java.lang.ref.WeakReference;

import android.content.Context;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.widget.ToggleButton;

public class GUIManager {

    // Custom views:
    private View overlayView;
    private ToggleButton startButton;
    private Button clearButton;
    private Button deleteButton;

    // A Handler for working with the GUI from other threads:
    static class MyActivityHandler extends Handler
    {
        private WeakReference<GUIManager> guiManager;
        private Context context;

        MyActivityHandler(GUIManager guim, Context c)
        {
            guiManager = new WeakReference<GUIManager>(guim);
            context = c;
        }
    }
}
```

```
public void handleMessage(Message msg)
{
    Button deleteButton = guiManager.get().deleteButton;
    ToggleButton startButton = guiManager.get().startButton;

    switch (msg.what)
    {
        case SHOW_DELETE_BUTTON:
            if (deleteButton != null)
            {
                deleteButton.setVisibility(View.VISIBLE);
            }
            break;

        case HIDE_DELETE_BUTTON:
            if (deleteButton != null)
            {
                deleteButton.setVisibility(View.GONE);
            }
            break;

        case ENABLE_START_BUTTON:
            if (startButton != null)
            {
                startButton.setEnabled(true);
            }
            break;

        case DISABLE_START_BUTTON:
```

```

if (startButton != null)
    {
startButton.setEnabled(false);
    }
break;

case TOGGLE_START_BUTTON:
if (startButton != null)
    {
startButton.setChecked(true);
    }
break;

case DISPLAY_INFO_TOAST:
    String text = (String) msg.obj;
int duration = Toast.LENGTH_LONG;
    Toast toast =
Toast.makeText(context, text, duration);
toast.show();
break;

default:
break;
    }
}

private MyActivityHandler mainActivityHandler;

// Flags for our Handler:
public static final int SHOW_DELETE_BUTTON = 0;

```



```

public static final int HIDE_DELETE_BUTTON = 1;
public static final int ENABLE_START_BUTTON = 2;
public static final int DISABLE_START_BUTTON = 3;
public static final int TOGGLE_START_BUTTON = 4;
public static final int DISPLAY_INFO_TOAST = 5;

```

```

// Native methods to handle button clicks:

```

```

public native void nativeStart();
public native void nativeClear();
public native void nativeReset();
public native void nativeDelete();

```

```

/** Initialize the GUIManager. */

```

```

public GUIManager(Context context)
{

```

```

    // Load our overlay view:

```

```

    // NOTE: This view will add content on top of the camera /

```

OpenGL view

```

    overlayView = View.inflate(context, R.layout.interface_overlay, null);

```

```

    // Create a Handler from the current thread:

```

```

    // This is the only thread that can make changes to the GUI,

```

```

    // so we require a handler for other threads to make changes

```

```

    mainActivityHandler = new MyActivityHandler(this, context);

```

```

}

```

```

/** Button clicks should call corresponding native functions. */

```

```

public void initButtons()

```

```

        {
        if (overlayView == null)
        return;

        startButton =
(ToggleButton)overlayView.findViewById(R.id.start_button);
        startButton.setOnClickListener(new View.OnClickListener()
        {
        public void onClick(View v)
            {
            if (((ToggleButton) v).isChecked())
                {
                nativeStart();
                }
            else
                {
                nativeReset();
                }
            }
        });

        clearButton = (Button) overlayView.findViewById(R.id.clear_button);
        clearButton.setOnClickListener(new View.OnClickListener()
        {
        public void onClick(View v)
            {
            nativeClear();
            startButton.setChecked(false);
            }
        });

```

```

deleteButton = (Button) overlayView.findViewById(R.id.delete_button);
deleteButton.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        nativeDelete();
    }
});
}

```

```

    /** Clear the button listeners. */
    public void deinitButtons()
    {
        if (overlayView == null)
            return;

        if (startButton != null)
        {
            startButton.setChecked(false);
            startButton.setOnClickListener(null);
            startButton = null;
        }

        if (clearButton != null)
        {
            clearButton.setOnClickListener(null);
            clearButton = null;
        }
    }

```

```
if (deleteButton != null)
```

```
{
```

```
deleteButton.setOnClickListener(null);
```

```
deleteButton = null;
```

```
}
```

```
}
```

```
/** Send a message to our gui thread handler. */
```

```
public void sendThreadSafeGUIMessage(Message message)
```

```
{
```

```
mainActivityHandler.sendMessage(message);
```

```
}
```

```
/** Getter for the overlay view. */
```

```
public View getOverlayView()
```

```
{
```

```
return overlayView;
```

```
}
```

```
}
```