

Homework 7 (Aplicación Movimiento Circular)

Miguel Tlapa Juárez

17/03/2014



This document describes the system architecture and design about the body controller module, it's have block diagram and flowchart to describe software and hardware architecture.

Revision History

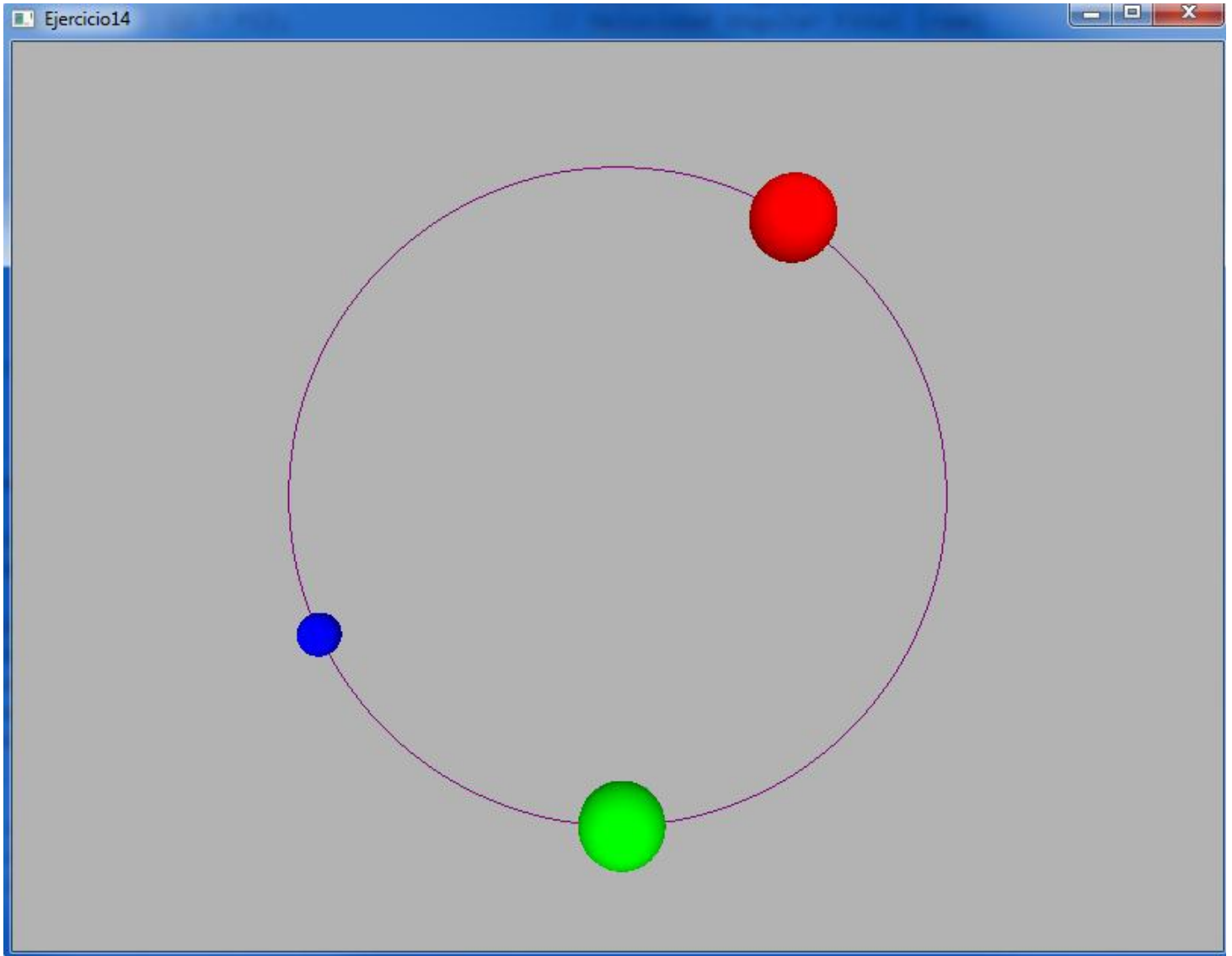
Date	Revision Number	Author/Editor	Modifications
January 2014	0.1	Miguel Tlapa	Created file

Disclaimers

1. Explanation

Tomando como referencia el ejercicio 15 , agregue 2 nuevos elementos, que se trasladan en la trayectoria circular.

Cada uno inicia con angulos diferentes.



```
// *****
```

```
// ***** Instituto Tecnológico y de Estudios Superiores de Occidente *****
```

```
// ***** ITESO, Universidad Jesuita de Guadalajara *****
```

```
// ***** *****
```

```
// ***** Especialidad en Computación Gráfica para Videojuegos *****
```

```
// ***** ECG2227A - Fisica para el Modelado de Sistemas Reales *****

// *****

// ***** Ejercicio Numero 14 *****

// *****

#include <math.h>

#include <stdio.h>

#include "PhAPI.h"

#define PI 3.14159265

using namespace PhAPI;

// *****

// ***** DEFINICION DE VALORES PARA LA MANIPULACION DEL ENTORNO *****

// *****

int i = 1; // Vector Unitario para
Direccionar X

int j = 1; // Vector Unitario para
Direccionar Y

int k = 1; // Vector Unitario para
Direccionar Z

float FPS = 60; // Muestras por Segundo

float w = 0; // Velocidad Angular Inicial [rpm]

float t = 5; // Tiempo [segundos]

float rev = 2; // Revoluciones [número]

float r = 1.5; // Radio de Giro [unidades]
```

```
// ***** Inicializacion de los Movimientos en 3D *****
```

```
int width = 800; // Ancho de la Ventana
int height = 600; // Alto de la Ventana
int cuenta = 1; // Control para Segmentos de Velocidad
float angulo = 0; // Control para Segmentos de Angulos
float angulo2 = 10;
float angulo3 = 20;
float x = 0; // Direccion Inicial en el Eje X
float y = 0; // Direccion Inicial en el Eje Y
float z = -5; // Direccion Inicial en el Eje Z
```

```
// Primitiva 2
```

```
float x2 = 0; // Direccion Inicial en el Eje X
float y2 = 0; // Direccion Inicial en el Eje Y
float z2 = -5; // Direccion Inicial en el Eje Z
```

```
// Primitiva 3
```

```
float x3 = 0; // Direccion Inicial en el Eje X
float y3 = 0; // Direccion Inicial en el Eje Y
float z3 = -5; // Direccion Inicial en el Eje Z
```

```
// ***** Calculo de los Parametros de Movimiento Circular *****
```

```
float wr = w * 2 * PI / 60; // Velocidad Angular Inicial [radianes / segundo]
```

```

float Ar = rev * 2 * PI; // Desplazamiento Angular Total [radianes]
float A = Ar * 180 / PI; // Desplazamiento Angular Total [grados]
float s = r * Ar; // Desplazamiento Lineal Total [unidades]
float a = 2 * (Ar - (wr * t)) / pow(t, 2); // Aceleracion Angular [radianes / segundo^2]
float Wr = wr + (a * t); // Velocidad Angular Final [radianes / segundo]
float W = Wr * 60 / (2 * PI); // Velocidad Angular Final [rpm]

```

```

// ***** Calculo de los tiempos de Movimiento *****

```

```

int divs = ((t * FPS) + 0.5); // Numero de Divisiones en el Segmento
float tm = t * 1000; // Tiempo de Trayectoria [milisegundos]
float incT = (float) s / divs; // Incremento Fijo en la Trayectoria
[segmentos/milisegundos]
float delaym = (float) incT * tm / s; // Tiempo para un solo segmento [milisegundos]
float delays = delaym / 1000; // Tiempo para un solo segmento [segundos]

```

```

// ***** Calculo del Vector de Velocidad Angular en Movimiento Circular **

```

```

float *incC = anguloC(divs, wr, a, delays);

```

```

// ***** Metodo para Realizar el Dibujo de las Primitivas *****

```

```

void DibujarVentana()
{
    // Dibujar la Circunferencia de Referencia
    setColor(1.0, 0.0, 1.0); drawCircle(0, 0, z, r);
}

```

```

// Dibujar la Primitiva
setColor(0.0, 0.0, 1.0); drawSphere(x, y, z, 0.1);
setColor(1, 0.0, 0);   drawSphere(x2, y2, z2, 0.2);
setColor(0, 1.0, 0);   drawSphere(x3, y3, z3, 0.2);

// Determinacion de la Posicion en Funcion del Tiempo
if ( cuenta < divs )
{
    angulo = angulo + incC[cuenta];
    angulo2 = angulo2 + incC[cuenta];
    angulo3 = angulo3 + incC[cuenta];

    x = i * r * cos(angulo);
    x2 = i * r * cos(angulo2);
    x3 = i * r * cos(angulo3);

    y = j * r * sin(angulo);
    y2 = j * r * sin(angulo2);
    y3 = i * r * sin(angulo3);

    //z = -5 + k * r * sin(angulo);
    cuenta++;
}
}

```

```

// *****
// ***** PROGRAMA PRINCIPAL *****
// *****

int main()
{
    printf("Velocidad Angular Inicial: %0.0f rpm \n"

        "Velocidad Angular Final: %0.0f rpm \n"

        "Aceleracion Angular: %0.2f rad/s2 \n"

        "Tiempo: %0.2f s \n"

        "Radio de Giro: %0.2f u \n"

        "Desplazamiento Lineal Total: %0.2f u \n"

        "Desplazamiento Angular Total: %0.2f grados \n", w, W, a, t, r, s, A);

    clear(); // Inicia el Dibujo de la Ventana

    setDelay(delaym);

    createWindow("Ejercicio14", width, height);

    setBackground(0.7, 0.7, 0.7);

    setDisplay(DibujarVentana);

    showWindow();

    return 0;
}

```