Dragon's Claw – Web Tutorial Series
Chapter 2
Empires

This tutorial series is about creating a sci-fi themed text based strategic role playing game. The tutorials will make more sense if you read them in order as each tutorial builds on the previous tutorials. You can find the list of tutorials on my blog: Dragon's Claw. The source code for the tutorial series is available as a Git repository. I will be using Visual Studio 2019 Community for the series. The code should compile on the 2015 and 2017 versions as well.

I want to mention though that the series is released as Creative Commons 3.0 Attribution. It means that you are free to use any of the code or graphics in your own game, even for commercial use, with attribution. Just give credit to Cynthia McMahon and add a link to my site, https://mygameprogrammingadventures.blogspot,com. Screenshots of your project and/or a video of game play would be appreciated.

I also want to mention that I assume you have a basic understanding of C#, SQL, HTML, CSS, Javascript and jQuery. If you don't I recommend that you learn basic of each. Enough to be familiar with the syntax and how to read code.

There is something that is bugging me from the last tutorial. In my game I used Font Awesome instead of v for expanding panels and ^ for collapsing panels. I can't package Font Awesome in my source but since glyphicons came in the default project I will use them. Add the following code to the dragons-claw.less file to create a font-family and some CSS for the up and down chevrons.

```
/* Glyphicons */
@font-face {
    font-family: "Glyphicons Halflings";
    src: url("../fonts/glyphicons-halflings-regular.eot");
    src: url("../fonts/glyphicons-halflings-regular.eot?#iefix") format("embedded-opentype"),
url("../fonts/glyphicons-halflings-regular.woff2") format("woff2"), url("../fonts/glyphicons-
halflings-regular.woff") format("woff"), url("../fonts/glyphicons-halflings-regular.ttf")
format("truetype"), url("../fonts/glyphicons-halflings-
regular.svg#glyphicons_halflingsregular") format("svg");
}

.glyphicon {
    position: relative;
    top: 1px;
    display: inline-block;
    font-family: "Glyphicons Halflings";
    font-style: normal;
    font-weight: 100;
    line-height: 1;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}

.glyphicon-chevron-up:before {
    content: "\e113";
}

.glyphicon-chevron-down:before {
    content: "\e114";
}
```

The CSS came from Bootstrap. It creates a font-family from the glyphicon fonts that are included with Bootstrap. There is a class for the fonts. It has position relative so we can use :before. It has a top of 1px. Display is set to inline-block so everything will display inline. The font-family is set to Glyphicons Halflings that we just created. The font style is normal and the font-weight is 100. Line-height is 1. There are so fillers for -webkit and -moz-osx. I added two other classes with :before to display specific characters.

Now I'm going to update the Index view of the Game controller to use the new font. Replace the Index.cshtml in the Game folder with the following.

```
@model DragonsClaw.Models.PlayerViewModel
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_GameLayout.cshtml";
}
@Html.Partial("_PlayerHeader", Model)
<div id="game-container">
    <div id="star-date">Star Date 23000</div>
    <div id="bot">
        <div id="bot-menu" class="nav-bar">
            <span data-spec="The Bot pane allows you to interact with the bot."
tabindex="0">Bot</span>
            <span style="float: right;">
                <a href="#!" onclick="botHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="botShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="bot-content" class="content">
        </div>
    </div>
    <div id="high-council">
        <div id="high-council-menu" class="nav-bar">
            <span data-spec="The High Council pane contains useful information about your
sector." tabindex="0">High Council</span>
            <span style="float: right;">
                <a href="#!" onclick="highCouncilHide();" class="nav-link"><span
class="glyphicon glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="highCouncilShow();" class="nav-link"><span
class="glyphicon glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="high-council-content" class="content">
        </div>
    </div>
    <div id="empire">
        <div id="empire-menu" class="nav-bar">
            <span data-spec="The Empire pane contains information about your
empire.">Empire</span>
            <span style="float: right;">
```

```
                <a href="#!" onclick="empireHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="empireShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="empire-content" class="content">
        </div>
    </div>
    <div id="news">
        <div id="news-menu" class="nav-bar">
            <span data-spec="The News pane contains information about events in your
sector.">News</span>
            <span style="float: right;">
                <a href="#!" onclick="newsHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="newsShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="news-content" class="content">
        </div>
    </div>
    <div id="empire-news">
        <div id="empire-news-menu" class="nav-bar">
            <span data-spec="The Empire News pane contains information about events in your
empire.">Empire News</span>
            <span style="float: right;">
                <a href="#!" onclick="empirenewsHide();" class="nav-link"><span
class="glyphicon glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="empirenewsShow();" class="nav-link"><span
class="glyphicon glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="empire-news-content" class="content">
        </div>
    </div>
    <div id="exploration">
        <div id="exploration-menu" class="nav-bar">
            <span data-spec="The Exploration pane allows you to expand the borders of your
sector by exploring for new systems.">Exploration</span>
            <span style="float: right;">
                <a href="#!" onclick="explorationHide();" class="nav-link"><span
class="glyphicon glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="explorationShow();" class="nav-link"><span
class="glyphicon glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="exploration-content" class="content">
        </div>
    </div>
    <div id="expansion">
        <div id="expansion-menu" class="nav-bar">
            <span data-spec="The Expansion pane grows your sector by devoting planets to
different specialties.">Expansion</span>
            <span style="float: right;">
                <a href="#!" onclick="expansionHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="expansionShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
```

```
            </span>
        </div>
        <div id="expansion-content" class="content">
        </div>
    </div>
    <div id="technology">
        <div id="technology-menu" class="nav-bar">
            <span data-spec="The Technology pane allows you to research technology that can
improve the effectiveness of different areas.">Technology</span>
            <span style="float: right;">
                <a href="#!" onclick="technologyHide();" class="nav-link"><span
class="glyphicon glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="technologyShow();" class="nav-link"><span
class="glyphicon glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="technology-content" class="content">
        </div>
    </div>
    <div id="military">
        <div id="military-menu" class="nav-bar">
            <span data-spec="The Military pane allows you to train your military for conquest
and defense.">Military</span>
            <span style="float: right;">
                <a href="#!" onclick="militaryHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="militaryShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="military-content" class="content">
        </div>
    </div>
    <div id="telepaths">
        <div id="telepaths-menu" class="nav-bar">
            <span data-spec="The Telepath is used to launch psychic attacks against enemy
sectors.">Telepaths</span>
            <span style="float: right;">
                <a href="#!" onclick="telepathsHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="telepathsShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="telepaths-content" class="content">
        </div>
    </div>
    <div id="psionics">
        <div id="psionics-menu" class="nav-bar">
            <span data-spec="The Psionics pane allows you to use psionic abilities on your
sector.">Psionics</span>
            <span style="float: right;">
                <a href="#!" onclick="psionicsHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="psionicsShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="psionics-content" class="content">
        </div>
    </div>
    <div id="war-advisors">
        <div id="war-advisor-menu" class="nav-bar">
            <span data-spec="The War Advisors pane aids in your conquest of the galaxy.">War
Advisors</span>
            <span style="float: right;">
                <a href="#!" onclick="warAdvisorHide();" class="nav-link"><span
class="glyphicon glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="warAdvisorShow();" class="nav-link"><span
class="glyphicon glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="war-advisor-content" class="content">
        </div>
    </div>
    <div id="espionage">
        <div id="espionage-menu" class="nav-bar">
            <span data-spec="The Espionage pane is where you gather intelligence on other
sectors.">Espionage</span>
            <span style="float: right;">
                <a href="#!" onclick="espionageHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="espionageShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="espionage-content" class="content">
        </div>
    </div>
    <div id="raider">
        <div id="raider-menu" class="nav-bar">
            <span data-spec="The Raiding pane allows you to take through subtly what you can't
through force.">Raiding</span>
            <span style="float: right;">
                <a href="#!" onclick="raiderHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="raiderShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="raider-content" class="content">
        </div>
    </div>
    <div id="array">
        <div id="array-menu" class="nav-bar">
            <span data-spec="The Arrays pane allows for the construction of arrays benificial
to your empire.">Arrays</span>
            <span style="float: right;">
                <a href="#!" onclick="arrayHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="arrayShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="array-content" class="content">
        </div>
    </div>
    <div id="platform">
        <div id="platform-menu" class="nav-bar">
            <span data-spec="The Platforms pane allows for the construction of platforms to
```

```
hinder your opponents.">Platforms</span>
            <span style="float: right;">
                <a href="#!" onclick="platformHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="platformShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="platform-content" class="content">
        </div>
    </div>
    <div id="aid">
        <div id="aid-menu" class="nav-bar">
            <span data-spec="The Assistance pane allows you to assist other sectors in your
empire.">Assistance</span>
            <span style="float: right;">
                <a href="#!" onclick="aidHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="aidShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="aid-content" class="content">
        </div>
    </div>
    <div id="market">
        <div id="market-menu" class="nav-bar">
            <span data-spec="The Marketplace pane allows you to buy and sell resources on the
galatic market.">Marketplace</span>
            <span style="float: right;">
                <a href="#!" onclick="marketHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="marketShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="market-content" class="content">
        </div>
    </div>
    <div id="quests">
        <div id="quests-menu" class="nav-bar">
            <span>Quests</span>
            <span style="float: right;">
                <a href="#!" onclick="questsHide();" class="nav-link"><span class="glyphicon
glyphicon-chevron-up"></span></a>
                <a href="#!" onclick="questsShow();" class="nav-link"><span class="glyphicon
glyphicon-chevron-down"></span></a>
            </span>
        </div>
        <div id="quests-content" class="content">
        </div>
    </div>
</div>
```

So, what I did was replace the v and ^ with spans. The collapse has classes for the glyphicon and glyphicon-chevron-up to display the up chevron. The expand has classes for the glyphicon and glyphicon-chevron-down to display the down chevron.

So what we're really going to tackle in the tutorial is creating empires, joining empires and inviting players to join your empire. For that I will be adding a controller and some views. First thing we need is are some models. The first holds a subset of the information in the Players table that is displayed on the empire panel. Right click the Models folder, select Add and then Class. Name this new class EmpirePlayerViewModel. Here is the code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace DragonsClaw.Models
{
    public enum Role { Host, ViceHost, Member }

    public class EmpirePlayerViewModel
    {
        public string Ruler { get; set; }
        public string Sector { get; set; }
        public Role Role { get; set; }
        public string Race { get; set; }
        public string Class { get; set; }
        public int Planets { get; set; }
        public int Networth { get; set; }
    }
}
```

Other than an enumeration for the roles in an empire there isn't much out of the ordinary here. There are properties for the ruler of the sector, the sector name, the role in the alliance, the race, the class, the number of planets and the player's net worth.

In the alliance view we need all of the fields and a list of players to display in a grid. Right click the Models folder, select Add and then Class. Name it EmpireViewModel. Here is the code.

This is just a simple class that holds properties for the fields in the database. For the host instead of an integer I will be grabbing the name from the database.

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Web;

namespace DragonsClaw.Models
{
    public class EmpireViewModel
    {
        public int EmpireId { get; set; }
        public string EmpireName { get; set; }
        public string EmpireDescription { get; set; }
        public string Decree { get; set; }
        public string Host { get; set; }
        public int ViceHost1 { get; set; }
        public int ViceHost2 { get; set; }
        public Point Unimatrix { get; set; }

        public List<EmpirePlayerViewModel> Players { get; set; }

        public EmpireViewModel()
        {
            Players = new List<EmpirePlayerViewModel>();
        }
    }
}
```

The class has properties for the fields in the Empires table. There is a List<EmpirePlayerViewModel> that holds the players in the empire. The constructor initializes the list.

There is also a model that holds a list of all of the empires. Right click the Models folder, select Add and then Class. Name this new class EmpireListViewModel. Here is the code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace DragonsClaw.Models
{
    public class EmpireListViewModel
    {
        public List<EmpireViewModel> Empires = new List<EmpireViewModel>();
    }
}
```

There is just one field in this class that is a List<EmpireViewModel>. It initializes the List when it is created.

There is one other model that I added. That is a model for the creation of empires. Right click the Models folder, select Add and then Class. Name this new class CreateEmpireModel. Here is the code for that class.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace DragonsClaw.Models
{
    public class CreateEmpireModel
    {
        public string EmpireName { get; set; }
        public bool Create { get; set; }
        public string Message { get; set; }
    }
}
```

This model has three properties. EmpireName is the name of the empire. Create is for a checkbox that the player has to select to create an empire. Message holds error messages passed back if creating the empire fails.

Now we need a controller. Right click the Controllers folder in the Solution Explorer, select Add then Controller. From the list that comes up choose MVC 5 Controller – Empty. For the name enter EmpireController. Here is the code.

```
using DragonsClaw.Models;
using Microsoft.AspNet.Identity;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace DragonsClaw.Controllers
{
    public class EmpireController : Controller
    {
        // GET: Empire
        [Authorize]
        public ActionResult Index()
        {
            EmpireViewModel model = DataLayer.GetEmpire(User.Identity.GetUserName());

            if (model.EmpireId == -1)
            {
                return RedirectToAction("List");
            }

            return PartialView("_EmpireView", model);
        }

        [Authorize]
        public ActionResult List()
        {
            EmpireListViewModel model = DataLayer.GetEmpireList();

            return PartialView("_EmpireListView", model);
        }

        [Authorize]
        public ActionResult Create()
        {
            return PartialView("_CreateEmpireView");
        }

        [Authorize]
        [HttpPost]
        public ActionResult Create(CreateEmpireModel model)
        {
            PlayerViewModel player = DataLayer.GetPlayer(User.Identity.GetUserName());

            if (!ModelState.IsValid)
            {
                model.Message = "Please enter a name and check the confirmation check box.";
                return PartialView("_CreateEmpireView", model);
            }
            if (string.IsNullOrEmpty(model.EmpireName))
            {
                model.Message = "You must enter a name for the empire.";
                return PartialView("_CreateEmpireView", model);
            }

            if (model.Create == false)
            {
                model.Message = "You must accept the terms for creating an empire.";
                return PartialView("_CreateEmpireView", model);
            }

            if (player.Credits < 1000000)
            {
                model.Message = "You do not have the 1,000,000cr.";
                return PartialView("_CreateEmpireView", model);
            }

            bool result = DataLayer.CreateEmpire(model, player);

            if (result)
            {
                return PartialView("_CreateEmpireSuccessView");
            }

            model.Message = "There was an error creating the empire. Try again later.";

            return PartialView("_CreateEmpireView", model);
        }
    }
}
```

All of the actions have the [Authorize] attribute so a user has to be signed into the game in order to access them. If they are not signed in they will be redirected to the login page. The Index action creates a EmpireViewModel by calling the GetEmpire method of the DataLayer class passing in the user name. If the EmpireId returned is -1 I redirect to the List action which lists the empires. I then return a partial view, _EmpireView, passing it the model. The partial view will be rendered in the empire panel.

The List action calls the GetEmpireList of the DataLayer class to get an EmpireListViewModel. It then returns a partial view, _EmpireListView, passing it the model that was returned.

There are two Create actions. One for the GET verb and one for the POST verb. The GET action just returns a partial view, _CreateEmpireView. The POST action takes a CreateEmpireModel model. It calls the GetPlayer method of the DataLayer passing in the user name of the current user to get the player model. If the ModelState is not valid it sets the Message property of the model and returns a partial view _CreateEmpireView passing it the model. If the EmpireName property is null or empty I set the Message property of the model and return the partial view passing in the model. I do the same thing if the Create field is false and if the player does not have the one million credits it costs to create an empire. If the model is valid and the data is good I call the CreateEmpire method of the DataLayer passing in the model and the player. If the return result is true I return a partial view indicating success. Otherwise I set the Message property and return the _CreateEmpireView passing in the model.

Now I'm going to add the methods I added to the DataLayer. Open the DataLayer and add the following code.

```
        internal static EmpireViewModel GetEmpire(string username)
        {
            EmpireViewModel model = new EmpireViewModel
            {
                EmpireId = -1
            };

            PlayerViewModel player = DataLayer.GetPlayer(username);
            int hostId = -1;

            try
            {
                using (SqlConnection connection = new SqlConnection())
                {
                    connection.ConnectionString = 
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                    connection.Open();

                    using (SqlCommand command = new SqlCommand())
                    {
                        command.Connection = connection;
                        command.CommandText = "SELECT * FROM Empires WHERE EmpireId = 
@EmpireId";

                        int empireId = DataLayer.GetEmpireId(player.PlayerId);

                        SqlParameter p = new SqlParameter
                        {
                            ParameterName = "@EmpireId",
                            Value = empireId
                        };

                        command.Parameters.Add(p);

                        var reader = command.ExecuteReader();

                        if (reader.HasRows)
                        {
                            reader.Read();

                            hostId = reader.GetInt32(3);

                            model.EmpireId = reader.GetInt32(0);
                            model.EmpireName = reader.GetString(1);
                            model.EmpireDescription = reader.GetString(2);
                            model.Host = DataLayer.GetPlayer(reader.GetInt32(3)).PlayerName;
                            model.ViceHost1 = reader.GetInt32(4);
                            model.ViceHost2 = reader.GetInt32(5);
                            model.Decree = reader.GetString(8);
                            model.Unimatrix = new System.Drawing.Point(
                                reader.GetInt32(6),
                                reader.GetInt32(7));
                        }

                        reader.Close();
                        command.CommandText = "SELECT * FROM EmpirePlayer AS e INNER JOIN " +
                            "Players AS p ON e.PlayerId = p.PlayerId WHERE e.EmpireId = 
@EmpireId";

                        command.Parameters.Clear();

                        p = new SqlParameter()
                        {
                            Value = empireId,
                            ParameterName = "@EmpireId"
                        };
                        command.Parameters.Add(p);
                        reader = command.ExecuteReader();

                        while (reader.Read())
                        {
                            EmpirePlayerViewModel q = new EmpirePlayerViewModel
                            {
                                Ruler = reader.GetString(5),
                                Sector = reader.GetString(6),
                                Race = GetRace(reader.GetInt32(19)),
                                Class = GetClass(reader.GetInt32(18)),
                                Planets = reader.GetInt32(8),
                                Networth = reader.GetInt32(32),
                                Role = Role.Member
                            };

                            int id = reader.GetInt32(1);

                            if (hostId == id)
                            {
                                q.Role = Role.Host;
                            }
                            else if (model.ViceHost1 == id || model.ViceHost2 == id)
                            {
                                q.Role = Role.ViceHost;
                            }

                            model.Players.Add(q);
                        }
                    }

                    connection.Close();
                }
            }
            catch (Exception exc)
            {
                LogMessage(exc.Message, "GetEmpire");
            }

            return model;
        }

        internal static bool CreateEmpire(CreateEmpireModel model, PlayerViewModel player)
        {
            bool success = false;

            try
            {
                using (SqlConnection connection = new SqlConnection())
                {
                    connection.ConnectionString = 
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                    connection.Open();

                    using (SqlCommand command = new SqlCommand())
                    {
                        command.Connection = connection;
                        command.CommandText =
                            "INSERT INTO Empires (EmpireName, EmpireDescription, HostID, " +
                            "ViceHost1Id, ViceHost2Id, UnimatrixX, UnimatrixY, Decree) VALUES " 
+
                            "(@Name, '', @HostId, -1, -1, 0, 0, '')";

                        SqlParameter p = new SqlParameter
                        {
                            Value = model.EmpireName,
                            ParameterName = "@Name"
                        };

                        command.Parameters.Add(p);

                        p = new SqlParameter()
                        {
                            Value = player.PlayerId,
                            ParameterName = "@HostId"
                        };

                        command.Parameters.Add(p);
                        command.ExecuteNonQuery();

                        int empireId = GetEmpireIdByHost(player.PlayerId);

                        command.Parameters.Clear();
                        command.CommandText = "INSERT INTO EmpirePlayer (PlayerId, EmpireId) " 
+
                            "VALUES (@PlayerId, @EmpireId)";

                        p = new SqlParameter()
                        {
                            Value = player.PlayerId,
                            ParameterName = "@PlayerId"
                        };

                        command.Parameters.Add(p);

                        p = new SqlParameter()
                        {
                            Value = empireId,
                            ParameterName = "@EmpireId"
                        };

                        command.Parameters.Add(p);
                        command.ExecuteNonQuery();
                        connection.Close();
                        return true;
                    }
                }
            }
            catch (Exception exc)
            {
                LogMessage(exc.Message, "CreateEmpire");
            }
            return success;
        }

        private static int GetEmpireId(int playerId)
        {
            int empireId = -1;

            try
            {
                using (SqlConnection connection = new SqlConnection())
                {
                    connection.ConnectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                    connection.Open();

                    using (SqlCommand command = new SqlCommand())
                    {
                        command.Connection = connection;
                        command.CommandText = "SELECT * FROM EmpirePlayer AS e INNER JOIN
Players AS p ON e.PlayerId = p.PlayerId  WHERE p.PlayerId = @PlayerId";

                        SqlParameter p = new SqlParameter
                        {
                            ParameterName = "@PlayerId",
                            Value = playerId
                        };

                        command.Parameters.Add(p);

                        var reader = command.ExecuteReader();

                        if (reader.HasRows)
                        {
                            reader.Read();
                            empireId = reader.GetInt32(0);
                        }
                    }

                    connection.Close();
                }
            }
            catch (Exception exc)
            {
                LogMessage(exc.Message, "GetEmpireId");
            }

            return empireId;
        }

        private static int GetEmpireIdByHost(int playerId)
        {
            int empireId = -1;

            try
            {
                using (SqlConnection connection = new SqlConnection())
                {
                    connection.ConnectionString = 
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                    connection.Open();

                    using (SqlCommand command = new SqlCommand())
                    {
                        command.Connection = connection;
                        command.CommandText = "SELECT * FROM Empires  WHERE HostId = 
@PlayerId";

                        SqlParameter p = new SqlParameter
                        {
                            ParameterName = "@PlayerId",
                            Value = playerId
                        };

                        command.Parameters.Add(p);

                        var reader = command.ExecuteReader();

                        if (reader.HasRows)
                        {
                            reader.Read();
                            empireId = reader.GetInt32(0);
                        }
                    }

                    connection.Close();
                }
            }
            catch (Exception exc)
            {
                LogMessage(exc.Message, "GetEmpireIdByHost");
            }

            return empireId;
        }

        internal static EmpireListViewModel GetEmpireList()
        {
            EmpireListViewModel model = new EmpireListViewModel();

            try
            {
                using (SqlConnection connection = new SqlConnection())
                {
                    connection.ConnectionString = 
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
                    connection.Open();

                    using (SqlCommand command = new SqlCommand())
                    {
                        command.Connection = connection;
                        command.CommandText = "SELECT * FROM Empires";

                        var reader = command.ExecuteReader();

                        while (reader.Read())
                        {
                            EmpireViewModel m = new EmpireViewModel
                            {
                                EmpireId = reader.GetInt32(0),
                                EmpireName = reader.GetString(1),
                                EmpireDescription = reader.GetString(2),
                                Host = DataLayer.GetPlayer(reader.GetInt32(3)).PlayerName,
                                ViceHost1 = reader.GetInt32(4),
                                ViceHost2 = reader.GetInt32(5),
                                Decree = reader.GetString(8),
                                Unimatrix = new System.Drawing.Point(
                                    reader.GetInt32(6),
                                    reader.GetInt32(7))
                            };

                            model.Empires.Add(m);
                        }
                    }

                    connection.Close();
                }
            }
            catch (Exception exc)
            {
                LogMessage(exc.Message, "GetEmpire");
            }

            return model;
        }
```

The first method is GetEmpire that takes a string parameter that is the username, or email. It creates a new EmpireViewModel and sets the EmpireId to -1. I then call GetPlayer to get the player, passing in the username parameter. I set a local variable hostId to -1. The database code takes place in a try-catch block as usual. As usual I create a SqlConnection, grab the connection string using the ConfigurationManager and DefaultConnection. I then create the SqlCommand, set the connection and the CommandText property. It is a SELECT that selects rows from the Empires table where the EmpireId is a parameter. To get the EmpireId I need to query the EmpirePlayer table and check if there is a row that has the PlayerId of the player. I do that by calling a method GetEmpireId. I create an SqlParameter, set its ParameterName property to @EmpireId and its Value property to the value retrieved by calling GetEmpireId. The parameter is added to the Parameters collection. I then call the ExecuteReader method to get the rows. If there are rows I read the first row. I set the hostId variable to the 3 column which is the HostId column. I then set the properties of the model variable to the columns of the row. For Unimatrix I create a point using the two columns. I now close the reader. I set the CommandText property of the command to a SELECT query that joins the EmpirePlay and Players tables on the PlayerId columns in both tables. I clear the Parameters collection and add a new parameter with a Value of empireId and ParameterName of @EmpireId. The parameter is added to the Parameters collection and I set reader to command.ExecuteReader to get the rows. While there are rows I create an EmpirePlayerViewModel. Ruler is set to the 6th column which is the 4th column in the Players table. Sector is set to the 7th column. Race is set to the result of GetRace passing in the 20th column. Class is set to the result of GetClass passing in the 19th column. Planets is the 9th column and Networth is the 33rd. I initially set Role to Role.Member. I then get the id of the player, which is the 2nd column. If hostId and id are the same this player is the host and Role is set to Role.Host. If id is either ViceHost1Id of the model or ViceHost2Id of the model then Role is set to Role.ViceHost. The player is then added to Players collection of the model. The method flows like the other methods that I created from here. The connection is close. In the catch box I call the LogMessage method to log the message in the database. I then return the model.

The CreateEmpire method takes a CreateEmpireModel and a PlayerViewModel parameter. It sets a success variable to false. There is a try-catch block and the SqlConnection/SqlCommand block as usual. For the CommandText is an INSERT into the Empires table. The INSERT specifies all fields in the table except the EmpireId which is an autoincrement identity column that doesn't have to be specified. For the values I have a parameter for the EmpireName and a parameter for the HostId fields. The EmpireDescription and Decree are set to empty strings, UnimatrixX and UnimatrixY are set to zero for now and the vice host fields are set to -1. I then create the parameters for the query. Then I call ExecuteNonQuery to run the command. I then call GetEmpireIdByHost passing in the player's id to get the empire id so I can insert into EmpirePlayer. I create an INSERT query but into EmpirePlayer this time. It has parameters @PlayerId and @EmpireId that are the player and empire ids. I creates parameters for then and add them to the Parameters collection. I then call ExecuteNonQuery to run the command. If no exception is thrown I close the connection and return true. The catch logs any exceptions. Finally the method returns the success variable.

The next method is GetEmpireId. It returns a player's empire id. It isn't easy to get at. What you have to do is join the EmpirePlayer table on the Players table on the PlayerId columns and then extract rows where the PlayerId in the Players table matches the desired PlayerId. After setting a local variable to code flows the same as other database methods. In the CommandText I create a SELECT statement that does what I described above. Since I used a parameter in the query I create a parameter and add it to the collection. I then call ExecuteReader to retrieve the rows. If the reader has rows I read the first row then set the empireId variable to the first column, the EmpireId. I close the connection, log any errors and finally return the empire id.

The next method is GetEmpireIdByHost. It takes an integer parameter that is the id for the host. It flows the same as the GetEmpireId method. The only difference is the query. Instead of doing a join on the Players table it just does a SELECT of the EmpirePlayer table and uses the HostId column in the WHERE clause.

The last method is GetEmpireList which gets a list of empires. It creates a new EmpireListViewModel to be returned at the end of the method. The database code is as usual until the CommandText which is a SELECT that selects all of the empires in the Empires table. Next it calls ExecuteReader to get a SqlDataReader for the rows returned by the query. While there are rows it creates an EmpireViewModel getting the properties of the model, except the list of players that we don't need.

Now I will add the views. I will start with the _EmpireView partial view. Expand the Views folder. Right click the Shared folder, select Add and then View. Check the Create as partial view checkbox. Name the new view _EmpireView. Here is the code.

```
@model DragonsClaw.Models.EmpireViewModel

<div class="game-content">
    <div class="header">
        <h2>Empire</h2>
    </div>
    <div class="title"></div>
    <div class="title">
```

```
        <ul class="menu-bar">
            <li class="active-tab" id="etab1"><a href="#"
onclick="showEPanel1()">Empire</a></li>
            <li id="etab2"><a href="#" onclick="showEPanel2()">Edit Empire</a></li>
        </ul>
    </div>
    <div id="tab-container">
        <div id="epanel1">
            <h3>Empire of @Model.EmpireName</h3>
            <table id="empire-list" class="data-table">
                <thead>
                    <tr>
                        <th>Sector Name</th>
                        <th>Ruler</th>
                        <th>Role</th>
                        <th>Race</th>
                        <th>Class</th>
                        <th>Planets</th>
                        <th>Networth</th>
                        <th>NWPP</th>
                    </tr>
                </thead>
                @foreach (var p in Model.Players)
                {
                    <tr class="align-center">
                        <td>@p.Sector</td>
                        <td>@p.Ruler</td>
                        <td>
                            @switch (p.Role)
                            {
                                case DragonsClaw.Models.Role.Host:
                                    <span>Host</span>
                                    break;
                                case DragonsClaw.Models.Role.ViceHost:
                                    <span>Vice Host</span>
                                    break;
                                default:
                                    <span>Member</span>
                                    break;
                            }

                        </td>
                        <td>
                            @p.Race
                        </td>
                        <td>
                            @p.Class
                        </td>
                        <td>@p.Planets.ToString("N0")</td>
                        <td>@p.Networth.ToString("N0")</td>
                        <td>@((p.Networth / (float)p.Planets).ToString("N1"))</td>
                    </tr>
                }
            </table>
        </div>
        <div id="epanel2">
            <form id="empire-details">
                <div>Empire name: @Html.TextBoxFor(m => m.EmpireName, new { @class = "attack-
field", maxlength = "50" })</div>
                <div>Empire description: @Html.TextAreaFor(m => m.EmpireDescription)</div>
                <div>Imperial orders: @Html.TextAreaFor(m => m.Decree)</div>
                <div><input id="update-button" type="button" value="Update" 
onclick="updateEmpire()" class="button" /></div>
            </form>
        </div>
    </div>
</div>
```

The model for the view is EmpireViewModel. The view is wrapped in a div with the class game-content that all partial views have. There is a div that views have that is a header. Inside the div is an h2 with the title of the header. I have an empty div with the class title. There is another div with class title that is a menu bar. The unordered list is the actual menu bar. There are two list items. The first has the class active-tab and the id etab1. There is an anchor with an onclick that calls the Javascript function showEPanel1. This will show the first of two divs defined below. The next list item has an id of etab2 and an anchor with an onclick of showEPanel2 that will display that panel. The next div has the id tab-container. Inside that div are the two panels, epanel1 and epanel2. epanel1 displays the name of the empire and a table with the players in the empire. The columns of the table are the ones from the EmpirePlayerViewModel plus net worth per planet. In a foreach loop I write out all of the players into the table. Then there is epanel2. Inside that div is a form with a text box for the empire name, a text area for the empire description and a text area for the decree. There is also a button with a onclick that calls a function updateEmpire.

The next view I will implement is _EmpireListView. Right click the Shared folder under Views, select Add and then View. Make sure the Create as partial view checkbox is selected and name the view _EmpireListView. Here is the code.

```
@using DragonsClaw.Models
@model DragonsClaw.Models.EmpireListViewModel

<div id="empire-list" class="game-content">
    <div class="header">
        <h2>Empire List</h2>
    </div>
    <div class="title"></div>
    <div>
        <span class="empire-text">You are not part of an empire.</span>
        <button id="create-empire" class="button" onclick="createEmpire()">Create</button>
    </div>
    <table id="empire-list-table" class="data-table">
        <thead>
            <tr>
                <th>Empire</th>
                <th>Host</th>
                <th>Apply</th>
            </tr>
        </thead>
        @foreach (EmpireViewModel m in Model.Empires)
        {
            <tr>
                <td>@m.EmpireName</td>
                <td>@m.Host</td>
                <td>
                    <button class="apply-button" onclick="applyEmpire(@m.EmpireId)">
                        Apply
                    </button>
                </td>
            </tr>
        }
    </table>
</div>
```

The model is EmpireListViewModel which holds a List<EmpireViewModel>. Inside the wrapper div there is a div with the header class and an h2 inside of it with the title. There is then a title div that displays a border. There is a div that contains a span with text stating you are not part of an empire. There is a button that is used to create a new empire. There is a table that lists the empires. It displays the name of the empire, the host and button that can be used to apply to join the empire. In a foreach loop I loop over all of the empires. I then write out the row of the table. For the button to apply to join the alliance I call a method applyEmpire passing in the id for the empire.

The next view that I will be adding is the _CreateEmpireView. Right click the Shared folder under Views, select Add and then View. Make sure you create a partial view and name it _CreateEmpireView. Here is the code.

```
@using DragonsClaw.Models
@model DragonsClaw.Models.CreateEmpireModel

<div class="game-content">
    <div class="header">
        <h2>Create Empire</h2>
    </div>
    <div class="title"></div>
    <form id="create-empire-form">
        <div class="form-field">
            <div class="create-label">Empire name:</div>
            <div>
                @Html.TextBoxFor(m => m.EmpireName)
            </div>
        </div>
        <div class="form-field">
            <div>
                Creating an empire costs 1,000,000 credits. Are you sure? @Html.CheckBoxFor(m
=> m.Create)
            </div>
        </div>
        <input type="button" value="Create" id="submit-empire" class="button"
onclick="doCreateEmpire()" />
    </form>
    @if (Model != null && !string.IsNullOrEmpty(Model.Message))
    {
        <div id="error-message" class="red">
            @Model.Message
        </div>
        <div class="title"></div>
    }
</div>
```

The model for the view is of type CreateEmpireModel. There is a wrapper div with the class game-content. Inside that is the header div that displays Create Empire. Next there is a title div to display a horizontal line. There is then a form to collect the required data for creating an empire. It requires the name of the empire and acknowledgement that it costs 1,000,000cr to create an empire. The button calls doCreateEmpire. There is a check that the model is not null and that there is a Message. If there a div with the id error-message and class red is displayed with the message and a horizontal line underneath it.

The last view I will be implementing is the _CreateEmpireSuccessView. Right click the Shared folder under the Views folder, select Add and then View. For the name enter _CreateEmpireSuccessView and make sure the Create as partial view checkbox is selected. Here is the code.

```
<div class="game-content">
    <div class="header">
        <h2>Create Empire</h2>
    </div>
    <div class="title"></div>
    <div id="error-message" class="green">
        Empire created successfully.
    </div>
    <div class="title"></div>
</div>
```

There is a wrapper div that has the class game-content. Inside of that is the header div that displays the text Create Empire. There is a title div to display a horizontal line. Next is a div with the id error-message and class green to display the background in green. It displays the message that the empire was created successfully.

Now I'm going to turn my attention to the CSS for styling the empire content. Add the following CSS to the dragons-claw.less file.

```
/* Tables */
.data-table {
    width: 100%;
    border-collapse: collapse;
    border: 1px solid white;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial,
sans-serif;
    font-size: @fontSize;
}

/* Empires */

.empire-text {
    margin-top: 10px;
    margin-left: 10px;
}
#empire-list {
    position: relative;
}

#empire-list-table {
    margin: 50px 0 10px 0;
}

#create-empire-form {
    width: 800px;
    padding: 5px;
    color: white;
}

.create-button {
    font-family: 'postamtregular';
    font-size: @fontSize;
    margin: 5px;
    border: 0;
    border-radius: 5px;
}

#create-empire {
    display: block;
    margin: 0 auto;
}

#submit-empire {
    display: block;
    margin: 0 auto;
}

#epanel1 {
    display: block;
}

#epanel2 {
    display: none;
}
/* Content */

.button {
    border: 0;
    border-radius: 5px;
    padding: 3px;
    color: white;
    background: @accent2;
    font-family: 'postamtregular';
}

.title {
    border-bottom: 5px ridge @grey3;
}

.header {
    background: rgb(221,221,221);
    background: radial-gradient(circle, rgba(221,221,221,1) 0%, rgba(136,136,136,1) 50%, 
rgba(51,51,51,1) 100%);
}

h2 {
    font-family: 'postamtregular';
    font-size: 2em;
    text-align: center;
    margin: 0;
    padding: 10px;
    text-shadow: 5px 5px 10px #000000;
}

td, th {
    border: 1px solid white;
    padding: 3px;
}

#error-message {
    width: 86%;
    text-align: center;
    border: 5px ridge;
    border-radius: 10px;
    margin-left: 5%;
    margin-top: 20px;
    padding: 10px;
    margin-bottom: 20px;
    font-size: 20px;
    text-shadow: 5px 5px 10px #000000;
}

.red {
    background-color: #FF9500;
    color: white;
}

.green {
    background-color: #00FF95;
    color: white;
}

.menu-bar {
    list-style: none;
    font-size: 20px;
    background: rgb(221,221,221);
    background: linear-gradient(#A59800, #D8C700, #D8C700, #A59800);
    width: 100%;
    margin: 0;
    padding: 0;
    font-family: 'postamtregular';
}

.menu-bar li {
    display: inline-block;
    padding-right: 10px;
    padding: 10px;
    background-color: @highlight1;
    margin: 0;
}

.menu-bar a {
    color: white;
    text-decoration: none;
}

.menu-bar a:hover {
    color: @accent1;
}
```

I placed comments to try and organize the CSS. The first area is Tables that holds CSS related to displaying tables. The data-table class has a width of 100%. The border-collapse is set to collapse so there are just single borders. There is a font-family with fall backs if the specified font does not exist on the player's computer. It also has a font-size of @fontSize.

The next area is Empires that holds the CSS specific to empires. There is a class empire-text that has a top margin of 10px and a left margin of 10px. The selector for the id empire-list has a position of relative. The empire-list-table id has a top margin of 50px, right margin of 0px, bottom margin of 10px and a left margin of 0px. The #create-empire-form has a width of 800px, padding of 5px and displays in color white.

The class create-button has a font-family of postamtregular with a font-size of @fontSize. It has a margin of 5px. It has no border and border-radius of 5px to give it rounded corners.

The create-empire and submit-empire ids have a display property of block so that they display on their own line. The have margins set to 0 auto so they are centered horizontally in the container.

The epanel1 id is set to block so it is visable and the epanel2 id is set to none so it is not visible.

The next section is Content which contains general CSS that is shared by multiple panels. The first class is button that is to be applied to buttons so they all share similar styling. The border is set to 0 for no border. The border-radius is set to 5px to give it slightly rounded corners. There is a padding of 3px. The foreground color is white and the background color is @accent2. The font-family is set to postamtregular.

The title class is used to display a horizontal line to divide sections within a panel. It is a bottom-border that is 5px high with the property ridge and a color of @grey3.

The header class draws a radial gradient with a fall back to a shade of gray. For drawing the gradient it uses the parameters circle with color stops of a light shade of gray at 0%, a darker shade of gray at 50% and an even darker shade of gray at 100%

I have styles set for h2 tags. The have the font-family postamtregular and font-size of 2em. They are aligned center with 0 margin and a padding of 10px. I also draw a text-shadow underneath them in black.

td and th tags share the same style. They have a border of 1px solid in white. They also have a padding of 3px.

The error-message id has a width of 86% so it does not fill the enter panel. The text in it is aligned in the center. It has a border of 5px with a ridge property. It has a border radius of 10px for a noticeable rounded corner. It has a margin-left of 5%, margin-top of 20px and margin-bottom of 20px. Font-size is set to 20px. It also has a black text-shadow.

The red class has a background color of red and foreground color of white. The green class has a background color of green and foreground color of white.

The menu-bar class is for drawing tabbed panels. It is applied to an unordered list. It has a list-style of none. It has a font-size of 20px. The background is a linear-gradient with four color stops. It is reflective vertically so that it starts and stops the same. If the browser does not support gradients it falls back to a gray color. It has a width of 100% with no margin or padding. The font-family is set to postamtregular.

The li for the menu bar are set to display as inline-block so they appear on the same line. They have a padding of 10px. The background color is set to @highlight1 and they have no margin. The anchor tags on the menu bar are in white with no text decoration. When you hover on the anchors they display in @accent1.

The last thing to add is the Javascript to power the interaction between the browser and the controller. Add the following code to the GameScript.js file.

```
/* Empire */
function empireHide() {
    $('#empire-content').slideUp();
}

function empireShow() {
    $.ajax({
        type: 'GET',
        url: '/Empire/Index',
        data: '',
        contentType: 'application/json',
        dataType: 'html',
        complete: function (response) {
            $('#empire-content').html(response.responseText);
            $('#empire-content').slideDown();
        }
    });

    fillPlayerHeader();
}

function createEmpire() {
    $('#empire-content').slideUp();

    $.ajax({
        type: 'GET',
        url: '/Empire/Create',
        data: '',
        contentType: 'application/json',
        dataType: 'html',
        complete: function (response) {
            $('#empire-content').html(response.responseText);
            $('#empire-content').slideDown();
        }
    });
}

function doCreateEmpire() {
    var data = $('#create-empire-form').serialize();

    $('#empire-content').slideUp();

    $.ajax({
        type: 'POST',
        url: '/Empire/Create',
        data: data,
        dataType: 'json',
        complete: function (response) {
            $('#empire-content').html(response.responseText);
            $('#empire-content').slideDown();
        }
    });
}

function showEPanel1() {
    $('#epanel1').show();
    $('#epanel2').hide();
}

function showEPanel2() {
    $('#epanel1').hide();
    $('#epanel2').show();
}

/* Player */
function fillPlayerHeader() {
}
```

The fist function is empireHide that is called when the user click the up chevron in the menu bar for the empire panel. It uses jQuery to select the element with the id empire-content and calls the slideUp function to hide the element.

The empireShow function is called with the user clicks the down chevron in the menu bar for the empire panel. There is an AJAX call to the Index action of the Empire controller using the GET verb. It passes no data and has a contentType of application/json and dataType of html. The complete setting is called when the request finishes. It places the responseText property of the response into the empire-content element. It then calls slideDown to display the content with an animation. It calls a function stub fillPlayerHeader.

The createEmpire function is called to display the create empire view. It slides up the empire-content div to hide it. It does an AJAX call to the Create action of the Empire controller. When the request finishes it sets the html of the empire-content div to the responseText property of the response from the server. It then slides down the panel.

The doCreateEmpire function is called when creating an empire. It serializes the create-empire-form into a format that can be passed to the server using AJAX. It slides up the panel. There is then an AJAX call to the Create action of the Empire controller using POST. When the request completes the html of the empire-content div is set to the responseText property of the response. It then slides down the panel.

The showEPanel1 function is called when the first tab is clicked on the empire panel. It shows epanel1 and hides epanel2. Similarly, showEPanel2 hides epanel1 and shows epanel2.

The fillPlayerHeader function will be called to update the header of the page. For now it is just a stub but I will fill it out in another tutorial.

You should be able to build and run now. When you go to the game you will be presented with the game page. You should be able to expand the empire panel now, create a new empire and display the empire details.

That's a lot to digest in one sitting so I'm going to wrap the tutorial up here. I encourage you to follow my blog, follow me on Facebook @GameProgrammingAdventure or on Twitter @LadyAmethyst416 for the latest news on my tutorials.

I wish you the best in your Game Programming Adventures!