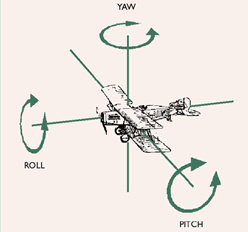
**Getting started with XNA – First Person Camera**

Posted on [**July 31, 2007**](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/) by [**Brecht Kets - Microsoft XNA MVP**](http://www.3dgameprogramming.net/author/brechtkets/)

What’s a game or an interactive presentation without a way to move around? So in this tutorial, I’ll show you how to make a first person camera. As usual, this tutorial will work on the code of the [previous one](http://www.3dgameprogramming.net/2007/07/09/getting-started-with-xna-animating-a-3d-model/), available for download [here](http://www.3dgameprogramming.net/_tutorials/XNA/AnimateModel.rar).

Let’s start with some basic concepts: yaw, pitch and roll. Yaw is when you move your head from right to left or left to right. Pitch is when you move your head up and down, and roll is when you roll your head from one shoulder to another. The illustration below shall shed some more light.



Two other concepts you need to comprehend are view and projection. The view establishes the orientation of the camera: its position, where it’s looking, and its orientation. The projection establishes how the camera projects its view onto the screen: the field of view, the aspect ratio of the viewport, and the boundaries (in distance) of what the camera can see: the near plane and the far plane.

Ok, let’s start coding. Open up your project and add a class called Camera. Make sure it inherits from DrawableGameComponent.  
public class Camera : DrawableGameComponent

First, we’re going to add a static variable, so we have easy access to the active camera. Most games don’t have just one camera, they have a first person camera, third person camera, … Note that there are better ways to do this, using a factory class and interfaces, but for this tutorial, it will do just fine.  
private static Camera activeCamera = null;

Then add a variable for our view and projection, who represent (as explained above) the orientation of the camera and the way it’s projected to the screen.  
// View and projection  
private Matrix projection = Matrix.Identity;  
private Matrix view = Matrix.Identity;

Then we’ll add some variables we’ll use to calculate our view and projection: the position of the camera, the angle of the camera, the speed of movement and the turn speed.  
//  
private Vector3 position = new Vector3(0, 0, 1000);  
private Vector3 angle = new Vector3();  
private float speed = 250f;  
private float turnSpeed = 90f;

Next we’ll add some properties, so we can access some of these variables externally.  
public static Camera ActiveCamera  
{  
get { return activeCamera; }  
set { activeCamera = value; }  
}

public Matrix Projection  
{  
get { return projection; }  
}

public Matrix View  
{  
get { return view; }  
}

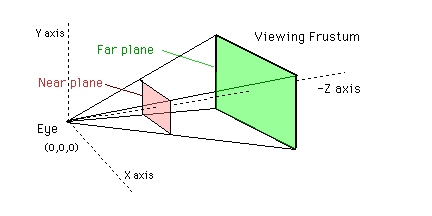
public Vector3 Position  
{  
get { return position; }  
set { position = value; }  
}

public Vector3 Angle  
{  
get { return angle; }  
set { angle = value; }  
}

Also, in the constructor, we’ll check if there is an active camera. If not, we’ll set the active camera to this camera.  
public Camera(Game game) : base(game)  
{  
if (ActiveCamera == null)  
ActiveCamera = this;  
}

When you’re in a game, you can keep turning around if u want, there is no limit to the amount of rotation. In order to achieve this, we need to keep our mouse cursor in the center of the screen. We’ll start by doing this in the Initialize method. Then we’ll calculate the movement and reset the mouse to the center of the screen each time the update method is called.  
public override void Initialize()  
{  
int centerX = Game.Window.ClientBounds.Width / 2;  
int centerY = Game.Window.ClientBounds.Height / 2;  
//  
Mouse.SetPosition(centerX, centerY);  
//  
base.Initialize();  
}

Ok, we’ve set the basis. Only two more methods to implement. In the LoadGraphicsContent method we’ll calculate our projection. In the Update method, we’ll calculate our view. First the projection:  
protected override void LoadGraphicsContent(bool loadAllContent)  
{  
float ratio = (float)GraphicsDevice.Viewport.Width / (float)GraphicsDevice.Viewport.Height;  
projection = Matrix.CreatePerspectiveFieldOfView(MathHelper.PiOver4, ratio, 10, 10000);  
//  
base.LoadGraphicsContent(loadAllContent);  
}

The projection will be a perspective field of view, with a near plane of 10 and a far plane of 10000. Below a graphic illustration.  


We’re almost there. Now to calculate our view. Start by adding the update method.  
public override void Update(GameTime gameTime)  
{  
base.Update(gameTime);  
}

First, we’ll put the time that has gone past since the previous update in a variable called delta.  
float delta = (float)gameTime.ElapsedGameTime.TotalSeconds;

And then we’ll capture the current state of the mouse and keyboard, so we’ll know which keys were pressed and where the mouse has moved to.  
KeyboardState keyboard = Keyboard.GetState();  
MouseState mouse = Mouse.GetState();

Now we have the current state, let’s just set the mouse cursor back in the center of the screen, so we don’t forget it later on.  
int centerX = Game.Window.ClientBounds.Width / 2;  
int centerY = Game.Window.ClientBounds.Height / 2;

Mouse.SetPosition(centerX, centerY);

Let’s adjust our angle variable according to the mouse movement.  
angle.X += MathHelper.ToRadians((mouse.Y - centerY) \* turnSpeed \* 0.01f); // pitch  
angle.Y += MathHelper.ToRadians((mouse.X - centerX) \* turnSpeed \* 0.01f); // yaw

Now that we have our angle, we can easily calculate our pitch and yaw vector (relative movement). I have to warn you, you’ll need your basic trigonometry, if you’re a bit rusty, fresh it up ;-). We’ll need this so we can move in the direction we’re looking at.  
Vector3 forward = Vector3.Normalize(new Vector3((float)Math.Sin(-angle.Y), (float)Math.Sin(angle.X), (float)Math.Cos(-angle.Y)));  
Vector3 left = Vector3.Normalize(new Vector3((float)Math.Cos(angle.Y), 0f, (float)Math.Sin(angle.Y)));

Ok, so let’s update our position according to the keys pressed.  
if (keyboard.IsKeyDown(Keys.Up))  
position -= forward \* speed \* delta;

if (keyboard.IsKeyDown(Keys.Down))  
position += forward \* speed \* delta;

if (keyboard.IsKeyDown(Keys.Right))  
position -= left \* speed \* delta;

if (keyboard.IsKeyDown(Keys.Left))  
position += left \* speed \* delta;

if (keyboard.IsKeyDown(Keys.PageUp))  
position += Vector3.Down \* speed \* delta;

if (keyboard.IsKeyDown(Keys.PageDown))  
position += Vector3.Up \* speed \* delta;

So, that’s it, we have all we need to calculate our view. We’ll translate to our position and multiply that by our rotations, just that simple.  
view = Matrix.Identity;  
view \*= Matrix.CreateTranslation(-position);  
view \*= Matrix.CreateRotationZ(angle.Z);  
view \*= Matrix.CreateRotationY(angle.Y);  
view \*= Matrix.CreateRotationX(angle.X);

There, that’s it, our camera class is finished. All we need to do now is add a camera to the components of our game class and tell our forklift to use the view and the projection of our camera. So put this in the Initialize method of our Game1 class.  
this.Components.Add(new Camera(this));

Finally, update the view and projection in the draw code of our forklift.  
effect.View = Camera.ActiveCamera.View;  
effect.Projection = Camera.ActiveCamera.Projection;

So, when you press F5 know, you’ll be able to move and look around. You can download the [full source](http://www.3dgameprogramming.net/_tutorials/XNA/FirstPersonCamera.rar) of this tutorial of course.

[](http://www.3dgameprogramming.net/wp-content/uploads/2007/07/firstpersoncamera.png)

[Digg This](http://digg.com/submit?url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F&title=Getting+started+with+XNA+-+First+Person+Camera)

[Reddit This](http://www.reddit.com/submit?url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F&title=Getting+started+with+XNA+-+First+Person+Camera)

[Stumble Now!](http://www.stumbleupon.com/submit?url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F&title=Getting+started+with+XNA+-+First+Person+Camera)

[Buzz This](http://buzz.yahoo.com/buzz?targetUrl=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F&headline=Getting+started+with+XNA+-+First+Person+Camera)

[Vote on DZone](http://www.dzone.com/links/add.html?title=Getting+started+with+XNA+-+First+Person+Camera&url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Share on Facebook](http://www.facebook.com/sharer.php?t=Getting+started+with+XNA+-+First+Person+Camera&u=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Bookmark this on Delicious](http://delicious.com/save?title=Getting+started+with+XNA+-+First+Person+Camera&url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Kick It on DotNetKicks.com](http://www.dotnetkicks.com/kick/?title=Getting+started+with+XNA+-+First+Person+Camera&url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Shout it](http://dotnetshoutout.com/Submit?title=Getting+started+with+XNA+-+First+Person+Camera&url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Share on LinkedIn](http://www.linkedin.com/shareArticle?mini=true&url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F&title=Getting+started+with+XNA+-+First+Person+Camera&summary=&source=)

[Bookmark this on Technorati](http://www.technorati.com/faves?add=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Post on Twitter](http://twitter.com/home?status=Reading+http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

[Google Buzz (aka. Google Reader)](http://www.google.com/buzz/post?url=http%3A%2F%2Fwww.3dgameprogramming.net%2F2007%2F07%2F31%2Fgetting-started-with-xna-first-person-camera%2F)

This entry was posted in [All](http://www.3dgameprogramming.net/category/all/), [Tutorials](http://www.3dgameprogramming.net/category/tutorials/), [XNA](http://www.3dgameprogramming.net/category/xna/) by [Brecht Kets - Microsoft XNA MVP](http://www.3dgameprogramming.net/author/brechtkets/). Bookmark the [permalink](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/).

13 thoughts on “Getting started with XNA – First Person Camera”

1. [Ian Suttle](http://www.iansuttle.com/blog) on [August 3, 2007 at 4:04 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-859) said:

I think you have a typo:

int centerX = Game.Window.ClientBounds.Width / 2;  
int centerY = Game.Window.ClientBounds.Width / 2;

centerY should be based on Height, not Width.

1. [Brecht Kets](http://www.brechtkets.net/) on [August 3, 2007 at 8:31 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-863) said:

Thanks Ian, the error has been corrected.

1. extrakun on [August 23, 2007 at 8:02 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-1573) said:

I am just interested in how you calculate your view matrix. In some textbooks I’ve checked (at least the one I use for studies), the calculation for view is yaw \* pitch \* roll, but in your case, you have roll \* yaw \* pitch.

Am I missing out something?

1. [Brecht Kets](http://www.brechtkets.net/) on [August 24, 2007 at 10:43 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-1592) said:

The only thing that matters here is that you first translate your view to your new position and after that you can apply your rotations. You can put the rotations in any order you want because I use Matrix.CreateRotationX, CreateRotationY or CreateRotationZ. So you can put it in the order you choose.  
For example:  
view = Matrix.Identity;  
view \*= Matrix.CreateTranslation(-position);  
view \*= Matrix.CreateRotationX(angle.X);  
view \*= Matrix.CreateRotationY(angle.Y);  
view \*= Matrix.CreateRotationZ(angle.Z);

Does this answer your question?

1. [Graeme Pollard](http://www.warehouse8.co.uk/) on [January 17, 2008 at 3:17 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-8124) said:

I believe your forward vector code is wrong. If you think about it when you are looking straight down the camera will go down but the yaw is going to make it move forward and back too. Therefore you need to modify those values by the cos of the pitch.

forward = Vector3.Normalize(new Vector3((float)Math.Sin(-angle.Y) \* (float)Math.Cos(angle.X), (float)Math.Sin(angle.X), (float)Math.Cos(-angle.Y) \* (float)Math.Cos(angle.X)));

It appears to work perfectly for me. Great tutorial :)If you see this please get back to me by email if I was right with that, I’d very much like to know :)

1. Nico Toscani on [February 10, 2008 at 9:32 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-8482) said:

Hello everybody.  
I have a question.I dont know how can i getting started.What should i do?I already installed Microsoft Direct X Package and Microsoft Visual C++,but i dont know what should i do.Can somebody write me,how can i getting started,please?  
Thank you

1. Nico Toscani on [February 13, 2008 at 5:23 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-8521) said:

I installed Microsoft XNA Game Studio 2.0 but i dont know where must i start!!!!Can somebody tell,what should i do now?

Thanks

1. Jonte on [February 28, 2008 at 10:13 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-8854) said:

Great tutorial!

I’m having trouble trying to bind an objects rotationmatrix to the same rotation of the camera though. I’ve tried atleast 20 different solutions but the object just doesn’t follow the direction of the view, but spins around in different directions than the one I change my camera to.

Do you think you could help me out on getting an objects rotation set to the same as the camera?

1. [XRage](http://www.tnt-squad.com/) on [March 21, 2008 at 1:11 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-9230) said:

Excellent tutorial. thanks, im doing a project for school. i need help with Planes.. I just want to know how to position one at each side of my Map, to stop people from exiting, i dont know how to place a plane. if i knew that i can do intersects, and all that. Any help by email would be greatly appreciated.

1. Adar on [September 28, 2011 at 9:08 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-13282) said:

That is a hell of a tutorial.  
The best I’ve seen….  
that was realy helpful, ty.. :)

I have a problem, I have followed your tuturial to the letter but my first person shooter appeers to be moving like an airplan, it turns on the Roll factor…

how can I nutrizllize this factor?

thenks ahead…. XD

1. [Randalthor](http://sourcecode/) on [September 3, 2012 at 7:53 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-13420) said:

The download links seems to be broken, anyone know where i can down load the source?

1. middle on [November 29, 2012 at 3:59 pm](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-13492) said:

please..i have the same problem.. download links seems to be broken, anyone know where i can down load the source? ^^  
Send me a message pleeeeease

1. Robert on [November 20, 2013 at 11:50 am](http://www.3dgameprogramming.net/2007/07/31/getting-started-with-xna-first-person-camera/comment-page-1/#comment-15033) said:

@Brecht,

Could you please upload the source code again. I’ve read a small part of your tutorials and wanted to look at the source, but as the others already said, the download links are broken.

Could you please upload these sources again?!