**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*

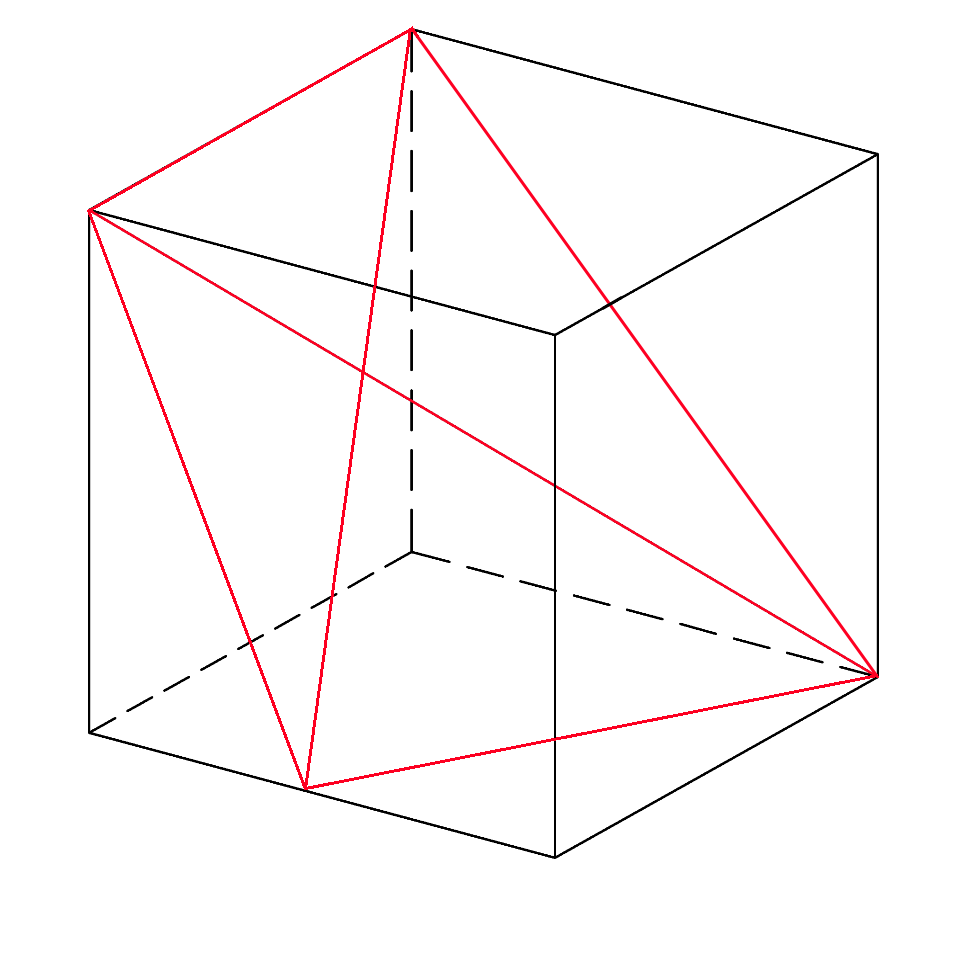
СОФИЯ

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**

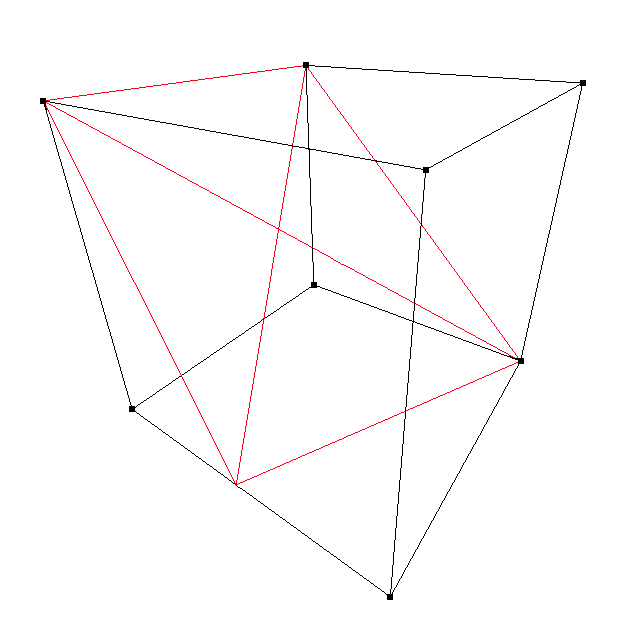
**D1**

**C1**

**D**

**А1**

**B1**



*Фиг 0.2 Чертеж в перспективна проекция [1]*

**B**

**D1**

**B1**

**А1**

**D**

**C**

**А**

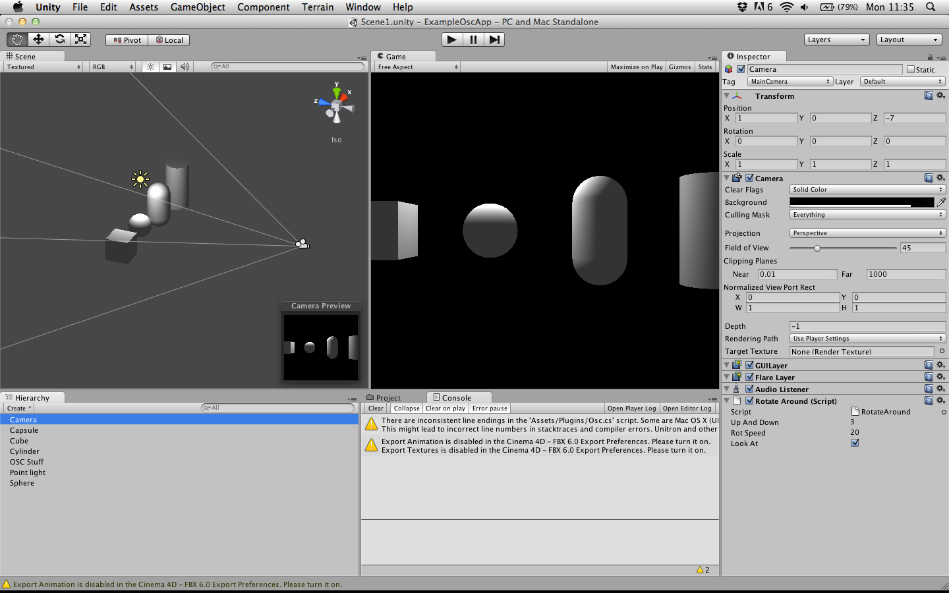
“Axyno” е единственото приложение към момента на пазара, което представлява опростен редактор на триизмерни обекти, създаден специално за използване със смартфон във виртуална среда. Приложението може да бъде използвано в училища и университети за елиминиране на проблемите при начертаването на една задача 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 [2].

**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**





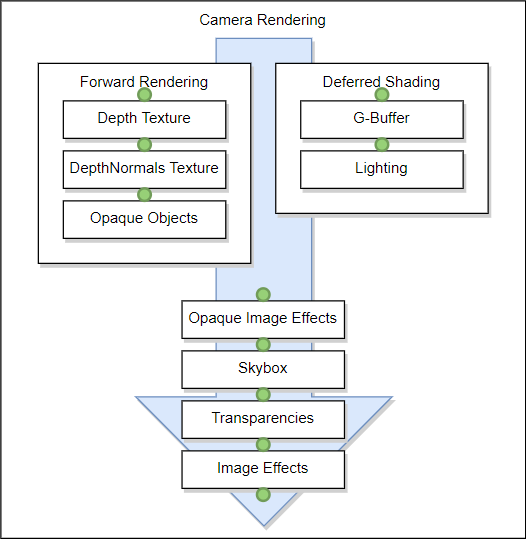
*Фиг 1.2 Потребителския интерфейс на Unity*

*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.

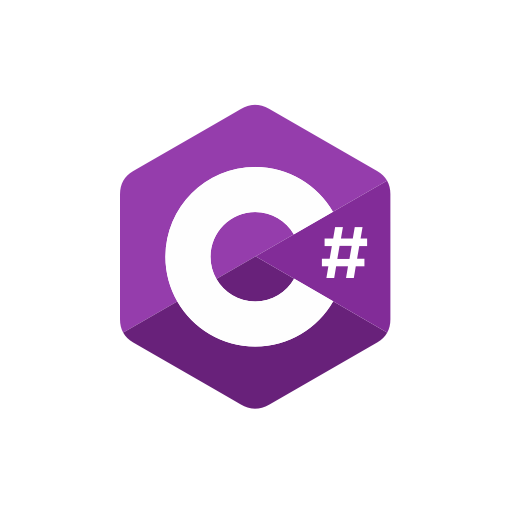
Реализацията чрез Unity има своите предимства и недостатъци, като сред главните предимства са:

* 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.

* 1. **Мобилни платформи**
     1. **Android**
     2. **iOS**
  2. **Хардуер**
     1. **Google Cardboard**