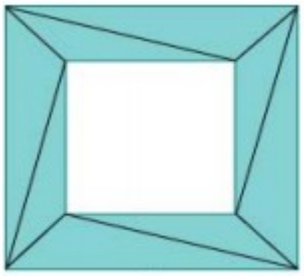
13.10.2019

Hazırlayan: Tayfun GÜRLEVİK
Öğrenci No: N19139647
ÖDEV NO: 1

Soru1:

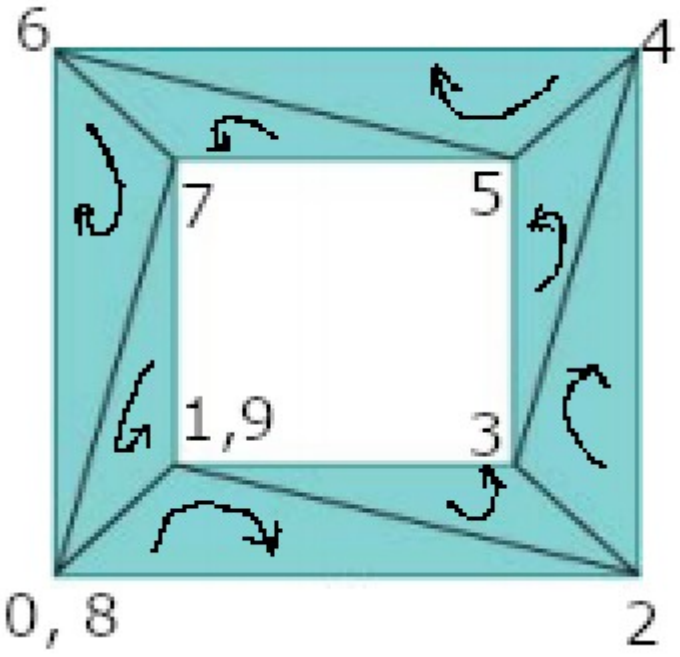
OpenGL kütüphanesini kullanarak aşağıdaki şekli tekbir üçgen şerit kullanarak çizdiren C++ programını çalıştırınız.



Yanıt:

a) Çözüm

Sorunun çözümü için üçgenleri oluşturacak vertexler aşağıdaki şekildeki gibi numaralandırılmıştır.



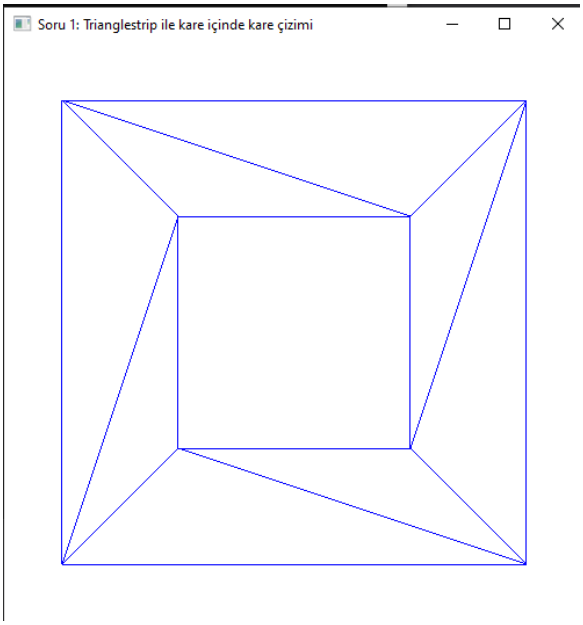

Koordinatlar(x,y,z):

Vertex0=Vertex8=(10,10,0)
Vertex1=Vertex9=(30,30,0)
Vertex2=(90,10,0)
Vertex3=(30,70,0)
Vertex4=(90,90,0)
Vertex5=(70,70,0)
Vertex6=(10,90,0)
Vertex7=(30,70,0)

b) Kod

Sorunun çözümü için Square.cpp dosyasında drawScene() yordamı aşağıdaki şekilde değiştirilmiştir.

```
void drawScene(void)
{
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0.0, 0.0, 0.0);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
glBegin(GL_TRIANGLE_STRIP);
glColor3f(0.0, 0.0, 1.0);
glVertex3f(10.0, 10.0, 0.0); //Vertex0
glVertex3f(30.0, 30.0, 0.0); //Vertex1
glVertex3f(90.0, 10.0, 0.0); //Vertex2
glVertex3f(70, 30.0, 0.0); //Vertex3
glVertex3f(90, 90, 0.0); //Vertex4
glVertex3f(70, 70, 0.0); //Vertex5
glVertex3f(10, 90.0, 0.0); //Vertex6
glVertex3f(30.0, 70.0, 0.0); //Vertex7
glVertex3f(10.0, 10.0, 0.0); //Vertex8 =Vertex0
glVertex3f(30.0, 30.0, 0.0); //Vertex9 =Vertex1
glEnd();
glFlush();
}
```




c)Ekler

Soru1.cpp, BCA611 Homework1-Soru1.exe

Soru2:
Inverse square root problemi: Bilgisayar oyunları programlarında en sık kullanılan matematiksel işlemlerden biri $1/\sqrt{x}$ işlemidir. Bu işlemin sonucu farklı bir yolla bulan bir program ilk olarak Quake oyununda kullanılmıştır. Aşağıda yer alan bağlantılarda bu farklı yöntem açıklanmaktadır. Sizlerin de bu yöntemi okuyup anlamanız ve bahsi geçen kodları çalıştırmanız beklenmektedir.

https://betterexplained.com/articles/understanding-quakes-fast-inverse-square-root/

https://medium.com/hard-mode/the-legendary-fast-inverse-square-root-e51fee3b49d9
Yanıt:
a) Çözüm
Bir denklemin köklerini sayısal yöntemler ile bulmada en yaygın olarak kullanılan yöntemlerden birisi Newton yöntemidir.
Bu yöntem istenilen denklem kökünün ilk tahmini yapılarak, iterasyonlar sonucunda yakınsamanın elde edilmesiyle istenilen sonucu alınmasına dayanmaktadır.
$x_0$=İlk tahmin olmak üzere

$x_1=x_0-f(x_0)/f'(x_0)$,
.
.
.
$x_{n+1}=x_n-f(x_n)/f'(x_n)$
şeklindeki iterasyonlar sonucunda $(x_{n+1}-x_n)/x_n$ değeri 0'a yakınsadığı zaman denklemin kökü elde edilmiş olur.

Burada iterasyon sayısını azaltmak için ilk tahmin değerinin gerçek değere yakınlığı önemlidir.

$1/\sqrt{x}$ denklemini Newton yöntemi ile çözmeye çalışırsak:

$y= 1/\sqrt{x}$
$y^2=1/x$
$x=1/y^2$
$1/y^2-x=0$
ve
$f(y)=1/y^2-x$ foksiyonunu elde ederiz.

Newton tanımından,

$y_{n+1}=y_n-f(y_n)/f'(y_n)$, $n\geq 0$
$y_{n+1}=1/2y_n(3-xy_n^2)$ bu işlem aşağıdaki kodda x=x*(1.5f -xhalf*x*x); ile ifade edilmiştir.

```
float InvSqrt(float x){
      float xhalf = 0.5f * x;
      int i = *(int*)&x;            // store floating-point bits in integer
      i = 0x5f3759df - (i >> 1);    // initial guess for Newton's method
      x = *(float*)&i;              // convert new bits into float
      x = x*(1.5f - xhalf*x*x);     // One round of Newton's method
      return x;
   }
```

İlk tahminin elde edilmesi:
yordama float türünde gönderilen x değerini önce bitlere dönüştürüp i adındaki integer değişkene aktarılıyor.
int i = *(int*)&x;
0x5f3759df sihirli rakamıyla (http://www.lomont.org/papers/2003/InvSqrt.pdf adresindeki makalede neden bu değerin kullanıldığı açıklanıyor.) ilk tahmin değeri elde ediliyor. Bölmeden kaynaklanacak hataları minimize etmek amacıyla bitwise operatör kullanılıyor.

Bu ilk tahmin değeri yarımıyla ilk iterasyonda yaklaşık binde 2 hata ile tahmin yapılıyor. İkinci iterasyonda ise hata 0'a yakınsıyor.

b)Kod

```
#include <iostream>
using namespace std;

float InvSqrt(float x) {

	float xhalf = 0.5f * x;
	int i = *(int*)&x;            // store floating-point bits in integer
	//burada 0x5f3759df hexadecimal sayýsýný 1 bit yana kaydýrarak ilk tahmini oluþturacak sayýyý hexadecimal formatýnda elde ediyor
	i = 0x5f3759df - (i >> 1);    // initial guess for Newton's method
	//Bu yöntemin büyüsü 0x5f3759df sayýsýnda olduðu kadar C/C++ dilinin bit kaydýrma operatörününde de saklý.
	x = *(float*)&i;              // convert new bits into float
	x = x * (1.5f - xhalf * x * x);     // One round of Newton's method
	return x;
}
float Q_rsqrt(float number)
{
	long i;
	float x2, y;
	const float threehalfs = 1.5F;

	x2 = number * 0.5F;
	y = number;
	i = *(long*)&y;    // evil floating point bit level hacking
	i = 0x5f3759df - (i >> 1);               // what the fuck?
	y = *(float*)&i;
	y = y * (threehalfs - (x2 * y * y));   // 1st iteration
	y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration,
                                                                                // this can be removed

	return y;
}
int main()
{
	float gercekDeger = 1 / sqrt(2);
	float quakeDeger = InvSqrt(2);
	float quakeDeger2 = Q_rsqrt(2);
	cout << "1/kok(2) nin gercek degeri: " << gercekDeger << endl;
	cout << "1 / kok(2) nin quake yontemiyle hesaplanan degeri: " << quakeDeger << endl;
	cout << "Hata orani: " << abs(gercekDeger - quakeDeger) / gercekDeger << endl;
	cout << "Newton yontemiyle 2 iterasyon sonucu 1 / kok(2):" << quakeDeger2 << endl;
	cout << "Hata orani: " << abs(gercekDeger - quakeDeger2) / gercekDeger << endl;
	return 0;
}
```

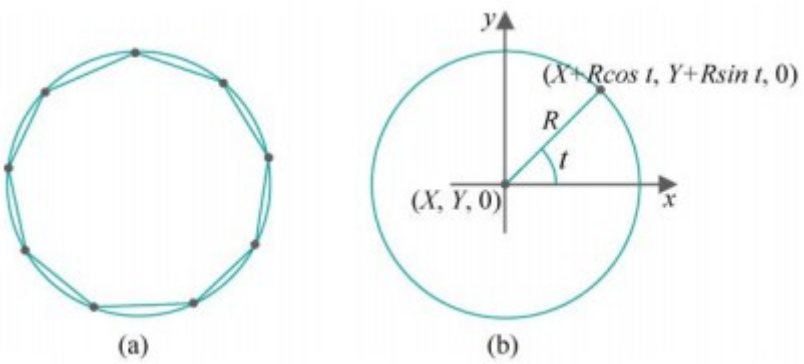
```
Microsoft Visual Studio Debug Console
1/kok(2) nin gercek degeri: 0.707107
1 / kok(2) nin quake yontemiyle hesaplanan degeri: 0.70693
Hata orani: 0.000249931
Newton yontemiyle 2 iterasyon sonucu 1 / kok(2):0.707107
Hata orani: 1.68587e-07

C:\Users\Tayfun\Documents\GitHub\BCA-611-Odev-ve-Projeler\BCA611 Homework1 Soru2\Debug\BCA611 Homework1 Soru2.exe (proce
ss 16668) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

c)Ekler:
Soru2.cpp, `BCA611 Homework1 Soru2.exe`

Soru3:
Çemberin parametrik denklemi (X,Y,0) çemberin merkezi ve çemberin yarıçapı R olmak üzere

$$x = X + R\cos(t), y = Y + R\sin(t), \quad z = 0, \quad 0 \le t \le 2\pi$$

ile verilmektedir.
Bu parametrik denklemden yararlanarak OpenGL ile çemberi çizdiren bir C++ programı yazınız.




Yanıt:
a)Çözüm

Çemberi çizdirmek için GL_LINE_LOOP kullanılmıştır.

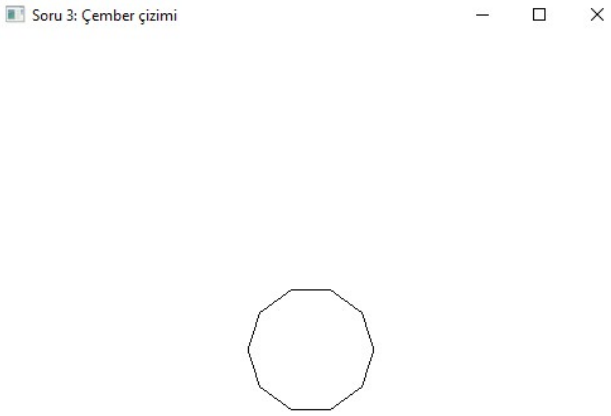
V, çizgi loopunu oluşturacak vertex sayısı olmak üzere,
iki vertex arasındaki açı;
t=2π/V olacaktır.
Bu durumda çizgi loopunu oluşturmak için;
for(int i=0;i<V;i++)
glVertex2f(X+R*cos(i*t),Y+R*sin(i*t));
şeklinde bir döngü yazmamız yeterli olacaktır.

b)Kod

```
#include <GL/glew.h>
#include <GL/freeglut.h>
#define _USE_MATH_DEFINES
#include <math.h>

void CemberCiz(float merkezX, float merkezY, float yaricap, int vertexSayisi)
{
	glBegin(GL_LINE_LOOP);
	for (size_t i = 0; i < vertexSayisi; i++)
```

```
	{
		glVertex2f(merkezX + yaricap * cos(i * 2 * M_PI / vertexSayisi), merkezY +
yaricap * sin(i * 2 * M_PI / vertexSayisi));
	}

	glEnd();
}

// Drawing routine.
void drawScene(void)
{
	glClear(GL_COLOR_BUFFER_BIT);

	glColor3f(0.0, 0.0, 0.0);
	int vertex = 10;
	float X = 50.0f;
	float Y = 50.0f;
	float R = 10;
	CemberCiz(X, Y, R, vertex);


	glFlush();
}

// Initialization routine.
void setup(void)
{
	glClearColor(1.0, 1.0, 1.0, 0.0);
}

// OpenGL window reshape routine.
void resize(int w, int h)
{
	glViewport(0, 0, w, h);

	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	glOrtho(0.0, 100.0, 0.0, 100.0, -1.0, 1.0);

	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
}

// Keyboard input processing routine.
void keyInput(unsigned char key, int x, int y)
{
	switch (key)
	{
	case 27:
		exit(0);
		break;
	default:
		break;
	}
}

// Main routine.
int main(int argc, char** argv)
{
	glutInit(&argc, argv);

	glutInitContextVersion(4, 3);
	glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);

	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
```

```
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(100, 100);

        glutCreateWindow("Soru 3: Çember çizimi");

        glutDisplayFunc(drawScene);
        glutReshapeFunc(resize);
        glutKeyboardFunc(keyInput);

        glewExperimental = GL_TRUE;
        glewInit();

        setup();

        glutMainLoop();
}
```




c) Ekler

Soru3.cpp, `BCA611 Homework1-Soru3.exe`