BEAT the BOTS!!

BEAT the BOTS!!

# Introduction

A simple automated game(s) of Rock, Paper, Scissors over HTTP - using your own locally hosted “Bot” Vs whoever else dares enter the arena!!

# Game Rules

## https://d14nx13ylsx7x8.cloudfront.net/lesson_image_blocks/assets/000/003/357/original/rock-paper-scissors.png The Standard Game

You've played Rock, Paper, Scissors before, you know the basics:

**Rock** beats **scissors**, **scissors** beat **paper**, **paper** beats **rock**.

Simples yes….well guess again…

## Extra Moves

We will also be using 2 extra moves:

* **Dynamite** beats all of the classic moves.
* **Waterbomb** beats **Dynamite**, BUT loses to all of the classic moves.

## Rolling Over...

* This isn't a game of one round; it's repeated, several hundred times. The player first to a (default of) 1000 points wins.
* You have a limited supply of dynamite, 100 is the default. If you run out of dynamite yet still throw a dynamite move the server will automatically play **WATERBOMB** instead.
* Points from draws are carried over to the next round. So, for example. If you and your opponent had both played **ROCK** twice in a row, the next throw would be worth 3 points.

# Setup

1. Download Example client zip file from <https://github.com/davidseanlittlewood/BeatTheBotsClient>.



1. Extract the zip file locally e.g. c:\temp, and open the solution file (BotExample.sln) in Visual Studio.
2. Feel free to amend the code to your heart’s desire (note: those of you that are new to C# - the basic Example client will work without any changes….it probably won’t win much, but it will work).
3. "Build Solution" in Visualstudio (F6) - if successful this will generate the .exe in the **Bin\Release** folder of the extracted project e.g. C:\temp\BeatTheBotsClient-master\BotExample\bin\Release\BotExample.exe.
4. You can run the "Bot" by double-clicking the .exe - this will set it up by default to run on port 5999.

OR

If you run via the command console you can specify the port e.g.

Start>run "CMD.exe"

"C:\temp\BeatTheBotsClient-master\BotExample\bin\Release\BotExample 1234"

1. The bot is now running - in order to participate in the games you then need to provide the following three things to [David Littlewood](mailto:david.littlewood@lexisnexis.co.uk) :
   1. Your PC IP address You can get this via Start>run>cmd.exe > “IPConfig”
   2. The port number you are using.
   3. And most importantly - the name of your Bot :-)

GAME ON!

# How the Game works

1. Each bot plays every other bot, one at a time.
2. Server sends a **start** POST web request to the two selected competing bots – with the request containing the four field value pairs (opponentName, pointsToWin, maxRounds, dynamite).
3. Server sends a **move** GET web request to each of the competing bots – asking for the Bot’s move for this round. The server records the result of this round and updates the relevant variables.
4. Server sends a **move** POST web request to each of the competing bots detailing the move submitted by the opponent bot (lastOpponentsMove).
5. Steps 2 and 3 are repeated until the maxRounds number is reached for the match.
6. The server then selects the next bots to compete and repeats.

# Hints

* You can test your Bot by using the free Chrome/Mac app [Postman](http://www.getpostman.com/) - use this to send the three web requests to your bot:

* http://<your\_bot\_url>/start  Used to initialise the match with the four **POST**ed values:
* string opponentName
* int pointstoWin
* int maxRounds
* int dynamite

* http://<your\_bot\_url>/move Used to **GET** your move

* http://<your\_bot\_url>/move Used to **POST** your opponents move
* string lastOpponentsMove

* The first main area of code you may want to look at is held with the **GetMove()** method - this holds the code that defines which "move" (e.g. ROCK/PAPER/SCISSOR/DYNAMITE/WATERBOMB) you play. You can find this within the **BotAIClass.cs** :

Machine generated alternative text: Solution ‘BotExample’ (1 project)
‘ IJ BotExample
A fr Properties
C” Assemblylnfo.cs
fr •-• References
yD App.config
A C” BotAJClass.cs
A  BotAiClass
‘ _opponentName: string
“ _lastOpponentsMove: string
_pointstoWin: mt
_maxRounds: mt
_dynamite: mt
‘i’ _winRatio: mt
Q _round: mt
0, SetStartValues(string, mt, mt, mt) : void
O SetLastOpponentsMove(string) : void
0, GetMoveO: string
0, GetRandomResponseO: string
fr C” HttpListener.cs
fr C” Program.cs

* Read the [game rules](#_Game_Rules) carefully - certain elements are more important than others…..