

***** Code

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

namespace KataTennis
{
    public class GameState
    {
        int[] playerScore = new[] {0,0};

        public int GetScoreForPlayer(int p)
        {
            return playerScore[p] ;
        }

        internal void ScorePointFor(int p)
        {
            int otherPlayer = (p == 0 ? 1 : 0);
            if (playerScore[p] < 30)
            {
                playerScore[p] += 15;
            }
            else if (playerScore[p] == 30)
            {
                playerScore[p] = 40;
            }
            else if (playerScore[otherPlayer] < 40)
            {
                playerScore[p] = 60;
            }
            else
            {
                playerScore[p] = 50;
            }
        }
    }
}
```

***** Test

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Xunit;
using Xunit.Extensions;
using Ploeh.AutoFixture.Xunit;

namespace KataTennis
{
    public class GameStateTest
    {
        [Theory, AutoData]
        public void Test1(GameState sut) // do this at home
        {
            for (int x =0; x<2; x++)
                Assert.Equal(0, sut.GetScoreForPlayer(x));
        }
        [Theory, AutoData]
        public void TestScoreForPlayer0(GameState sut)
        {
            sut.ScorePointFor(0);
            Assert.Equal(15, sut.GetScoreForPlayer(0));
        }
        [Theory, AutoData]
        public void TestScoreForPlayer1(GameState sut)
        {
            sut.ScorePointFor(0);
            Assert.Equal(0, sut.GetScoreForPlayer(1));
        }

        [Theory, AutoData]
        public void TestScoreTwice(GameState sut)
        {
            sut.ScorePointFor(0);
            sut.ScorePointFor(0);
            Assert.Equal(30, sut.GetScoreForPlayer(0));
        }
    }
}
```

```
[Theory, AutoData]
public void TestScoreTrice(GameState sut)
{
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    Assert.Equal(40, sut.GetScoreForPlayer(0));
}

[Theory, AutoData]
public void TestCharlieSheen(GameState sut)
{
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    Assert.Equal(60, sut.GetScoreForPlayer(0));
}

[Theory, AutoData]
public void TestDouce(GameState sut)
{
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    sut.ScorePointFor(0);
    sut.ScorePointFor(1);
    sut.ScorePointFor(1);
    sut.ScorePointFor(1);
    sut.ScorePointFor(0);
    Assert.Equal(50, sut.GetScoreForPlayer(0));
}
}
}
```