

Online Battleground Instructions

The **online battleground** is an internet service allowing *RoPaSci 360* Player programs to easily play the game without sharing code (as would entail a breach of academic integrity). We provide a **battleground client** program which will help you connect your Player to the online battleground where you can compete against other students' programs. This document describes the usage of the battleground client and gives some notes about how this resource relates to your Project Part B assessment.

Overview

The battleground client works with the Player interface used in project part B. When you start the battleground client, it connects to a battleground server and waits for another student to do the same. When two students connect, the server sets up a game between their programs, and the battleground client displays the results.

Usage

To play a game using the battleground client, first ensure that the **battleground** package (the directory named **battleground**) is in the **same directory** as the **referee** package (the directory named **referee**), and your player package (the directory named with your team name). These three directories all need to be *next to each other* for the programs to work.

When you are ready, invoke the battleground client as follows, where **python** is the name of a Python 3.6 interpreter, **<player package>** is the name of a package containing the class **Player** to be used for your player, and **<online name>** is a string that will be used to identify your player to your opponents:

```
python -m battleground <player package> <online name>
```

For example, if your team name is **jan_kenpo** but you want to use the online name **onigiri**, call the program like this:

```
python -m battleground jan_kenpo onigiri
```

The battleground client offers many additional command-line options, including for limiting the opponents matched using a 'channel'/'passphrase' (see below); controlling output verbosity; creating an action log; and using other player classes (not named **Player**) from a package. You can find information about these additional options by running:

```
python -m battleground --help
```

Channels

To restrict your opponents such as for a private game or for any other reason, you can specify a third argument, called a 'channel', to the client. The server will only pair you up for games with other students who have included the same channel in their command.

For example, if you and a friend want to play the game together but there are many other students on the main channel at the same time, you could agree to use the channel **secret_tunnel**, as below. If nobody else uses this channel, you will find each other more easily on the battleground:

```
python -m battleground jan_kenpo onigiri secret_tunnel
```

Special channels

There are two special channels, **random** and **greedy**, which will automatically match your program for a game against two example benchmark players. These players are not the same as the ones we will use while marking, but they might still be interesting to challenge.

Rules

While you are using the battleground client and server, please keep in mind the following two important rules:

1. Please do not use offensive or malicious strings as your online name.
2. Please try not to overload the server, intentionally or accidentally, e.g. by sending thousands of connection requests within a short period.

Relation to Project Part B

The use of the battleground client and server is completely optional for Project Part B. There is no assessment connected to using or not using the battleground. We simply hope it will be a fun and possibly useful resource for students looking to test and improve their programs. Please note:

- We will try our best to keep the server online. However, it's best not to rely on the server as there may be unexpected issues or downtime, especially at popular times.
- Games played through the server do not allow the enforcement of resource limits. This means you can throw as much computation time and space at the games as you have patience and RAM. Please coordinate this with your opponent separately.
- You probably won't be able to connect to the battleground from dimefox (most internet access from dimefox is blocked for security reasons).
- The benchmark players available through the special channels are not the same as the ones we will use when testing your program after the deadline. For example, the Greedy player we use will have a different evaluation function and it won't use the throw action so much, and the Random player we use might use the throw action a lot more. You should perform your own tests if you want to be more sure your player will be able to beat our benchmarks.

Troubleshooting

Please bring issues with the battleground client or server to Matt's attention through email or the discussion board.