Creating a Role Playing Game with XNA Game Studio 3.0
Part 3
Modifying the Create Character Screen

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 Role Playing Game Tutorials You will also find the latest version of the tutorial on the web site. If you want to follow along and type in the code from this PDF, you can find the previous project at this link: XNA RPG 2

In hopes on getting this project done faster, I've decided to move from a skill based character system to a class based character system. To do this, I'm going to modify the **Create Character Screen** and add the ability to change the class, or profession, of the character. There will be four professions to choose from, Fighter, Wizard, Thief or Priest.

This will be a short tutorial, as I'm just adding in selecting a profession when creating a new charcter. So, let's get started. The first thing you will want to do is open the last version of the project. I will start with the changes to **CreatePCScreen**. The first thing to do is add an array of strings for the different professions and an integer to hold the current profession index.

```
int className = 0;

string[] classNames = {
          "Fighter",
          "Wizard",
          "Thief",
          "Priest" };
```

You also have to change the menu that you pass to the **MenuComponent** of the screen. I added the new option between choosing gender and difficulty level. This is the new menu:

```
string[] menuItems = {
          "Select New Name",
          "Make Him a Woman",
          "Change Profession",
          "Change Difficulty Level",
          "Return to Start Screen",
          "Begin the Adventure" };
```

Now you have to add a method to the class to change the profession of the character. This method will be just like the other methods that change the name, gender and difficulty level. I don't think that there is anything hard about it so I'm just going to give you the method.

```
public void ChangeClass()
{
    className++;
    if (className == classNames.GetLength(0))
        className = 0;
}
```

There is one more thing that you will need to do. That is change the **Draw** method to draw the profession. All I did was change the position of where I drew the name, gender and difficulty level to make room for the profession. It is pretty much the same as the other **Draw** method. So, again, I will just give you the code. You can change the position of the text, I just liked the look of it. This is the new **Draw** method:

```
public override void Draw(GameTime gameTime)
{
    Vector2 position = new Vector2();

    position = new Vector2(150, 100 - spriteFont.LineSpacing - 5);

    spriteBatch.DrawString(spriteFont,
        "Name",
        position,
        Color.Yellow);

    position.X = 250;
    spriteBatch.DrawString(spriteFont,
        "Gender",
        position,
        Color.Yellow);

    position.X = 340;
    spriteBatch.DrawString(spriteFont,
        "Profession",
        position,
        Color.Yellow);

    position.X = 500;
    spriteBatch.DrawString(spriteFont,
        "Game Mode",
        position,
        Color.Yellow);

    position = new Vector2(150, 100);

    if (gender)
    {
        spriteBatch.DrawString(spriteFont,
            femaleNames[name],
            position,
            Color.White);

        position.X = 250;
        spriteBatch.DrawString(spriteFont,
            "Female",
            position,
            Color.White);
    }
    else
    {
        spriteBatch.DrawString(spriteFont,
            maleNames[name],
            position,
            Color.White);
```

```
            position.X = 250;
            spriteBatch.DrawString(spriteFont,
                "Male",
                position,
                Color.White);
        }

        position.X = 340;
        spriteBatch.DrawString(spriteFont,
            classNames[className],
            position,
            Color.White);

        position.X = 500;
        spriteBatch.DrawString(spriteFont,
            difficultyLevels[difficultyLevel],
            position,
            Color.White);

        base.Draw(gameTime);
    }
```

The only other thing that needs to be done is to update the **HandleCreatePCScreen**, in the **Game1** class, to reflect the changes to the **CreatePCScreen**. All that I did was update **switch** statement to call the new method to change the profession and change to order of the rest of the menu. This is the updated method:

```
private void HandleCreatePCScreenInput()
{
    if (CheckKey(Keys.Enter) || CheckKey(Keys.Space))
    {
        switch (createPCScreen.SelectedIndex)
        {
            case 0:
                createPCScreen.ChangeName();
                break;
            case 1:
                createPCScreen.ChangeGender();
                break;
            case 2:
                createPCScreen.ChangeClass();
                break;
            case 3:
                createPCScreen.ChangeDifficulty();
                break;
            case 4:
                activeScreen.Hide();
                activeScreen = startScreen;
                activeScreen.Show();
                break;
            case 5:
                activeScreen.Hide();
                activeScreen = startScreen;
                activeScreen.Show();
                break;
```

```
            }
        }
}
```

Well, that is all for this tutorial. Like I said, it is a short one. In the next tutorial I will start with creating the player character class, not to be confused with the player class that will intereact with the player. Please visit my web site again soon and hopefully I will have a new tutorial up and ready.