

ДУЙШЕНАЛИЕВ А., СООДАЕВ А., КАЗАКБАЕВА З.М.

Кыргыз-Турк «Манас» университети,
Бишкек

DUISHENALIEV A., SOODAEV A., KAZAKBAEVA Z.

KYRGYZ-TURKISH «MANAS» UNIVERSITY, BISHKEK

**MS VISUAL C++ КОЛДОНУП ФИГУРАЛАРДЫН ПРОПОРЦИЯСЫН САКТОО МЕНЕН
ТЕРЕЗЕЛЕРДИ ТҮЗҮҮ****Окна с сохранением пропорций фигуры с использованием MS Visual C++
Window with saving figure proportions with using MS Visual C++**

Аннотация: Баарыбызга маалым болгондой, азыркы заманда 3D-моделдөө, анимация жана объекттерди визуалдаштыруу маанилүү роль ойнойт. Ал эми DirectX жана OpenGL индустрияда 2D жана 3D-графика үчүн эң көп колдонулган API-интерфейс болуп саналат. Үч өлчөмдүү дүйнөнү чагылдыруу, моделдөө, материалдардын жайгашуусу, жарык булактары, камера, визуалдаштыруу, анимация сыяктуу баскычтардан турат. Биздин максат - компьютердик графикада жөнөкөй үч өлчөмдүү сүрөттөрдү жаратууда OpenGL китепканаларын колдонууну көрсөтүү. Убакыттын басымдуу бөлүгү API OpenGL тилинин негизинде үч өлчөмдүү сүрөттөрдүн программаларын жазуу үчүн жумшалгандыктан, биз бул макалада "MS Visual C++ колдонуп фигуралардын пропорциясын сактоо менен терезелерди түзүү" темасындагы сабакты карап чыгабыз. Ошондой эле математикалык куралдарды жана геометриялык моделдөө ыкмаларын түшүндүрүп, ар кандай маселелерди визуалдаштырууда эң таанымал алгоритмдердин жардамы менен ишке ашырабыз. Андан соң жөнөкөй кодду иштетип чыккан жеке видео сабактары чагылдырылат, жана видео сабактарды түзүүдөгү мүмкүнчүлүктөр көрсөтүлөт. Азыркы учурда биз компьютердик графиканын алгоритмдик негиздерин чагылдырган вебсайт үстүндө иштеп жатабыз.

Аннотация: 3D моделирование, анимация и визуализация объектов очень важны. В настоящее время DirectX и OpenGL являются наиболее широко распространенным API-интерфейсом для 2D и 3D-графики в отрасли. Формирование трехмерного мира состоит из следующих этапов: моделирование, верстка материалов, источники света, камера, визуализация, анимация.

Цель этой статьи - показать использование библиотеки OpenGL для создания простой трехмерной сцены в компьютерной графике. Основное внимание уделяется обучению написанию программ для трехмерной графики на основе графического процессора с использованием языка API OpenGL.

В этой статье мы рассмотрим урок «Окно с сохранением пропорций фигуры» с помощью Visual C ++. Мы описываем математические инструменты и методы геометрического моделирования на их основе и воспроизводим наиболее известные алгоритмы для решения различных задач визуализации и обработки изображений. Мы делимся примером кода. В статье также представлены персональные оригинальные видеоуроки, разработанные в Кыргызско-Турецком университете «Манас», и рассказывается об опыте создания видеоуроков. Мы находимся в процессе разработки веб-сайта по компьютерной графике, где мы показываем алгоритмические основы компьютерной графики.

Annotation: 3D modeling, animation and visualization of the objects is very important. Nowadays DirectX and OpenGL is the most widely adopted 2D and 3D graphics API in the industry. The formation of a three-dimensional world consists of the following stages: modeling, imposition of materials, light sources, camera, visualization, animation.

The purpose of this article is to show the use of the OpenGL library to create a simple 3D scene in the computer graphics. The focus is on teaching how to write GPU based 3D graphics programs using the OpenGL API language.

In this article we will see about the lesson "Window with saving figure proportions using Visual C ++. We describe mathematic tools and methods of geometric modeling based on them and reproduce the best known algorithms for solving various problems of visualization and image processing. We are sharing the sample code.

The paper also presents personal original video tutorials developed in the Kyrgyz-Turkish University Manas and shares the experience of video lessons creation. In this article, we share our experience, which will come in useful for each teacher and students. We are in the process of developing the web site on computer graphics where we show algorithmic basis of computer graphics.

Урунттуу сөздөр: эки өлчөмдүү жана үч өлчөмдүү графика; моделдөө; OpenGL, визуализация; фигуранын пропорциясы; алгоритмдик программалар; видео-сабактар; мультимедиа

Ключевые слова: 2D, 3D компьютерная графика; моделирование; графическая библиотека OpenGL; визуализация; пропорции фигуры; анимация; видео уроки; мультимедиа

Keywords: 2D, 3D computer graphics; modeling; graphics library OpenGL; visualization; figure proportions; animation; video lessons; multimedia

Киришүү

Азыркы заманда компьютердик графика тармагында объекттерди визуалдаштыруу, анимация жана үч өлчөмдүү моделдөө маанилүү роль ойнойт. 3D графика дүйнөсү - бул оюн жана кинодон баштап архитектура жана инженерияга чейин камсыз кылган татаал аймак болуп саналат. 3D сүрөтчүлөр жана дизайнерлер айлана-чөйрөнү экранга чагылдырууда, 3D дизайн, көргөзмө, анимация сыяктуу атайын техникалык процесстерди колдонушат. Үч өлчөмдүү сүрөттөрдү, 3D басып чыгаруу менен физикалык объекттерге айландыруусу 3D моделдөө деп аталат. Учурда билим берүү мекемелеринде 3Dмоделдөө боюнча орус тилинде жана өзгөчө мамлекеттик тилибиз кыргыз тилинде окуу материалдары жетишсиз. Мындан тышкары, программалоо боюнча жогорку классификациядагы адистер да жокко эсе. (Билим берүүнү жеңилдетүү максатында визуалдаштырууга болгон көз караштар жогорулайт. Демек, дүйнө жүзүндөгү компьютердик графика адистерине болгон суроо-талаптын өсүшү байкалат.)

Иштин максаты:компьютердик графикада жөнөкөй үч өлчөмдүү сүрөттөрдү жаратууда OpenGL китепканаларын колдонууну көрсөтүү. Убакыттын басымдуу бөлүгү API OpenGL тилинин негизинде үч өлчөмдүү сүрөттөрдүн программаларын жазуу үчүн жумшалгандыктан биз бул макалада "MS Visual C++ колдонуп фигуралардын пропорциясын сактоо менен терезелерди түзүү" темасындагы сабакты карап чыгабыз.

MS Visual Studio платформасында OpenGL китепканасын колдонуу менен практикалык тажрыйба топтоо. Жөнөкөй мисал катары үч өлчөмдүү объектти OpenGL китепканаларын пайдалануу аркылуу көрсөтүү. Visual Studio 2010 платформасына OpenGLди туташтыруу. Терезенин салыштырмалуу пропорциясын сактап калуу. Кош буферлөө жана анимация. Клавиатура. GLUT китепканасы. Чычкан менен иштөө. Камераны жылдыруу.

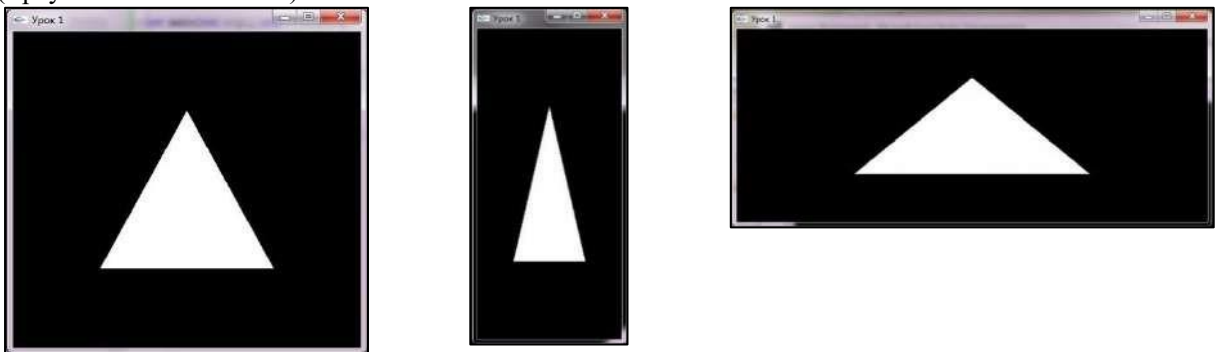
Маселе:OpenGL китепканасынын жардамы менен MS Visual Studio платформасын пайдалануу жана C++ тилинде коду ишке ашырууда тажрыйба топтоо.Ошондой эле, кыргыз тилинде үч өлчөмдүү моделдөө жөнүндө видео-сбактарды даярдоо. **Практика:**

Кыргыз-Түрк “Манас” университетинде «Компьютердик графика» сабагы алгоритмдик деңгээлде окутулат. Графикалык программалоо OpenGLжанаDirectX китепканаларын колдонуу аркылуу VisualC++ программалоо тилинде ишке ашат [1,2].

Open Graphics Library (OpenGL) – “ачык графикалык китепкана” дегенди билдирет. Башкача айтканда OpenGL – бул өзүнө жүздөгөн функцияларды камтыган спецификация. Ал програмдоо тилинен көз карандысыз кросс-платформалуу програмдык интерфейсти аныктайт. Анын жардамы менен програмчы эки жана үч өлчөмдүү компьютердик графиканы колдонгон тиркеме түзө алат. Эң жөнөкөй 3 өлчөмдүү көргөзмөлөрдү чагылдырууда OpenGL китепканасынын колдонулушун демонстрациялоо. (C++ тилинде коду ишке ашыруу)

Сабак. Фигуранын пропорцияларын сактоо

Буга чейин даярдалган коду иштетип көрөлү: консоль терезеси жана OpenGL терезеси чыгат. Эми терезенин өлчөмүн эни бийиктигине дал келбегендей кылып өзгөрткүлө. Үч бурчтук формасы бузулуп калат. (Треугольник исказится.)



Сүрөт 1. Унчукпоо боюнча түзүлгөн терезе.

Сүрөт 2. Пропорциялары өзгөргөн терезелер.

Мунун себеби перспективдүү терезенин туура эмес түзүлгөнү. Унчукпоо боюнча көрсөтүлгөн элементтердин пропорциясы 1:1. Ошондуктан пропорция өзгөргөндө перспектива бузулат. Перспектива бузулбаш үчүн терезе өлчөмдөрү өзгөргөндө, перспективаны кайра эсептөөчү функцияны аныкташ керек. Ал функцияны визуализация функциясы сыяктуу эле GLUT – та каттоо керек.

glutReshapeFunc функциясы , перспективаны кайра эсептөөчү функцияны каттайт. `void glutReshapeFunc(void(*func)(int width, int height));` параметрлери:

*func - терезе өлчөмдөрү өзгөргөндө, перспективаны кайра эсептөөчү функциянын аталышы. Эми өткөн сабактагы main функциясына кайтып, *glutReshapeFunc* чакырылышын кошуу керек. Терезенин өлчөмүнүн өзгөрүшүн көзөмөлдөгөн функциябызды *ChangeSize* деп атайлы.

```
int main(int argc, char ** argv) {
// GLUT-ту инициалдоо
glutInit(&argc,argv); //Экранга чыгаруу режимин аныктоо
glutInitDisplayMode(GLUT_RGB|GLUT_DEPTH|GLUT_DOUBLE); //RGB - түс
моделин колдонуу| тереңдик буферин колдонуу | кош буфер колдонуу
glutInitWindowPosition(100,100); //терезенин позициясын (координаттарын) коюу
glutInitWindowSize(500,500); // терезенин өлчөмдөрүн коюу
glutCreateWindow ("Сабак: Фигуранын пропорцияларын сактоо"); //Терезе түзүү

//функцияларды каттоо
glutDisplayFunc(renderScene); //сүрөт тартуу функциясы

//Жаңы функция
glutReshapeFunc(changeSize); //терезенин өлчөмдөрүн кайра эсептөөчү функциясы

// GLUT-тун негизги циклы
glutMainLoop();

return 1;
}
```

Кийинки кадамда перспективаны көзөмөлдөгөн *ChangeSize* функциясын аныктайлы. Бул функция эки аргументке ээ, булар клиенттик областын эни жана бийиктигинин жаңы маанилери.

```
void changeSize(int w,int h) //терезенин өлчөмдөрүн кайра эсептөөчү функция
{
if(h==0) h=1; //терезе бийиктиги нөлгө барабар эмес экенин текшеребиз
float ratio=1.0*w/h;
// Проекция матрицасын колдонобуз, бул матрица сценанын көлөмүн
аныктайт
glMatrixMode(GL_PROJECTION);
// Проекция матрицасын инициалдоо үчүн бирдик матрицаны жүктөйбүз
(Матрицаны баштапкы абалга келтиребиз)
glLoadIdentity(); //бирдик матрицаны жүктөө
// Кароо терезесин аныктайбыз
glViewport(0,0,w,h);
// Туура перспектива коёбуз
gluPerspective(45,ratio,1,1000);
// моделге кайтабыз
glMatrixMode(GL_MODELVIEW);
}
```

Биринчи этапта эн жана бийиктиктин катышын эсептейбиз. Аны туура эсептеш үчүн бийиктик нөлгө барабар болгон учурду эске алыш керек. Андан кийин туура проекция матрицасын коёбуз. Бул матрица сценаны көлөмүн аныктайт. Аны инициалдоо үчүн бирдик матрицасын жүктөйбүз. Анан терезе *glViewport* функциясы менен көрсөтүлсүн дейбиз. Алгачкы эки параметр жаңы терезе үчүн координатанын башталышы, акыркы экөө терезенин эни жана бийиктиги.

GluPerspective OpenGL –дин башка китепканасындагы функция. Ал китепкана OpenGL утилиталар китепканасы же GLU. GLU OpenGL реализациясынын стандарттуу компоненти. *GluPerspective* кароо параметрлерин коёт. Алгач YZ тегиздигиндеги кароо талаасы менен бурч коюлат, анан кароонун эни менен бийиктигинин катышы коюлат. Акыркы эки параметр кесилиш

тегиздиктеринин жакынкы жана алыскы чегин аныктайт. Жакынкы мааниден жакын жана алыскы мааниден алыс болгон бардык нерселер сценадан кесилет (сценада көрсөтүлбөйт). Бул тескөөлөр менен этият болуш керек, же болбосо эч нерсе көрүнбөй калышы мүмкүн. Акыры бардык матрицалык операциялар модель матрицасын колдонорун OpenGL –ге көрсөтөбүз.

Жыйынтыктоочу код:

```
#include<gl\glut.h>

void renderScene()//сүрөт тартуу функциясы {
    //түс жана тереңдик буферин тазалоо
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glBegin(GL_TRIANGLES); //фигураны тартып башто, үч бурчтук тартуу
        //үч бурчтуктун координаталары
        //xy    z
        glVertex3f(-1.5,-1.5,-5.0); //чекиттин координаталары
        (x,y,z) glVertex3f( 0.0, 1.5,-5.0); glVertex3f( 1.5,-1.5,-5.0);
    glEnd(); //фигураны тартып бүтүү

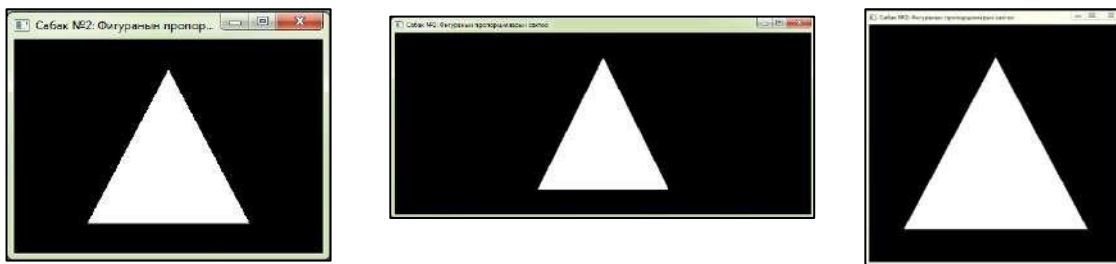
    glutSwapBuffers(); //буферлерли алмаштыруу
} voidchangeSize(intw,inth)//терезинин өлчөмдөрүн кайра эсептөөчү

функциясы

{
//терезе бийиктиги нөлгө барабар эмес экенин
текшеребиз if(h==0) h=1; float ratio=1.0*w/h;
// Проекция матрицасын колдонобуз, бул матрица сценанын көлөмүн аныктайт
glMatrixMode(GL_PROJECTION);

// Проекция матрицасын инициялоо үчүн бирдик матрицаны жүктөйбүз (Матрицаны
баштапкы абалга келтиребиз) glLoadIdentity();//бирдик матрицасы // Кароо терезесин аныктайбыз
glViewport(0,0,w,h);
// Туура перспектива коёбуз
gluPerspective(45,ratio,1,1000);
// моделгекайтабыз
glMatrixMode(GL_MODELVIEW);
}
int main(intargc, char ** argv)
{
// GLUT-туинициялоо
glutInit(&argc,argv);
//Экранга чыгаруу режимин аныктоо
glutInitDisplayMode(GLUT_RGB|GLUT_DEPTH|GLUT_DOUBLE); //RGB – түс моделин
колдонуу|) тереңдик буферин колдонуу | кош буфер
колдонуу
glutInitWindowPosition(100,100); //терезенин позициясын (координаттарын) коюу
glutInitWindowSize(500,500); // терезенин өлчөмдөрүн коюу

glutCreateWindow("Сабак: Фигуранын пропорцияларын сактоо");//Терезе түзүү
// функцияларды каттоо
glutDisplayFunc(renderScene);//сүрөт тартуу функциясы
glutReshapeFunc(changeSize);//терезинин өлчөмдөрүн кайра эсептөөчү функциясы
// GLUT-тун негизги циклы
glutMainLoop(); return 0; }
```



Сүрөт 3.фигуралардын пропорциясын сактоо менен терезелерди түзүү

Жогорудагы мисалда көрсөткөндөй, математикалык куралдарды жана геометриялык моделдөө ыкмаларын түшүндүрүп, ар кандай маселелерди визуалдаштырууда эң таанымал алгоритмдердин жардамы менен ишке ашырабыз. Биз жөнөкөй коддун иштешин көрсөттүк.

Видео окутууларда, мугалимдер үчүн пайдалуу боло турган видео-сабактарды түзүү аркылуу тиешелүү тажрыйбаларыбыз менен бөлүшөбүз. Атайын программалык жана графикага багытталган тиркемелердин жардамы менен, сабактардын канчалык деңгээлде кызыктуу өтүлөөрүн өзүбүздүн видео сабактарыбызда тартуулап кетмекчибиз[3].

Практикада, аптада бир гана сабактын, айрыкча, үч өлчөмдүү графика сабагынын жетишсиз экенин көрсөтөт. Ошондуктан студенттер компьютердик класстарга келип, өз алдынча иштегенге убакыт бөлүшү зарыл. Сабактан тышкары студенттер бир тема тандап, проект даярдашат. Бул проект жалпы баанын 50%ы катары бааланат. Ал эми финалдык сынак, жазуу түрүндөгү тесттин жыйынтыгы менен жалпы баанын 50%ын түзөт.

Жыйынтык:

- Заманбап үч өлчөмдүү компьютердик графика кадимки бир сүрөттөлүштөн айырмалоого кыйын болгондой реалдуу моделди түзүүгө мүмкүнчүлүк берет.
- Студенттер объекттин калыбын (макет) даярдай алышат жана уникалдуу долбоорлорун ишке ашыра алышат.
- Татаал математикалык моделдерди көрүнүктүү болгондой демонстрациялайт. □
- Студенттер компьютердик графиканын учурдагы зарылчылыгын түшүнүшөт.
- Кыргыз тилинде видео сабактар даярдалат. *Колдонулган адабияттардын тизмеси*

1. Казакбаева З. М., Байбосунова М. «Применения компьютерной графики в учебных курсах» Вестник КНУ им. Ж.

Баласагына, 2010, серия 3, вып.4, с. 110-112

2. Казакбаева З.М., Бекболотова Г. "Создание видео уроков по компьютерной графике" Вестник Вестник БГУ,2015, №40, с.142-144

Рецензенты: Бримкулов У.Н.– доктор технических наук, профессор, член-корр. НАН КР, Шаршенбаев Б.Ш. – кандидат физико-математических наук, доцент