

Creating a Role Playing Game with XNA Game Studio 3.0

Part 34

Adding More Sprites

To follow along with this tutorial you will have to have read the previous tutorials to understand much of what it going on. You can find a list of tutorials here: [XNA 3.0 Role Playing Game Tutorials](#). You will also find the latest version of the project on the web site on that page. If you want to follow along and type in the code from this PDF as you go, you can find the previous project at this link: <http://www.jtmbooks.com/rpgtutorials/New2DRPG33.zip> You can download the graphics from this link: [Graphics.zip](#)

In this tutorial I will be adding in how to add sprites for other characters to your game. To do this you will need sprites for the other characters. I have modified many of the sprites from the game Last Guardian from EvilMana.com for this tutorial. I have made 38 sprite sheets from the sprites. You can find the sprite sheets on my web site at <http://xna.jtmbooks.com/Downloads/Sprites.zip>.

Again, these sprites are part of the creative commons license which means you are free to use them for non-commercial use. Before I go any farther since I will be working on the game in this tutorial make sure that you right click the game project and select **Set as StartUp Project**.

You will have to make one change to the **AnimatedSprite** class. Instead of making all of the sprites with the **Down** animation as the first animation I changed it so that the **Up** animation is now the first animation. The reason was it was easier for me to just change the original sprite I was using and make a quick change to the **AnimatedSprite** class. You will want to change the **AnimationKey** enum to the following.

```
public enum AnimationKey { Up, Down, Left, Right };
```

After you have downloaded the sprites and extracted them to a folder right click the **Sprites** folder in the **Content** folder and choose **Add** then **Existing Item**. Navigate to the folder that you extracted the sprites to and select all of the sprites. These are the file names of the sprites: **amg1large.png, bmg1.png, dvl1.png, ftr1.png, gsd1.png, isd1.png, jli1.png, kin1.png, knt1.png, man1.png, mnt1.png, mnv1.png, mst1.png, nja1.png, npc1.png, npc2.png, npc3.png, npc4.png, npc5.png, npc6.png, npc7.png, npc8.png, npc9.png, pdn1.png, scr1.png, skl1.png, smr1.png, spd1.png, thf1.png, trk1.png, wmg1.png, wmn1.png, wmn2.png, wmn3.png, wnv1.png, ybo1.png, ygr1.png, and zph1.png**. Many of the other sprites just have differing skin color so I didn't change them.

The rest of the work for this tutorial is going to take place in the **ActionScreen** class. The first thing that you will want to do is add in a few new fields to the class. Add them just above the constructor for the class.

```
List<AnimatedSprite> animatedSprites = new List<AnimatedSprite>();  
string[] assetNames =  
{  
    @"Sprites\knt1",  
    @"Sprites\nja1",
```

```

        @"Sprites\skill" });
Texture2D[] spriteTextures;
Random random = new Random();

```

The first new field, **animatedSprites**, is a **List<AnimatedSprite>** that will hold the sprites that will be added to the game to show that everything is working properly. The next field is called **assetNames** and is an array of strings. What I did was choose three of the sprites from the converted sprites. You can choose more or different ones. Just make sure that you include the folder name. The next field is an array of **Texture2D** called **spriteTextures** and will hold the images of the sprites. The last one is **random** in is of the class **Random**. It will be used to select where to place the sprites and which direction they will be facing.

I added a new method to the class, **CreateSprites**, and I call it from the constructor. The **CreateSprites** method has one parameter, the current **Game** object. In the **LoadContent** method I load in the textures for the sprites. All I do to load the sprites is create an array of the same length as the **assetNames** field. Then in a for loop I set the values using the **Content.Load** method passing in the value of the corresponding value in the **assetNames** field. This is the new constructor and **LoadContent** method.

```

public ActionScreen(Game game, SpriteFont gameFont, string tilesetName)
    : base(game)
{
    playerCharacter = new PlayerCharacter(game);
    this.gameFont = gameFont;

    this.tilesetName = tilesetName;
    LoadContent();
    tileMap = new TileMap(@"Content\TileSets\tilemap.tmap", tileset, game);
    Components.Add(tileMap);
    tileMap.Show();

    viewportWidth = TileEngine.ViewPortWidth;
    viewportHeight = TileEngine.ViewPortHeight;
    screenWidth = game.Window.ClientBounds.Width;
    screenHeight = game.Window.ClientBounds.Height;
    CreateSprites(game);
}

protected override void LoadContent()
{
    base.LoadContent();
    tileset = Content.Load<Tileset>(@"TileSets\" + tilesetName);
    chest = Content.Load<Texture2D>(@"Items\chest");
    characterHUDTexture = Content.Load<Texture2D>(@"Backgrounds\characterhud");
    this.interfaceFont = Content.Load<SpriteFont>(@"smallFont");

    spriteTextures = new Texture2D[assetNames.Length];
    for (int i = 0; i < assetNames.Length; i++)
        spriteTextures[i] = Content.Load<Texture2D>(assetNames[i]);
}

```

Now I will move onto the **CreateSprites** method. Some of the code will look familiar to you because it follows the code from adding a sprite to the game for the player to control. Some of it won't though. This is the code for the **CreateSprites** method.

```

private void CreateSprites(Game game)
{
    List<Animation> listAnimation = new List<Animation>();

    Animation tempAnimation = new Animation(2, 64, 64, 0, 0);
    listAnimation.Add(tempAnimation);

    tempAnimation = new Animation(2, 64, 64, 128, 0);
    listAnimation.Add(tempAnimation);

    tempAnimation = new Animation(2, 64, 64, 256, 0);
    listAnimation.Add(tempAnimation);

    tempAnimation = new Animation(2, 64, 64, 384, 0);
    listAnimation.Add(tempAnimation);

    for (int i = 0; i < assetNames.Length; i++)
    {
        List<Animation> animations = new List<Animation>();

        foreach (Animation a in listAnimation)
        {
            Animation clonedAnimation = (Animation)a.Clone();
            animations.Add(clonedAnimation);
        }

        animatedSprites.Add(
            new AnimatedSprite(game,
                spriteTextures[i],
                animations));

        animatedSprites[i].IsAnimating = true;
        int direction = random.Next(0, 4);

        switch (direction)
        {
            case 0:
                animatedSprites[i].CurrentAnimation = AnimationKey.Up;
                break;
            case 1:
                animatedSprites[i].CurrentAnimation = AnimationKey.Down;
                break;
            case 2:
                animatedSprites[i].CurrentAnimation = AnimationKey.Left;
                break;
            case 3:
                animatedSprites[i].CurrentAnimation = AnimationKey.Right;
                break;
        }

        Vector2 position = new Vector2();

        position.X = TileEngine.TileWidth * random.Next(1, 10);
        position.Y = TileEngine.TileHeight * random.Next(1, 10);

        animatedSprites[i].Position = position;
    }
}

```

```
public override void Update (GameTime gameTime)
{
    base.Update (gameTime);
    playerCharacter.Update (gameTime);
    foreach (AnimatedSprite sprite in animatedSprites)
        sprite.Update (gameTime);
}

public override void Draw (GameTime gameTime)
{
    base.Draw (gameTime);
    if (Visible)
    {
        Vector2 position = new Vector2 (0, 0);
        position.Y = viewportHeight;
        spriteBatch.Draw (characterHUDTexture, position, Color.White);
        playerCharacter.Draw (gameTime);
        foreach (AnimatedSprite sprite in animatedSprites)
            sprite.Draw (gameTime);
    }
}
```

Well that is it for this tutorial. I know that it was a little short but I feel that it was important to do. I have already started coding the next part of Eyes of the Dragon. I encourage you to keep either visiting my site <http://xna.jtmbooks.com> or my blog, <http://xna-rpg.blogspot.com> for the latest news on my tutorials.