Eyes of the Dragon Tutorials
Part 30
Updating Weapons

I'm writing these tutorials for the MonoGame 3.8 framework using Visual Studio 2019. The tutorials will make more sense if they are read in order. You can find the list of tutorials on the Eyes of the Dragon page of my web blog. I will be making each version of the project available on GitHub here. It will be included on the page that links to the tutorials.

I wasn't going to modify weapons to use effects but in the end it makes sense to update them. It is going to break a few things so this is going to be a rather tedious tutorial. In the end it will be well worth the effort though.

The first thing I want to do is go back to have for weapon damage type crushing, piercing and slashing. I also want to add the basic magic damage type as well. Update the **DamageType** enumeration back to the following.

```
public enum DamageType { Crushing, Piercing, Slashing, Magic, Poison, Disease, Fire, Water,
Air, Earth }
```

The next step is to add in an override of the **ToString** method of **DamageEffectData**. Add this override of the **ToString** method to the **Virtual Method Region** of **DamageEffectData**.

```
public override string ToString()
{
    string toString = Name + ", " + DamageType.ToString() + ", ";

    toString += AttackType.ToString() + ", ";
    toString += DieType.ToString() + ", ";
    toString += NumberOfDice.ToString() + ", ";
    toString += Modifier.ToString();

    return toString;
}
```

Nothing hard there. I just create a local variable and build a string that represents the instance of **DamageEffectData** and then return it. The next step is to update the **WeaponData** class. You want to replace the old field related to damage with a **List<DamageEffectData>** field so that you can have weapons with multiple effects like a flaming sword that does slashing damage and fire damage. Then update the **ToString** method to use **List<DamageEffectData>** now. Update the **WeaponData** class to the following.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using RpgLibrary.EffectClasses;
namespace RpgLibrary.ItemClasses
{
    public class WeaponData
```

```
    {
        public string Name;
        public string Type;
        public int Price;
        public float Weight;
        public bool Equipped;
        public Hands NumberHands;
        public int AttackValue;
        public int AttackModifier;
        public List<DamageEffectData> DamageEffectData = new List<DamageEffectData>();
        public string[] AllowableClasses;

        public WeaponData()
        {
        }

        public override string ToString()
        {
            string toString = Name + ", ";

            toString += Type + ", ";
            toString += Price.ToString() + ", ";
            toString += Weight.ToString() + ", ";
            toString += NumberHands.ToString() + ", ";
            toString += AttackValue.ToString() + ", ";
            toString += AttackModifier.ToString();

            foreach (DamageEffectData effect in DamageEffectData)
            {
                toString += ", " + DamageEffectData.ToString();
            }

            foreach (string s in AllowableClasses)
            {
                toString += ", " + s;
            }

            return toString;
        }
    }
}
```

I replaced the **DamageValue** and **DamageModifier** fields with a **List<DamageEffectData>** field that represents the damage that the weapon does. I also update the **ToString** method to write out the **List<DamageEffectData>** field. I also added in a using statement for **EffectClasses**. The **Weapon** class took a little more work. Update the **Weapon** class to the following.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using RpgLibrary.EffectClasses;
namespace RpgLibrary.ItemClasses
{
    public class Weapon : BaseItem
    {
        #region Field Region
```

```
Hands hands;
int attackValue;
int attackModifier;
List<DamageEffectData> damageEffectDatas;

#endregion
#region Property Region

public Hands NumberHands
{
    get { return hands; }
    protected set { hands = value; }
}

public int AttackValue
{
    get { return attackValue; }
    protected set { attackValue = value; }
}

public int AttackModifier
{
    get { return attackModifier; }
    protected set { attackModifier = value; }
}

public List<DamageEffectData> DamageEffects
{
    get { return damageEffectDatas; }
    protected set { damageEffectDatas = value; }
}

#endregion

#region Constructor Region

public Weapon(
    string weaponName,
    string weaponType,
    int price,
    float weight,
    Hands hands,
    int attackValue,
    int attackModifier,
    List<DamageEffectData> damageEffectData,
    params string[] allowableClasses)
    : base(weaponName, weaponType, price, weight, allowableClasses)
{
    NumberHands = hands;
    AttackValue = attackValue;
    AttackModifier = attackModifier;
    DamageEffects = damageEffectData;
}

#endregion

#region Abstract Method Region

public override object Clone()
{
```

```
            string[] allowedClasses = new string[allowableClasses.Count];
            List<DamageEffectData> effects = new List<DamageEffectData>();

            for (int i = 0; i < allowableClasses.Count; i++)
            {
                allowedClasses[i] = allowableClasses[i];
            }

            foreach (DamageEffectData e in DamageEffects)
            {
                effects.Add(e);
            }

            Weapon weapon = new Weapon(
                Name,
                Type,
                Price,
                Weight,
                NumberHands,
                AttackValue,
                AttackModifier,
                effects,
                allowedClasses);

            return weapon;
        }

        public override string ToString()
        {
            string weaponString = base.ToString() + ", ";

            weaponString += NumberHands.ToString() + ", ";
            weaponString += AttackValue.ToString() + ", ";
            weaponString += AttackModifier.ToString();

            foreach (DamageEffectData e in DamageEffects)
            {
                weaponString += ", " + e.ToString();
            }

            foreach (string s in allowableClasses)
            {
                weaponString += ", " + s;
            }

            return weaponString;
        }

        #endregion
    }
}
```

Again there is a using statement for the **EffectClasses** namespace. I replaced the **damageValue** and **damageModifier** fields with a **List<DamageEffectData>** field. I updated the properties for the new field as well. The constructor now takes a **List<DamageEffectData>** field for damage instead of the two integer fields. It sets the fields with the values passed in. I updated the **Clone** and **ToString** methods as well to use the new field.

That ends up breaking a lot of things. It breaks the item editor and the weapons that were created. The best solution that I could come up with is to delete the weapons that were added and update the editor.

Browse to the **Weapon** folder in the **Content** folder of the project. Select all of the entries, right click on them and select **Delete**. Open the **MonoGame Pipeline Tool**. Drill down to the **Items** folder. Right click the **Weapon** folder and select **Exclude From Project**. Save the project and close the **MonoGame Pipeline Tool**.

Now, right click the **RpgEditor** project and select **Set as StartUp Project**. You will have four errors, all of them in **FormWeaponDetails**. The first step is to redesign the form. I'm going to make our life a little easier though. A weapon only causes **Weapon** damage. The will also only attack **Health**. That leaves having to change the form so you can select the **DieType**, number of dice, and modifier. The finished code from the designer appears next.

```
namespace RpgEditor
{
    partial class FormWeaponDetail
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.tbName = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.mtbPrice = new System.Windows.Forms.MaskedTextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.nudWeight = new System.Windows.Forms.NumericUpDown();
            this.label5 = new System.Windows.Forms.Label();
            this.cboHands = new System.Windows.Forms.ComboBox();
            this.label6 = new System.Windows.Forms.Label();
```

```
this.mtbAttackValue = new System.Windows.Forms.MaskedTextBox();
this.label11 = new System.Windows.Forms.Label();
this.mtbAttackModifier = new System.Windows.Forms.MaskedTextBox();
this.lbClasses = new System.Windows.Forms.ListBox();
this.label9 = new System.Windows.Forms.Label();
this.lbAllowedClasses = new System.Windows.Forms.ListBox();
this.label10 = new System.Windows.Forms.Label();
this.btnMoveAllowed = new System.Windows.Forms.Button();
this.btnRemoveAllowed = new System.Windows.Forms.Button();
this.btnOK = new System.Windows.Forms.Button();
this.btnCancel = new System.Windows.Forms.Button();
this.lblDamageEffects = new System.Windows.Forms.Label();
this.lbDamageEffects = new System.Windows.Forms.ListBox();
this.btnAdd = new System.Windows.Forms.Button();
this.btnEdit = new System.Windows.Forms.Button();
this.btnDelete = new System.Windows.Forms.Button();
this.groupBox1 = new System.Windows.Forms.GroupBox();
this.label7 = new System.Windows.Forms.Label();
this.cboDieType = new System.Windows.Forms.ComboBox();
this.label8 = new System.Windows.Forms.Label();
this.nudDice = new System.Windows.Forms.NumericUpDown();
this.mtbModifier = new System.Windows.Forms.MaskedTextBox();
this.label12 = new System.Windows.Forms.Label();
this.cboDamageType = new System.Windows.Forms.ComboBox();
this.label13 = new System.Windows.Forms.Label();
this.cboAttackType = new System.Windows.Forms.ComboBox();
((System.ComponentModel.ISupportInitialize)(this.nudWeight)).BeginInit();
this.groupBox1.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.nudDice)).BeginInit();
this.SuspendLayout();
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(106, 20);
this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(55, 20);
this.label1.TabIndex = 0;
this.label1.Text = "Name:";
//
// tbName
//
this.tbName.Location = new System.Drawing.Point(172, 15);
this.tbName.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.tbName.Name = "tbName";
this.tbName.Size = new System.Drawing.Size(148, 26);
this.tbName.TabIndex = 0;
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(112, 60);
this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(47, 20);
this.label2.TabIndex = 2;
this.label2.Text = "Type:";
//
// label3
```

```
            //
            this.label3.AutoSize = true;
            this.label3.Location = new System.Drawing.Point(112, 100);
            this.label3.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(48, 20);
            this.label3.TabIndex = 4;
            this.label3.Text = "Price:";
            //
            // mtbPrice
            //
            this.mtbPrice.Location = new System.Drawing.Point(172, 95);
            this.mtbPrice.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
            this.mtbPrice.Mask = "000000";
            this.mtbPrice.Name = "mtbPrice";
            this.mtbPrice.Size = new System.Drawing.Size(148, 26);
            this.mtbPrice.TabIndex = 2;
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(98, 138);
            this.label4.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(63, 20);
            this.label4.TabIndex = 6;
            this.label4.Text = "Weight:";
            //
            // nudWeight
            //
            this.nudWeight.DecimalPlaces = 2;
            this.nudWeight.Location = new System.Drawing.Point(172, 135);
            this.nudWeight.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
            this.nudWeight.Name = "nudWeight";
            this.nudWeight.Size = new System.Drawing.Size(150, 26);
            this.nudWeight.TabIndex = 3;
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.Location = new System.Drawing.Point(102, 182);
            this.label5.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(60, 20);
            this.label5.TabIndex = 8;
            this.label5.Text = "Hands:";
            //
            // cboHands
            //
            this.cboHands.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.cboHands.FormattingEnabled = true;
            this.cboHands.Location = new System.Drawing.Point(172, 177);
            this.cboHands.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
            this.cboHands.Name = "cboHands";
            this.cboHands.Size = new System.Drawing.Size(148, 28);
            this.cboHands.TabIndex = 4;
            //
            // label6
            //
            this.label6.AutoSize = true;
```

```
this.label6.Location = new System.Drawing.Point(57, 223);
this.label6.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(104, 20);
this.label6.TabIndex = 10;
this.label6.Text = "Attack Value:";
//
// mtbAttackValue
//
this.mtbAttackValue.Location = new System.Drawing.Point(172, 218);
this.mtbAttackValue.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.mtbAttackValue.Mask = "000";
this.mtbAttackValue.Name = "mtbAttackValue";
this.mtbAttackValue.Size = new System.Drawing.Size(148, 26);
this.mtbAttackValue.TabIndex = 5;
//
// label11
//
this.label11.AutoSize = true;
this.label11.Location = new System.Drawing.Point(42, 263);
this.label11.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label11.Name = "label11";
this.label11.Size = new System.Drawing.Size(119, 20);
this.label11.TabIndex = 10;
this.label11.Text = "Attack Modifier:";
//
// mtbAttackModifier
//
this.mtbAttackModifier.Location = new System.Drawing.Point(172, 258);
this.mtbAttackModifier.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.mtbAttackModifier.Mask = "000";
this.mtbAttackModifier.Name = "mtbAttackModifier";
this.mtbAttackModifier.Size = new System.Drawing.Size(148, 26);
this.mtbAttackModifier.TabIndex = 6;
//
// lbClasses
//
this.lbClasses.FormattingEnabled = true;
this.lbClasses.ItemHeight = 20;
this.lbClasses.Location = new System.Drawing.Point(357, 55);
this.lbClasses.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.lbClasses.Name = "lbClasses";
this.lbClasses.Size = new System.Drawing.Size(178, 224);
this.lbClasses.TabIndex = 14;
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(357, 18);
this.label9.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(143, 20);
this.label9.TabIndex = 13;
this.label9.Text = "Character Classes:";
//
// lbAllowedClasses
//
this.lbAllowedClasses.FormattingEnabled = true;
this.lbAllowedClasses.ItemHeight = 20;
this.lbAllowedClasses.Location = new System.Drawing.Point(668, 55);
```

```
this.lbAllowedClasses.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.lbAllowedClasses.Name = "lbAllowedClasses";
this.lbAllowedClasses.Size = new System.Drawing.Size(178, 224);
this.lbAllowedClasses.TabIndex = 15;
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new System.Drawing.Point(663, 18);
this.label10.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(128, 20);
this.label10.TabIndex = 13;
this.label10.Text = "Allowed Classes:";
//
// btnMoveAllowed
//
this.btnMoveAllowed.Location = new System.Drawing.Point(546, 123);
this.btnMoveAllowed.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.btnMoveAllowed.Name = "btnMoveAllowed";
this.btnMoveAllowed.Size = new System.Drawing.Size(112, 35);
this.btnMoveAllowed.TabIndex = 10;
this.btnMoveAllowed.Text = ">>";
this.btnMoveAllowed.UseVisualStyleBackColor = true;
//
// btnRemoveAllowed
//
this.btnRemoveAllowed.Location = new System.Drawing.Point(546, 189);
this.btnRemoveAllowed.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.btnRemoveAllowed.Name = "btnRemoveAllowed";
this.btnRemoveAllowed.Size = new System.Drawing.Size(112, 35);
this.btnRemoveAllowed.TabIndex = 11;
this.btnRemoveAllowed.Text = "<<";
this.btnRemoveAllowed.UseVisualStyleBackColor = true;
//
// btnOK
//
this.btnOK.Location = new System.Drawing.Point(867, 11);
this.btnOK.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.btnOK.Name = "btnOK";
this.btnOK.Size = new System.Drawing.Size(112, 35);
this.btnOK.TabIndex = 12;
this.btnOK.Text = "OK";
this.btnOK.UseVisualStyleBackColor = true;
//
// btnCancel
//
this.btnCancel.Location = new System.Drawing.Point(869, 56);
this.btnCancel.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
this.btnCancel.Name = "btnCancel";
this.btnCancel.Size = new System.Drawing.Size(112, 35);
this.btnCancel.TabIndex = 13;
this.btnCancel.Text = "Cancel";
this.btnCancel.UseVisualStyleBackColor = true;
//
// lblDamageEffects
//
this.lblDamageEffects.AutoSize = true;
this.lblDamageEffects.Location = new System.Drawing.Point(33, 292);
this.lblDamageEffects.Name = "lblDamageEffects";
```

```
this.lblDamageEffects.Size = new System.Drawing.Size(129, 20);
this.lblDamageEffects.TabIndex = 16;
this.lblDamageEffects.Text = "Damage Effects:";
//
// lbDamageEffects
//
this.lbDamageEffects.FormattingEnabled = true;
this.lbDamageEffects.ItemHeight = 20;
this.lbDamageEffects.Location = new System.Drawing.Point(172, 292);
this.lbDamageEffects.Name = "lbDamageEffects";
this.lbDamageEffects.Size = new System.Drawing.Size(363, 184);
this.lbDamageEffects.TabIndex = 17;
//
// btnAdd
//
this.btnAdd.Location = new System.Drawing.Point(378, 482);
this.btnAdd.Name = "btnAdd";
this.btnAdd.Size = new System.Drawing.Size(75, 35);
this.btnAdd.TabIndex = 18;
this.btnAdd.Text = "Add";
this.btnAdd.UseVisualStyleBackColor = true;
//
// btnEdit
//
this.btnEdit.Location = new System.Drawing.Point(459, 482);
this.btnEdit.Name = "btnEdit";
this.btnEdit.Size = new System.Drawing.Size(75, 35);
this.btnEdit.TabIndex = 19;
this.btnEdit.Text = "Edit";
this.btnEdit.UseVisualStyleBackColor = true;
//
// btnDelete
//
this.btnDelete.Location = new System.Drawing.Point(540, 482);
this.btnDelete.Name = "btnDelete";
this.btnDelete.Size = new System.Drawing.Size(75, 35);
this.btnDelete.TabIndex = 20;
this.btnDelete.Text = "Delete";
this.btnDelete.UseVisualStyleBackColor = true;
//
// groupBox1
//
this.groupBox1.Controls.Add(this.label13);
this.groupBox1.Controls.Add(this.cboDamageType);
this.groupBox1.Controls.Add(this.label12);
this.groupBox1.Controls.Add(this.mtbModifier);
this.groupBox1.Controls.Add(this.nudDice);
this.groupBox1.Controls.Add(this.label8);
this.groupBox1.Controls.Add(this.cboDieType);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Location = new System.Drawing.Point(540, 292);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(306, 184);
this.groupBox1.TabIndex = 21;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Damage Effect";
//
// label7
//
this.label7.AutoSize = true;
```

```
this.label7.Location = new System.Drawing.Point(72, 34);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(75, 20);
this.label7.TabIndex = 0;
this.label7.Text = "Die Type:";
//
// cboDieType
//
this.cboDieType.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cboDieType.FormattingEnabled = true;
this.cboDieType.Location = new System.Drawing.Point(159, 31);
this.cboDieType.Name = "cboDieType";
this.cboDieType.Size = new System.Drawing.Size(121, 28);
this.cboDieType.TabIndex = 1;
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(63, 67);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(84, 20);
this.label8.TabIndex = 2;
this.label8.Text = "Die Count:";
//
// nudDice
//
this.nudDice.Location = new System.Drawing.Point(160, 65);
this.nudDice.Minimum = new decimal(new int[] {
1,
0,
0,
0});
this.nudDice.Name = "nudDice";
this.nudDice.Size = new System.Drawing.Size(120, 26);
this.nudDice.TabIndex = 3;
this.nudDice.Value = new decimal(new int[] {
1,
0,
0,
0});
//
// mtbModifier
//
this.mtbModifier.Location = new System.Drawing.Point(160, 97);
this.mtbModifier.Mask = "00";
this.mtbModifier.Name = "mtbModifier";
this.mtbModifier.Size = new System.Drawing.Size(121, 26);
this.mtbModifier.TabIndex = 4;
//
// label12
//
this.label12.AutoSize = true;
this.label12.Location = new System.Drawing.Point(78, 100);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(69, 20);
this.label12.TabIndex = 5;
this.label12.Text = "Modifier:";
//
// cboDamageType
//
```

```
this.cboDamageType.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cboDamageType.FormattingEnabled = true;
this.cboDamageType.Location = new System.Drawing.Point(160, 129);
this.cboDamageType.Name = "cboDamageType";
this.cboDamageType.Size = new System.Drawing.Size(121, 28);
this.cboDamageType.TabIndex = 6;
//
// label13
//
this.label13.AutoSize = true;
this.label13.Location = new System.Drawing.Point(42, 132);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(112, 20);
this.label13.TabIndex = 7;
this.label13.Text = "Damage Type:";
//
// cboAttackType
//
this.cboAttackType.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cboAttackType.FormattingEnabled = true;
this.cboAttackType.Location = new System.Drawing.Point(172, 59);
this.cboAttackType.Name = "cboAttackType";
this.cboAttackType.Size = new System.Drawing.Size(148, 28);
this.cboAttackType.TabIndex = 22;
//
// FormWeaponDetail
//
this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(992, 540);
this.ControlBox = false;
this.Controls.Add(this.cboAttackType);
this.Controls.Add(this.groupBox1);
this.Controls.Add(this.btnDelete);
this.Controls.Add(this.btnEdit);
this.Controls.Add(this.btnAdd);
this.Controls.Add(this.lbDamageEffects);
this.Controls.Add(this.lblDamageEffects);
this.Controls.Add(this.btnCancel);
this.Controls.Add(this.btnOK);
this.Controls.Add(this.btnRemoveAllowed);
this.Controls.Add(this.btnMoveAllowed);
this.Controls.Add(this.label10);
this.Controls.Add(this.label9);
this.Controls.Add(this.lbAllowedClasses);
this.Controls.Add(this.lbClasses);
this.Controls.Add(this.mtbAttackModifier);
this.Controls.Add(this.label11);
this.Controls.Add(this.mtbAttackValue);
this.Controls.Add(this.label6);
this.Controls.Add(this.cboHands);
this.Controls.Add(this.label5);
this.Controls.Add(this.nudWeight);
this.Controls.Add(this.label4);
this.Controls.Add(this.mtbPrice);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.tbName);
this.Controls.Add(this.label1);
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;
```

```
            this.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
            this.Name = "FormWeaponDetail";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "Weapon Details";
            ((System.ComponentModel.ISupportInitialize)(this.nudWeight)).EndInit();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.nudDice)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox tbName;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.MaskedTextBox mtbPrice;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.NumericUpDown nudWeight;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.ComboBox cboHands;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.MaskedTextBox mtbAttackValue;
        private System.Windows.Forms.Label label11;
        private System.Windows.Forms.MaskedTextBox mtbAttackModifier;
        private System.Windows.Forms.ListBox lbClasses;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.ListBox lbAllowedClasses;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Button btnMoveAllowed;
        private System.Windows.Forms.Button btnRemoveAllowed;
        private System.Windows.Forms.Button btnOK;
        private System.Windows.Forms.Button btnCancel;
        private System.Windows.Forms.Label lblDamageEffects;
        private System.Windows.Forms.ListBox lbDamageEffects;
        private System.Windows.Forms.Button btnAdd;
        private System.Windows.Forms.Button btnEdit;
        private System.Windows.Forms.Button btnDelete;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label label12;
        private System.Windows.Forms.MaskedTextBox mtbModifier;
        private System.Windows.Forms.NumericUpDown nudDice;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.ComboBox cboDieType;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.ComboBox cboDamageType;
        private System.Windows.Forms.ComboBox cboAttackType;
    }
}
```

The next step will be to add the code for the form. Bring up the code for **FormWeaponDetails** and update it to the following.

```
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using RpgLibrary;
using RpgLibrary.EffectClasses;
using RpgLibrary.ItemClasses;

namespace RpgEditor
{
    public partial class FormWeaponDetail : Form
    {
        #region Field Region

        WeaponData weapon = null;
        List<DamageEffectData> damageEffects = new List<DamageEffectData>();

        #endregion

        #region Property Region

        public WeaponData Weapon
        {
            get { return weapon; }
            set { weapon = value; }
        }

        #endregion

        #region Constructor Region

        public FormWeaponDetail()
        {
            InitializeComponent();

            this.Load += new EventHandler(FormWeaponDetails_Load);
            this.FormClosing += new FormClosingEventHandler(FormWeaponDetails_FormClosing);

            btnMoveAllowed.Click += new EventHandler(btnMoveAllowed_Click);
            btnRemoveAllowed.Click += new EventHandler(btnRemoveAllowed_Click);

            btnAdd.Click += BtnAdd_Click;
            btnEdit.Click += BtnEdit_Click;
            btnDelete.Click += BtnDelete_Click;

            lbDamageEffects.SelectedIndexChanged += LbDamageEffects_SelectedIndexChanged;

            btnOK.Click += new EventHandler(btnOK_Click);
            btnCancel.Click += new EventHandler(btnCancel_Click);
        }

        private void LbDamageEffects_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        private void BtnAdd_Click(object sender, EventArgs e)
        {
            DamageEffectData effect = new DamageEffectData
```

```
    {
        AttackType = (AttackType)Enum.Parse(
            typeof(AttackType),
            cboAttackType.Text),
        DamageType = (DamageType)Enum.Parse(
            typeof(DamageType),
            cboDamageType.SelectedItem.ToString()),
        NumberOfDice = (int)nudDice.Value,
        Modifier = int.Parse(mtbModifier.Text),
        DieType = (DieType)Enum.Parse(
            typeof(DieType),
            cboDieType.Text)
    };

    damageEffects.Add(effect);
    lbDamageEffects.Items.Add(effect);
}

private void BtnEdit_Click(object sender, EventArgs e)
{
}

private void BtnDelete_Click(object sender, EventArgs e)
{
    if (lbDamageEffects.SelectedIndex < 0)
    {
        return;
    }

    damageEffects.RemoveAt(lbDamageEffects.SelectedIndex);
    lbDamageEffects.Items.RemoveAt(lbDamageEffects.SelectedIndex);
}

#endregion

#region Event Handler Region
void FormWeaponDetails_Load(object sender, EventArgs e)
{
    foreach (string s in FormDetails.EntityDataManager.EntityData.Keys)
    {
        lbClasses.Items.Add(s);
    }

    foreach (AttackType attackType in Enum.GetValues(typeof(AttackType)))
    {
        cboAttackType.Items.Add(attackType);
    }

    foreach (Hands location in Enum.GetValues(typeof(Hands)))
    {
        cboHands.Items.Add(location);
    }

    foreach (DieType die in Enum.GetValues(typeof(DieType)))
    {
        cboDieType.Items.Add(die);
    }

    foreach (DamageType d in Enum.GetValues(typeof(DamageType)))
    {
```

```
            cboDamageType.Items.Add(d);
        }

        cboHands.SelectedIndex = 0;
        cboDieType.SelectedIndex = 0;
        cboDieType.SelectedValue = Enum.GetName(typeof(DieType), DieType.D4);
        if (weapon != null)
        {
            tbName.Text = weapon.Name;
            cboAttackType.Text = weapon.Type;
            mtbPrice.Text = weapon.Price.ToString();
            nudWeight.Value = (decimal)weapon.Weight;
            cboHands.SelectedIndex = (int)weapon.NumberHands;
            mtbAttackValue.Text = weapon.AttackValue.ToString();
            mtbAttackModifier.Text = weapon.AttackModifier.ToString();

            foreach (DamageEffectData effect in weapon.DamageEffectData)
            {
                lbDamageEffects.Items.Add(effect);
                damageEffects.Add(effect);
            }

            foreach (string s in weapon.AllowableClasses)
            {
                if (lbClasses.Items.Contains(s))
                    lbClasses.Items.Remove(s);

                lbAllowedClasses.Items.Add(s);
            }
        }
    }

    void FormWeaponDetails_FormClosing(object sender, FormClosingEventArgs e)
    {
        if (e.CloseReason == CloseReason.UserClosing)
        {
            e.Cancel = true;
        }
    }

    void btnMoveAllowed_Click(object sender, EventArgs e)
    {
        if (lbClasses.SelectedItem != null)
        {
            lbAllowedClasses.Items.Add(lbClasses.SelectedItem);
            lbClasses.Items.RemoveAt(lbClasses.SelectedIndex);
        }
    }

    void btnRemoveAllowed_Click(object sender, EventArgs e)
    {

        if (lbAllowedClasses.SelectedItem != null)
        {
            lbClasses.Items.Add(lbAllowedClasses.SelectedItem);
            lbAllowedClasses.Items.RemoveAt(lbAllowedClasses.SelectedIndex);
        }
    }

    void btnOK_Click(object sender, EventArgs e)
```

```
{
    int price = 0;
    float weight = 0f;
    int attVal = 0;
    int attMod = 0;
    int damVal = 0;
    int damMod = 0;

    if (string.IsNullOrEmpty(tbName.Text))
    {
        MessageBox.Show("You must enter a name for the item.");
        return;
    }

    if (!int.TryParse(mtbPrice.Text, out price))
    {
        MessageBox.Show("Price must be an integer value.");
        return;
    }

    weight = (float)nudWeight.Value;

    if (!int.TryParse(mtbAttackValue.Text, out attVal))
    {
        MessageBox.Show("Attack value must be an interger value.");
        return;
    }

    if (!int.TryParse(mtbAttackModifier.Text, out attMod))
    {
        MessageBox.Show("Attack value must be an interger value.");
        return;
    }

    if (damageEffects.Count == 0)
    {
        MessageBox.Show("Weapons require at least one damage effect.");
        return;
    }

    List<string> allowedClasses = new List<string>();
    List<DamageEffectData> effects = new List<DamageEffectData>();

    foreach (object o in lbAllowedClasses.Items)
    {
        allowedClasses.Add(o.ToString());
    }

    weapon = new WeaponData
    {
        Name = tbName.Text,
        Type = cboAttackType.Text,
        NumberHands = (Hands)Enum.Parse(typeof(Hands), cboHands.Text),
        Price = price,
        Weight = weight,
        AttackValue = attVal,
        AttackModifier = attMod,
        AllowableClasses = allowedClasses.ToArray(),
        DamageEffectData = damageEffects
    };
```

```
            this.FormClosing -= FormWeaponDetails_FormClosing;
            this.Close();
        }

        void btnCancel_Click(object sender, EventArgs e)
        {
            weapon = null;

            this.FormClosing -= FormWeaponDetails_FormClosing;
            this.Close();
        }

        #endregion
    }
}
```

Most of the code is the same but enough of it changed that I wanted to give you the code for it all. The first change was I added a **List<DamageEffectData>** that holds the damage that the weapon does. Next in the constructor I wire event handlers for adding a damage effect, deleting a damage effect and editing a damage effect.

There changes to the **FormWeaponDetails_Load** method. The first new code is that I add all of the values in the values in the **AttackType** enumeration to the **cboAttackType** combo box. I also add of the values of **DieType** enumeration to **cboDieType**. What is different is that you can't just cast **DieType** to an integer for the **SelectedIndex** property of **cboDieType** because **DieType** has values associated with each member of the enumeration. For example, 4 is associated with **D4** instead of 0. For that reason I set the **SelectedValue** property of **cboDieType** to the name of **DieType.D4**. I get the name using the **GetName** method passing in the type and the value. I also add all of the damage types to **cboDamageType** the same way as the other combo boxes.

The next change is in the if statement where I check to see if **weapon** is not null. I now loop over the **DamageEffectData** collection of the weapon. Inside the loop I add the effect to the list box and to the field that I added.

I also had to modify **btnOK_Click** because **WeaponData** changed as well as the controls on the form. I of course removed everything that had to do with damage to use **DamageEffectData**. The verifying of values on controls is like before. The new part is creating a new **WeaponData** object. You have to set the fields of **DamageEffectData**. I set the **Name** to be the **Text** property of **tbName** as weapons will be unique so the **DamageEffectData** associated with weapons will also be unique by name. For now I'm assuming that the **AttackType** will be **AttackType.Health** and that the **DamageType** will be **DamageType.Weapon**. You could easily have controls on the form to select these. The hard part was setting the **DieType** field. I used the **Enum.Parse** method which parses a string to be the associated value of an enumeration. For the type you pass in **typeof(DieType)** and for the string I passed in the item at **SelectedIndex** and the **ToString** method. I then set the **NumberOfDice** and **Modifier** fields using the **Value** property of **nudDice** and **damMod** variable respectively.

In the event handler for the **Click** event of **btnAdd** I create a new **DamageEffectData** use the values from the form. I have to convert the string values of the combo boxes to enums using the **Parse** method.

For now the **Click** event for **btnEdit** is wired but not functional. I will add that functionality in a different tutorial. The **Click** event handler for **btnDelete** checks to see if the **SelectedIndex** property of **lbDamageEffects** is less than zero. If it is I exit the method. If it is not I remove the selected index from the list of damage effects and **lbDamageEffects**. It is because you can add and delete that I didn't add the edit functionality. You can always remove the item and add it back in.

If you build and run the editor now things will work. You will be able to create weapons, save them and read them back in like before. I have added all of the game content I have created to my blog. You can download it from the link below: https://cynthiamcmahon.ca/blog/downloads/tutorial30.zip.

The last thing that you want to do for this tutorial is to add the new data to the project for the game. Open up the directory where your **Game** folder is in Windows Explorer. Now drag the **Game** folder from Windows Explorer onto the **Content** folder in your game. Open the **MonoGame Pipeline Tool**. If you drill down to the **Items** folder. Right click the Items folder, select **Add** and then **Existing Folder**. Select the **Weapon** folder and add it.

That's it for this tutorial. It covers the original tutorial with major modifications. So, I encourage you to visit my blog at, https://cynthiamcmahon.ca/blog/, for the latest news on my tutorials and other goodness.

Good luck in your Game Programming Adventures!

Cynthia