# Battle of the Bots

## Introduction

A simple automated game(s) of robot wars. You have a robot who has a number of offensive and defensive moves fighting other bots in a 2D area.

Contents

[Battle of the Bots 1](#_Toc529256345)

[Introduction 1](#_Toc529256346)

[Game Rules 2](#_Toc529256347)

[Setup 4](#_Toc529256348)

[Getting Started 5](#_Toc529256349)

[How the Game works 6](#_Toc529256350)

[Hints 7](#_Toc529256351)

[Testing your bot works 7](#_Toc529256352)

[Cheat Central – how to win this thing 7](#_Toc529256353)

## Game Rules

Movement

Forward – If the arena space in front of your bot is unoccupied and your opponent is not attempting to occupy the space then your bot will move one arena space forwards unless your opponent is using a shunt move. In this case your bot will be shunted back to their original position

Backward – Your bot will move one arena space backwards, be careful not to fall backwards out of the arena.

Offensive moves

Flip

If not flipped this move will attempts to flip your opponent. Flip Odds will decide on if your flip is a success.

If your bot is flipped then flip will self-right your bot.

IMPORTANT - If your bot is flipped then all moves will fail until you self-right..

RESOURCES - If you have exhausted all your Flips resource then flip will fail including the ability to self-right. Flip will reduce your remaining Flips resource by 1 every time it is used.

Shunt

Shunt will push your opponent back 1 square if your opponent is not flipped, on the square in front of your bot or moving forward into the square in front of your bot. If your opponent is flipped then they will be pushed back 2 squares.

Shunt will cause your bot to take 5 health damage regardless of flip status and will not damage your opponent.

Axe

Axe will perform an Axe attack on your opponent of they are 1 square in front of you.

A successful Axe attack will cause 10 health damage to your opponent unless your opponent is flipped and then the attack will cause 20 health damage.

Flame

Flame will fire a flamethrower attack at your opponent if they’re within 2 arena squares range.

A successful flamethrower attack will cause 20 health damage if your opponent is on the adjacent square and 10 health damage if they are 2 squares away.

Flamethrower attack will reduce your fuel by 1.

Resources

You have a limited number of the following resources.

Health - Amount of health your bot has before it becomes incapacitated.

Flips – Number of flips available in the game

Flamethrower Fuel – The amount of fuel available to use the flamethrower move

End game

Three circumstances can trigger the game to end and decide the victor.

* Your or your opponent’s health has dropped below 1. The victor is the Bot with health remaining.
* Your or your opponent falls off the edge of the arena. The victor is the Bot remaining in the arena, the game will be tied if both Bot’s simultaneously fall out of the arena and no win is awarded.
* The game moves expire (50 moves without either bot taking damage). The victor will be decided by a judge’s decision, the judge’s will use the following criteria applied in priority order.
  1. Bot who has taken least damage.
  2. Bot who has advanced furthest in the area

**NOTE: If both bots have the same damage and have advanced by the same number of squares then the game will be tied and no win awarded.**

# Setup

1. Download BotWarsClient zip file from <https://github.com/davidseanlittlewood/BotWarsClient> GIT repo.
2. Extract the zip file locally e.g. c:\temp, and open the solution file (BotExample.sln) in Visual Studio.
3. 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).
4. "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.
5. 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"

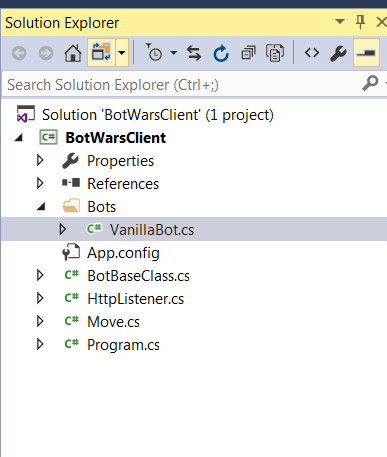
"C:\temp\BeatTheBotsClient-master\BotExample\bin\Release\BotExample 1234"

1. The bot is now running - in order to participate in the games you then need to provide the following three things to your game organiser(s).
   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 :-)

# Getting Started

Download code from the following GIT repository

<https://github.com/davidseanlittlewood/BotWarsClient>



Open the “VanillaBot.cs” class. All of the stub methods called by the games server are found in here. You can edit this class to create your own bot or create your own class inheriting from BotBaseClass.

# How the Game works

1. Each bot plays every other bot, one at a time.

Server sends a **start** POST web request to the two competing bots – with the request containing the four field value pairs (opponentBotName, health, arenaSize, flips, flipOdds, fuel, direction, startPosition

1. 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.
2. Server sends a **move** POST web request to each of the competing bots detailing the move submitted by the opponent bot (lastOpponentsMove).
3. Steps 2 and 3 are repeated until the maxRounds number is reached for the match.
4. The server then selects the next bots to compete and repeats.
5. Additionally the server sends a **flipped** POST web method when your bot has been flipped onto its back and an **opponentflipped** POST web method when your opponent’s bot has been flipped onto its back.

# Hints

## Testing your bot works

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:
  + - opponentBotName – the name of the opposing bot for this game
    - health – total health at the start of the game
    - arenaSize number of spaces in arena
    - flips number of flips available, note flips are also used to sel-right
    - flipOdds – odds percentage of flipping your opponents bot
    - fuel – how much flamethrower fuel you start with
    - direction – which direction your bot is facing
    - startPosition – which square on the arena your bot starts on
* http://<your\_bot\_url>/move Used to **GET** your move
* http://<your\_bot\_url>/move Used to **POST** your opponents move
  + - lastOpponentsMove – opponents last move
* http://<your\_bot\_url>/flipped Used to **POST** when your bot has been successfully flipped
  + http://<your\_bot\_url>/opponentflipped Used to **POST** when your opponents bot has been successfully flipped

## Cheat Central – how to win this thing

The game server provides you with essential status information.

If used properly the information from the server will help you track the exact state of each game.

Tracking state, understanding your health, opponent’s health, resource levels, bot positions and bot flip statuses will be paramount to developing a successful bot.

Understanding the patterns and strategies of your opponents and the house bot will give you the best chance of developing a winning bot.