**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Смартфон приложение за решаване на стереометрични задачи във виртуална реалност

Дипломант: Научен ръководител:

*Владимир Владинов Александър Ангелов*

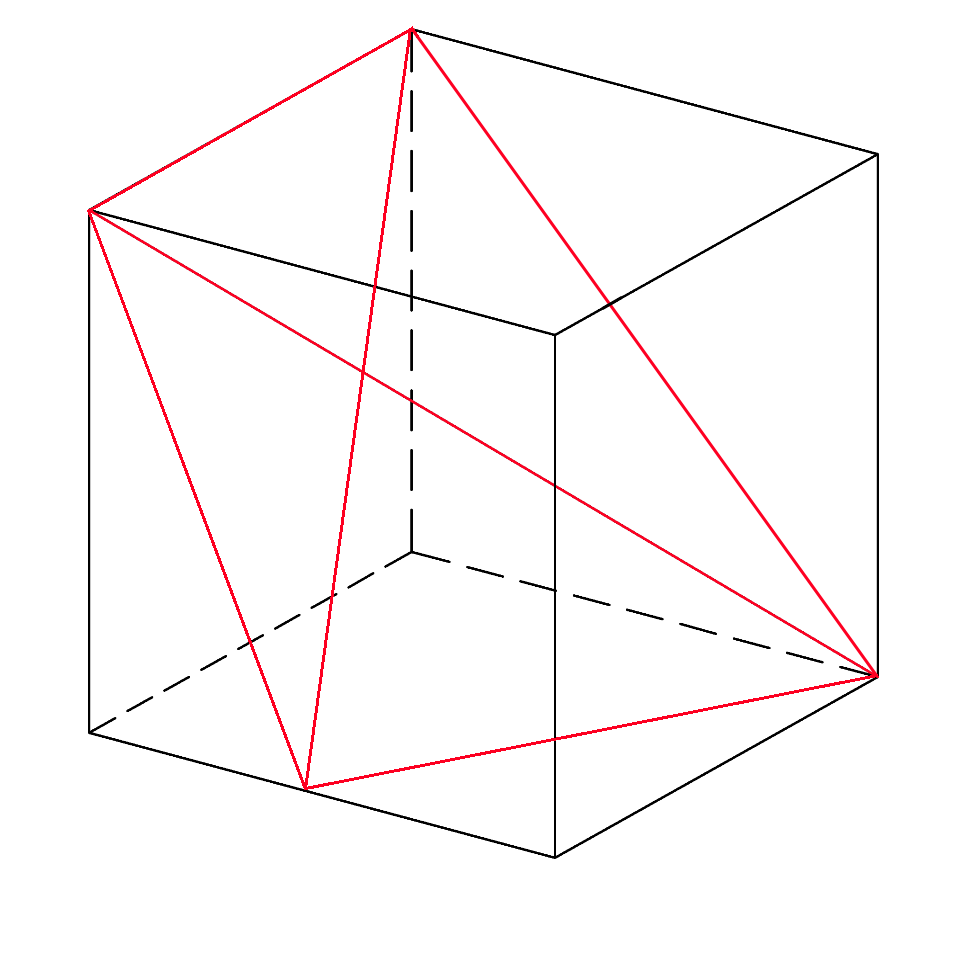
СОФИЯ

2019

**УВОД**

“Axyno” представлява смартфон приложение, предназначено за решаване на математически задачи, включващи стереометрия в тях. Целта е да се използват възможностите на мобилното устройство за симулация на виртуална заобикаляща среда, чрез която чертаенето и решаването на подобен тип задачи да могат да се пренесат в триизмерното пространство.

Проблемът, който приложението решава е невъзможността да се визуализира триизмерно тяло върху двуизмерна равнина достатъчно ясно, например чертеж върху лист хартия. Независимо чрез каква проекция представяме триизмерния образ винаги компенсираме определен аспект, което понякога прави пространственото възприемане на чертежа доста трудоемка задача.

**Пример:** В следния чертеж на вписана пирамида в куб, независимо от проекцията са нужни поне няколко секунди фокусиране и пространствено визуализиране, докато успеем да си представим изкривената форма на пирамидата.

**C1**

*Фиг 0.1 Чертеж в ортографска проекция*

**C**

**А**

**B**

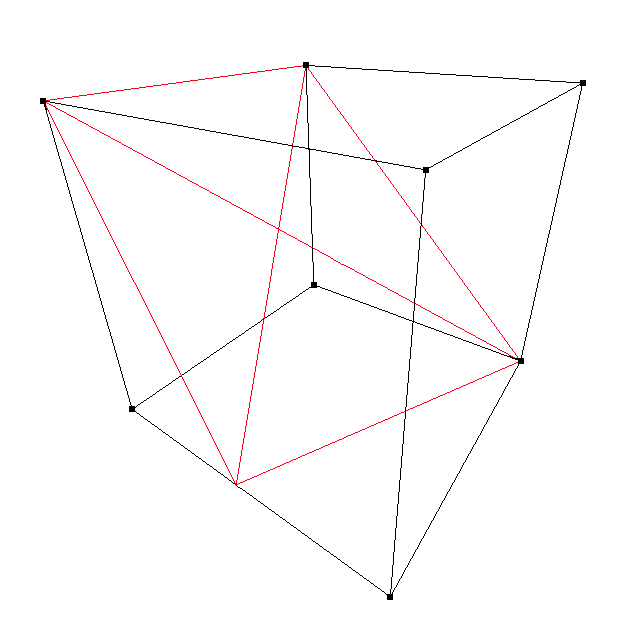
**D1**

**C1**

**D**

**А1**

**B1**



*Фиг 0.2 Чертеж в перспективна проекция*

**B**

**D1**

**B1**

**А1**

**D**

**C**

**А**

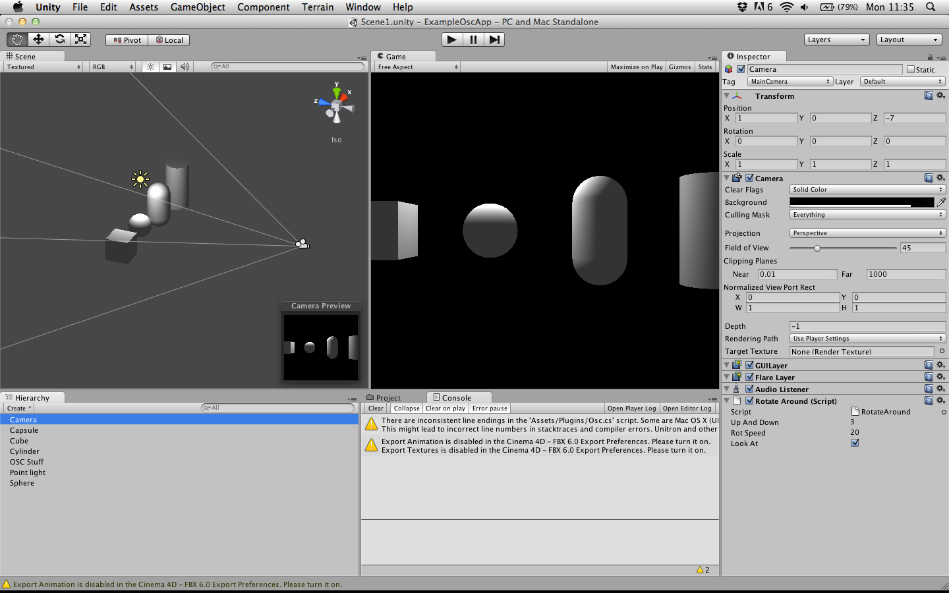
“Axyno” е единственото приложение към момента на пазара, което представлява опростен редактор на триизмерни обекти, създаден специално за използване със смартфон във виртуална среда. Приложението може да бъде използвано в училища и университети за елиминиране на проблемите при начертаването на една задача и акцентиране върху самата логическа последователност за решаването ѝ.

Работата с приложението става посредством отделен headset, предназначен за смартфони, като за целите на разработката беше използван Google Cardboard.

**ПЪРВА ГЛАВА**

Методи и технологии за реализиране на мобилни приложения

1. **Основни принципи, технологии и развойни среди, използвани при реализацията**
   1. **Технологии**
      1. **Unity Engine**





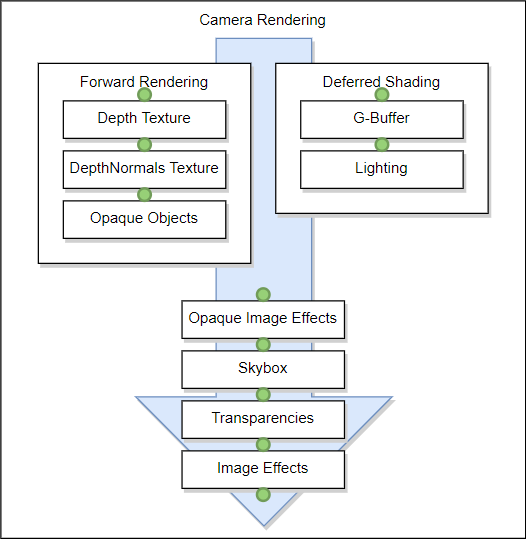
*Фиг 1.2 Потребителския интерфейс на Unity*

*Фиг 1.1 Unity лого*

Unity („Юнити“) представлява широко-разпространен междуплатформен игрови двигател („гейм енджин“/„game engine“), използван за разработката на видеоигри, симулации, анимации и архитектурни дизайни. Чрез него могат да се създават 2D и 3D приложения, а крайния продукт може да бъде експортиран за някоя от 27-те различни поддържани платформи. Програмният език е C#, а за целите на разработка на смартфон приложение - Unity поддържа следните графични библиотеки: Vulkan за Android, Metal за iOS и OpenGL ES за двете.

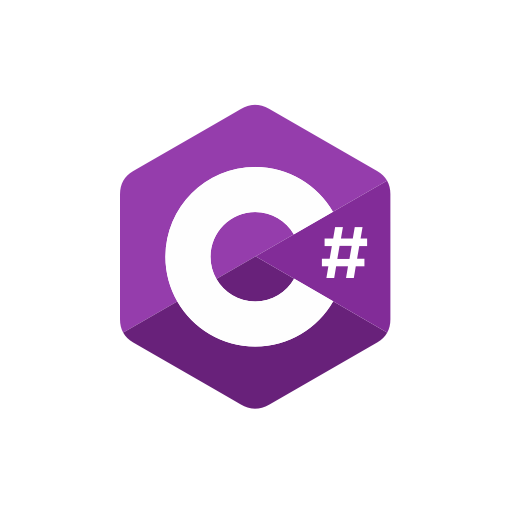
Реализацията чрез Unity има своите предимства и недостатъци, като сред главните предимства са:

* Опростен и удобен потребителски интерфейс за работа с триизмерни обекти както и възможност за тестване във всеки един момент;
* Приложението се оптимизира спрямо платформата от самия енджин;
* Графиките са приспособими за всеки екран и устройство без никаква деформация или компромис с качеството;
* Изключително лесна интеграция/вграждане на поддръжка за виртуална реалност.

Unity върши трудната работа по процеси като рендериране и оптимизиране, държейки ги далече от програмиста. По този начин вниманието и усилията му са насочени към програмируемата логика и естетическия вид на приложението.

*Фиг 1.3 Процеса на рендериране на сцена в Unity*

* + 1. **C#**



*Фиг 1.4 C# лого*

C# или C Sharp („Си Шарп“) е обектно-ориентиран език за програмиране, разработен от Microsoft, като част от софтуерната платформа „.NET“. Unity предлага първичен приложно-програмен интерфейс на C#, както за редактора под формата на различни плъгини, така и за самите разработвани приложения. Езикът е проектиран да балансира мощността на C++ с възможностите за бърза разработка на Visual Basic и Java.

В Unity поведението на определен игрален обект се управлява от закачени за него „компоненти“. Въпреки че има широк набор от вече готови такива може да е необходимо по-персонализирано управление. В такива случаи може да бъде създаден допълнителен собствен компонент с помощта на външен C# скрипт. Скриптът бива изпълнен от съответния интерпретатор, който го превежда от символен на машинен език, пропускайки стъпката по генериране на бинарен файл, както правят например компилаторите.

Едно от най-големите предимства на софтуерната платформа “.NET”, която Unity използва, е интегрираното автоматично управление на паметта. За това се грижи т.нар. система „garbage collector“, която е част от CLR (Common Language Runtime) компонента на платформата. Основната задача на тази система е да следи кога заделената памет за променливи и обекти вече не се използва и да я освобождава. Това освобождава програмиста от следене на използваните ресурси и проблеми като memory leaks („теч на памет“).

* 1. **Мобилни платформи**
     1. **Android**



*Фиг 1.5 Android лого*

Android („Андроид“) е операционната система на Google Inc. за мобилни устройства. Базирана е на Linux ядрото, като последната стабилна версия до момента е 9.0, именувана „Pie“. Unix-подобната операционна система е с open source (или „отворен код“) модел като повечето устройства съдържат и софтуер със затворен код.

За развитието на операционната система се грижат относително голям брой програмисти и софтуерни разработчици, които създават така наречените „Apps“ (от Applications - приложения), които представляват малки програми, целящи да разширят функциалността на системата. Тези приложения могат да бъдат сваляни от Интернет или от официалния магазин на Android, наречен Google Play, като приложенията от там могат да бъдат както безплатни, така и платени. Процеса по публикуване на приложение в магазина е доста лесен и общодостъпен – единственото условие е разработчика да има направен Google профил. Самия акаунт също е безплатен и за разлика от други мобилни магазини няма нито закупуване на лицензи, нито сертификационни тестове и периоди.

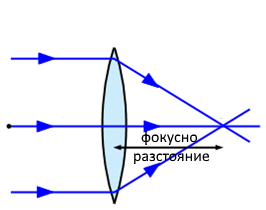
* + 1. **(iOS)**

(optional for now)

* 1. **Хардуер**
     1. **Google Cardboard**

*Фиг 1.6 Google Cardboard с поставен смартфон*

Google Cardboard е мобилна платформа за виртуална реалност, разработена от Google Inc. За разлика от други очила за виртуална реалност Google Cardboard използва процесорната мощност на смартфон за да симулира виртуална среда. Следенето на движенията и завъртанията също се осъществява от телефона, благодарение на вградените в него сензори - акселерометър и жироскоп.

 За да може образа на екрана да бъде чист и ясен продуктът използва двойно изпъкнали (биконвексни) лещи, поставени пред всяко око с фокусно разстояние от 45мм.

*Фиг 1.7 Действие на двойно изпъкнала леща*

Google предоставя три комплекта за разработка на софтуер (SDK – System Development Kit) за платформата си: един за операционната система Android, използващ Java, един за игровия двигател Unity, написан на C#, и един за операционната система iOS.

Понеже приложенията, проектирани за Google Cardboard биват пускани на мобилно устройство, след което самото то бива поставено в определеното за целта място, нямаме пряк достъп до сензорния екран на телефона. Затова интеракцията с приложението става посредством бутон, разположен в горния десен ъгъл на устройството, който притиска кондуктивно парче до екрана регистрирайки допир.



*Фиг 1.8 Местоположение на бутона за интеракция*

**ВТОРА ГЛАВА**

Проектиране на структурата на Android мобилни приложения за виртуална реалност

1. **Функционални изисквания към Android приложения**
   1. **Изисвания към изграждането на логиката**

Трябва да бъде изградена осносна структура на приложението. Да се направят основни класове чрез които добавянето на нови функционалности да става бързо и лесно. Да се изгради лесна и логически-издържана комуникация между различните части на приложението.

* 1. **Изисквания към графиката на приложението**

Приложението трябва успешно да симулира виртуална заобикаляща среда. Всеки текст, картинка и символ трябва да бъде ясно разграничаван и лесно разбираем. Графиката на приложението не трябва да забавя изпълнението на програмата, като кадрите в секунда е желателно да не падат под 30. Чертежът трябва да е с неограничена големина без това да се отразява директно на производителността.

* 1. **Потребителски интерфейс и итерация с потребителя**

Потребителският интерфейс на приложението трябва да бъде максималко опростен и разбираем. Цялата навигация ще трябва да се осъществява само чрез Google Cardboard визьора, затова всички менюта и бутони трябва да са с достатъчно голям размер че да не бъде изпитвано затруднение при използването им.

* 1. **Функционалност**

Приложението трябва да изисква възможно най-малко права от устройството. Данните, прочетени от акселерометъра и жироскопа на смартфона, трябва да бъдат извличани и обработвани достатъчно бързо, подсигурявайки следене на въртенето на главата в реално време.

* 1. **Производителност**

Приложението не трябва да забива или замръзва без причина. Моменти на насичане, прекъсване или други дефекти не трябва да бъдат забелязвани. Също така латентността трябва да бъде минимална през цялото време.

1. **Съображения за избор на програмни средства и развойната среда**
   1. **Игрови двигател Unity**

За реализацията на приложението беше взет избора да се използва игрови двигател поради многото си предимства при разработка, като например:

* Повечето от сложните и времеемки аспекти от приложението като управление на паметта, зареждане на асети, изчисляване на осветлението и др. са вече написани и готови за ползване;
* Повечето игрални двигатели са междуплатформени, което прави експортирането на приложение за специфична платформа доста лесна задача

Игровият двигател, който беше избран е Unity. Причината е че инструментите му не са непознати, тъй като вече различни проекти са били правени ползвайки него както за училище, така и за хоби. Това пропуска стъпката от заучаване на нови технологии и увеличава продуктивността.

* 1. **Google Cardboard платформа**

Понеже приложението е проектирано като продукт, имплементиращ виртуална реалност, с намерение да се използва предимно в училища използването на смартфон беше очевидния избор за устройство. Google Cardboard платформата допринася за лесната портативност и общодостъпността на приложението. Тя е евтина и лесна за работа, като не се изисква по-специализиран хадруер за нея.

* 1. **Операционна система Android**

След обстойно проучване на пазара за мобилни операционни системи се стигна до заключението че Android е най-уместният избор поради следните причини:

* Наличие на физически устройства, на които да бъде тествано приложението;
* Не се изискват допълнителни финансови средства;
* Липса на опит с други платформи;
* Лесно и бързо публикуване в магазина за приложения Google Play;
  1.  **GitHub система за контрол и управление на версиите**

*Фиг 2.1 GitHub лого*

След възникнали проблеми в системата за контрол на версиите на Unity, наречена Unity Collaborate, преместването към GitHub беше наложително. GitHub представлява уеб-базирана услуга за качване на софтуерни проекти върху отдалечен сървър (т.нар. хранилище) и осъществяването на съвместни разработки бърху тях. Базиран е на Git системите за контрол и е избран заради вече съществуващия опит с него.

* 1.  **Trello мениджър на задачи**

*Фиг 2.2 Trello лого*

Trello е уеб-базирано приложение за управление на проекти и отделните задачи по тях. Използването му се състои от класифициране на подзадачите спрямо различни критерии и следене на всичко планирано до момента, както и визуализиране на прогреса при разработката.

* 1. **Notepad++ текстов редактор**



*Фиг 2.3 Notepad++ лого*

Цялостната процедура по писане на кода беше осъществена в Notepad++. То представлява опростен текстов редактор, разпространяващ се като софтуер с отворен код. Главното му предимство е малкия му размер – почти 23MB след инсталация, както и оппростения му интерфейс и оцветяване на синтаксиса за над 50 програмни езика.

* 1. **blender моделиращ софтуер**
  2. **Paint.net графичен редактор**