

CSE 3901

Final Exam Review

1. Review all of the material from the midterm and meidterm review
2. Provide a detailed code review for the following code. Include comments and line numbers on the table below.

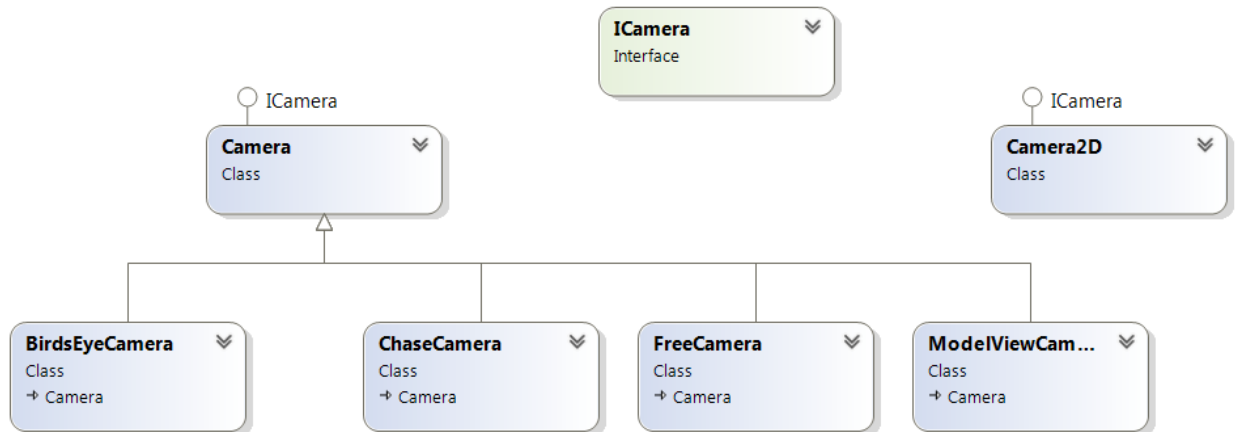
```
1. namespace Nine.Graphics
2. {
3.     using System;
4.     using Microsoft.Xna.Framework;
5.     using Microsoft.Xna.Framework.Input;
6.     using Microsoft.Xna.Framework.Graphics;
7.
8.     /// <summary>
9.     /// Defines a camera that emulates the bird eye view.
10.    /// </summary>
11.    public class BirdEyeCamera : Camera
12.    {
13.        public float Yaw { get; set; }
14.        public float Pitch { get; set; }
15.        public float Radius { get; set; }
16.        public float MinRadius { get; set; }
17.        public float MaxRadius { get; set; }
18.        public float WheelSpeed { get; set; }
19.        public float Speed { get; set; }
20.
21.        public MouseButton RotateButton { get; set; }
22.        public MouseButton TranslateButton { get; set; }
23.
24.        public Vector3 LookAt
25.        {
26.            get { return lookAt; }
27.            set { lookAt = value; }
28.        }
29.
30.        private Input input;
31.        private Vector3 lookAt = Vector3.Zero;
32.        private Point startPoint = Point.Zero;
33.
34.        public BirdEyeCamera(GraphicsDevice graphics) : this(graphics, 50,
MathHelper.PiOver2 * 0.6f) { }
35.        public BirdEyeCamera(GraphicsDevice graphics, float radius, float pitch) :
base(graphics)
36.        {
37.            #if WINDOWS_PHONE
38.                TranslateButton = MouseButton.Left;
39.            #else
40.                RotateButton = MouseButton.Middle;
41.                TranslateButton = MouseButton.Right;
42.            #endif
43.
44.                WheelSpeed = 1.0f;
45.                Speed = 1.0f;
46.
47.                Yaw = MathHelper.PiOver2;
48.                Radius = radius;
49.                MinRadius = 1f;
50.                MaxRadius = 500f;
51.                Pitch = pitch;
52.
53.                input = new Input();
54.                input.MouseDown += new EventHandler<MouseEventArgs>(Input_MouseDown);
55.                input.MouseMove += new EventHandler<MouseEventArgs>(Input_MouseMove);
56.                input.MouseWheel += new EventHandler<MouseEventArgs>(Input_Wheel);
57.                input.Update += new EventHandler<EventArgs>(Input_Update);
58.
```

```

59.         UpdateTransform();
60.     }
61.
62.     private void UpdateTransform()
63.     {
64.         var forward = new Vector3();
65.         forward.Y = -(float)Math.Sin(Pitch) * Radius;
66.         forward.Z = -(float)Math.Cos(Pitch) * Radius;
67.         forward.X = -(float)Math.Cos(Yaw) * forward.Z;
68.         forward.Z = -(float)Math.Sin(Yaw) * forward.Z;
69.
70.         var up = Vector3.Up;
71.         var eye = lookAt - forward;
72.         Matrix.CreateWorld(ref eye, ref forward, ref up, out transform);
73.         NotifyTransformChanged();
74.     }
75.
76.     void Input_Update(object sender, EventArgs e)
77.     {
78. #if XBOX
79.         GamePadState state = GamePad.GetState(PlayerIndex.One,
80. GamePadDeadZone.Circular);
81.         if (state.Buttons.LeftShoulder == ButtonState.Pressed)
82.             Radius -= (MaxRadius - MinRadius) * 0.005f * Speed;
83.         if (state.Buttons.RightShoulder == ButtonState.Pressed)
84.             Radius += (MaxRadius - MinRadius) * 0.005f * Speed;
85.
86.         if (Radius < MinRadius)
87.             Radius = MinRadius;
88.         else if (Radius > MaxRadius)
89.             Radius = MaxRadius;
90.
91.         float dx = -state.ThumbSticks.Right.X * Speed * 0.04f;
92.         float dz = state.ThumbSticks.Right.Y * Speed * 0.04f;
93.
94.         lookAt.X -= ((float)Math.Cos(Yaw) * dz + (float)Math.Sin(Yaw) * dx) * 0.1f;
95.         lookAt.Z -= ((float)Math.Sin(Yaw) * dz - (float)Math.Cos(Yaw) * dx) * 0.1f;
96. #endif
97.         UpdateTransform();
98.     }
99.
100.    void Input_Wheel(object sender, MouseEventArgs e)
101.    {
102.        Radius -= e.WheelDelta * (MaxRadius - MinRadius) * 0.0001f * WheelSpeed;
103.        Radius = MathHelper.Clamp(Radius, MinRadius, MaxRadius);
104.    }
105.
106.    void Input_MouseDown(object sender, MouseEventArgs e)
107.    {
108.        if (e.Button == RotateButton || e.Button == TranslateButton)
109.        {
110.            startPoint.X = e.X;
111.            startPoint.Y = e.Y;
112.        }
113.    }
114.
115.    void Input_MouseMove(object sender, MouseEventArgs e)
116.    {
117.        if (e.IsButtonDown(TranslateButton))
118.        {
119.            float dx = e.X - startPoint.X;
120.            float dz = e.Y - startPoint.Y;
121.
122.            startPoint.X = e.X;
123.            startPoint.Y = e.Y;
124.
125.            lookAt.X += ((float)Math.Cos(Yaw) * dz + (float)Math.Sin(Yaw) * dx) *
126. Speed * 0.04f;
127.            lookAt.Z += ((float)Math.Sin(Yaw) * dz - (float)Math.Cos(Yaw) * dx) *
128. Speed * 0.04f;

```


3. An Open Source XNA Engine (Nine from above) has the following (simplified) design for their cameras



We want to add additional cameras or camera features such as:

- Camera Shake
- Path Following Camera
- Pan Only Camera
- Pan Only with Zoom Camera
- AutoTargeting Camera (multiple targets)
- Google Earth Jump to camera (zooms out, pans, zooms in – much faster than panning).
- Shaking Camera with Pan Only.

Primarily, many of these classes are controllers for the camera. Restructure the classes to separate concerns of moving and controlling a camera from the camera itself (used in rendering).

4. Develop a Unit Test for the following code: