МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ  
«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ П.О. СУХОГО»

КАФЕДРА «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

КУРСОВАЯ РАБОТА  
по курсу «Объектно-ориентированное программирование»  
на тему:  
**Приложение, реализующее игру «Битва на воздушных шарах» с   
использованием Windows Form и графики OpenGL**

Выполнил студент гр. ИТИ-21  
Громыко И. В.   
Проверил доцент, кандидат   
технических наук   
Курочка К.С.

Гомель 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc129520026)

[1 АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ 4](#_Toc129520027)

[1.1 Средства решения задачи для создания графики 4](#_Toc129520028)

# **ВВЕДЕНИЕ**

**1 АНАЛИТИЧЕСКИЙ ОБЗОР СРЕДСТВ РЕШЕНИЯ ЗАДАЧИ**

* 1. **Средства решения задачи для создания графики**

В наше время для создания графики для игровых предложений существует масса подходов и вариантов решения, большинство из которых сводится к определению графического *API* (интерфейса прикладного программирования) или графической библиотеки, которые предоставляют программистам доступ к графическому оборудованию. Самыми популярными *API* на данный момент являются *DirectX*, *Vulkan* и *OpenGL*.

*OpenGL (Open Graphics Library)* – современный кроссплатформенный *API*, который позволяет работать как с двумерной, так и с трёхмерной графикой. Данный интерфейс существует уже более 20 лет и поддерживается на множестве платформ: от мобильных устройств до мощных станций. На *OpenGL* за свою историю существования было написано множество игровых приложений различных величин: от небольших проектов, до одних из самых крупных проектов в индустрии. Библиотеки для работы с *OpenGL* существуют практически для всех языков программирования, включая и *C*#. [1]

Ключевой идеей *OpenGL* является то, что он основан на модели клиент-сервер. В роли клиента выступает приложение, а в роли сервера – драйвер и графический процессор.

*OpenGL* представляет собой программный интерфейс к графическому процессору. Изначально интерфейс задумывался как не зависящий от аппаратного обеспечения *API*, то есть он может быть реализован на различных платформах. Именно поэтому в *OpenGL* не включены команды для работы с окнами и получения данных после ввода пользователя. Также у *OpenGL* есть отличия в работе с трёхмерным пространством: у интерфейса нет команд для описания различных трёхмерных моделей, сам *OpenGL* может лишь выводить примитивы точек, отрезков и треугольников. Остальные модели строятся поверх *OpenGL* на основании данных примитивов.

В течение времени спецификация *OpenGL* расширялась, в связи с чем выпускались новые версии спецификации, которые также включали поддержку *Shading Language* – шейдерного языка, который был разработан для выполнения математики для растеризации, т.е. получения растровых изображения.

Платформа Microsoft .Net Framework не имеет встроенных средств поддержки библиотеки *OpenGL*. Есть различные способы применения OpenGL на .Net Framework. Одной из таких реализаций является подключение динамической библиотеки Opengl.dll, которая является реализацией OpenGL в операционной системе Windows.

Также на данный момент существует уже несколько библиотек, в которых выполнена необходимая реализация вызовов функций из динамической библиотеки и которые предоставляют дополнительные возможности, позволяющие упростить использование OpenGL на платформе Microsoft .Net Framework. [2]