Arena Management

Map Creation

Overview

BattleArena comes with a handful of arenas built-in. An "Arena" in BattleArena represents an individual mode. A "Map" is used to denote an actual game map that is used. A "Competition" is what will occur in the map. Maps can exist without an active competition, but a competition cannot exist without a map.

By default, BattleArena comes with 6 different Arena types which are covered briefly on the Plugin Overview page.

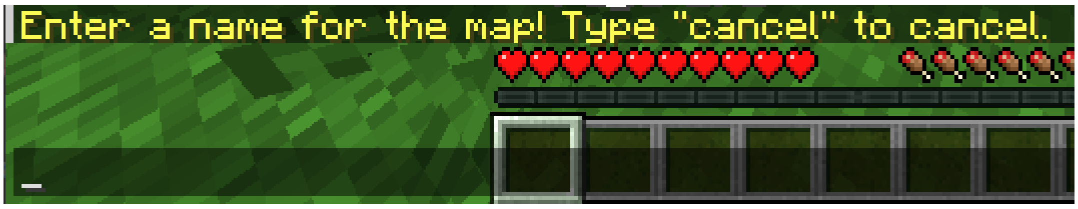
Creating a Map

A map can be created with the **/<arena> create** command. This command will open a wizard which will guide you through the map creation process. The root command determines which arena to create the map for, so **/battlegrounds create** will create a Battlegrounds map, whereas **/arena create** will create an Arena map.

The full steps are also shown below.

Map Name

The first step in creating a map is setting a map name. Type the name in chat to set the name of the map.



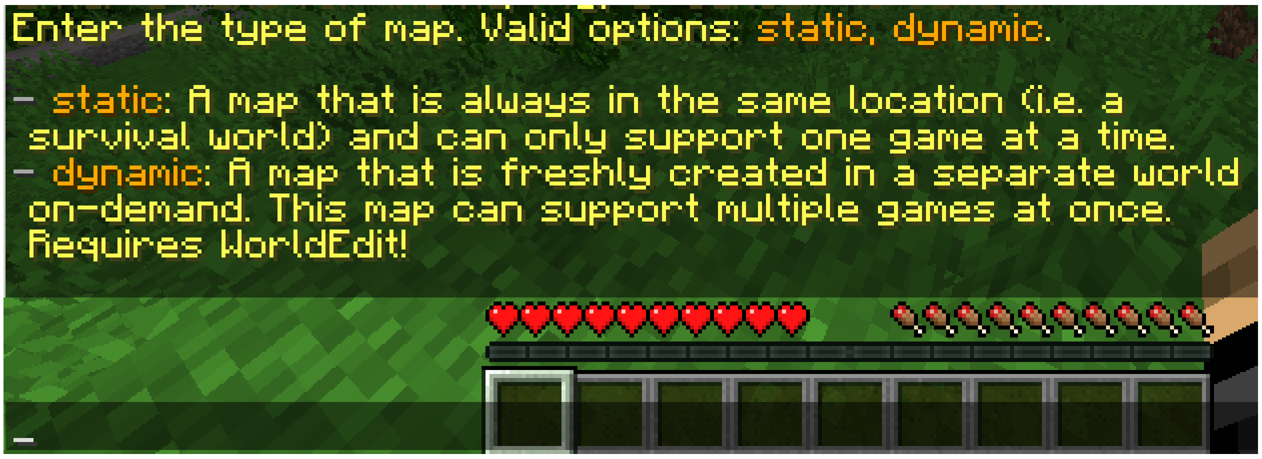
```
Enter a name for the map! Type "cancel" to cancel.
```

Map Type

You will next be prompted to choose which map type to select.

Static maps are maps that exist in the same place all the time. This means that no new world is created for your map when a competition starts. If you wish to have a gamemode in a colosseum on a survival server for instance, where players can see the active game, this is perhaps the best option. The disadvantage to this option is that there can only be one of this map available at a time.

If you have WorldEdit installed, you can select the dynamic option as well. Dynamic maps are maps created on-demand, meaning that when a player wishes to join the map, a new world is created with the map copied into it. The advantage to this option is that the map can scale up as needed, but may result in a performance hit on slower machines due to the cost of creating worlds and pasting the map contents in them. This can be mitigated slightly by using FastAsyncWorldEdit however.

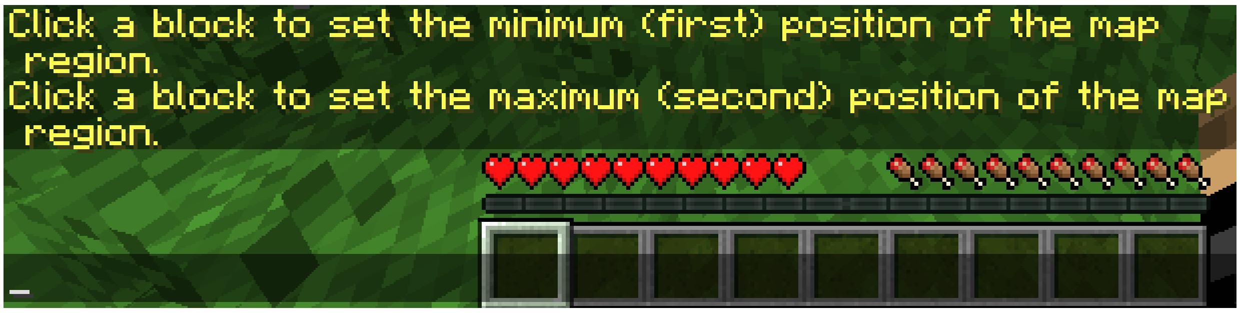


```
Enter the type of map. Valid options: static, dynamic.

- static: A map that is always in the same location (i.e. a
 survival world) and can only support one game at a time.
- dynamic: A map that is freshly created in a separate world
 on-demand. This map can support multiple games at once.
 Requires WorldEdit!
```

Map Bounds

The next step involves you selecting the region of the map. The wizard will ask you to click the blocks that should be the minimum and maximum positions. This is particularly important for dynamic maps, as it tells BattleArena what should be copied over to your dynamic world.

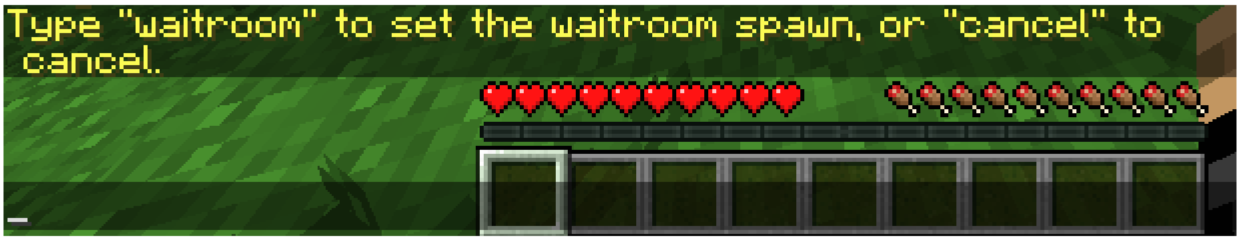


```
Click a block to set the minimum (first) position of the map
 region.
Click a block to set the maximum (second) position of the map
 region.
```

Waitroom Spawn

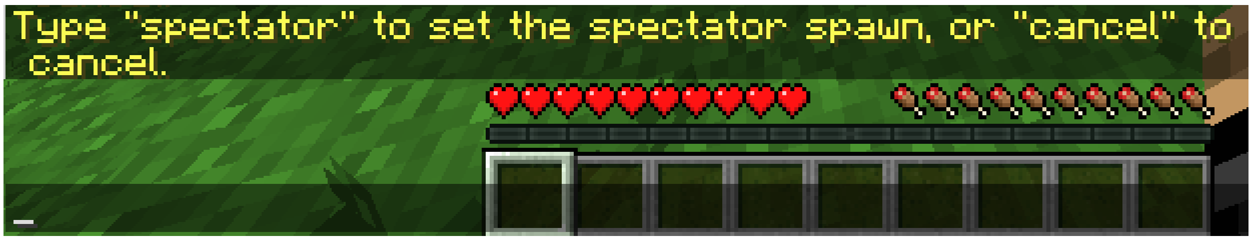
BattleArena will next ask you to select a waitroom spawn. This is where players will wait for a competition to start in arenas where a waitroom is configured. This will be set to your current position once you type "waitroom" in chat.

Even if your game mode does not have a waitroom, this option must be set to proceed with the wizard.


Type "waitroom" to set the waitroom spawn, or "cancel" to cancel.

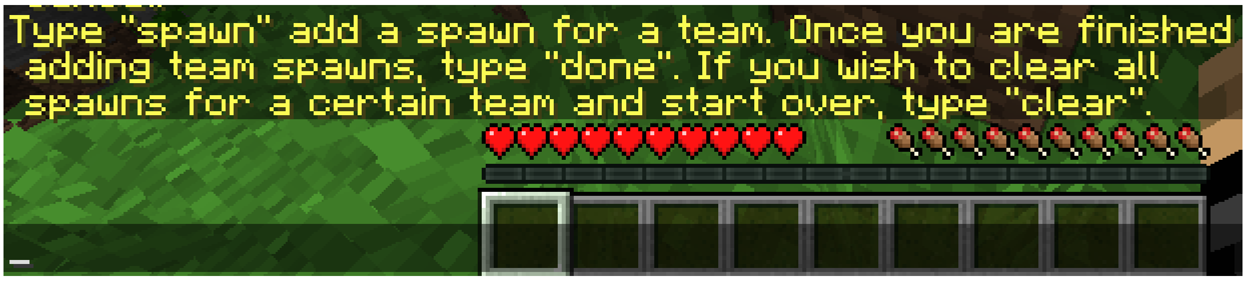
Spectator Spawn

Similar to above, BattleArena will next ask for you select a spectator spawn. This is where spectators will spawn when they spectate a competition in this map. This will be set to your current position once you type "spectator" in chat. This too is required to proceed with the wizard.

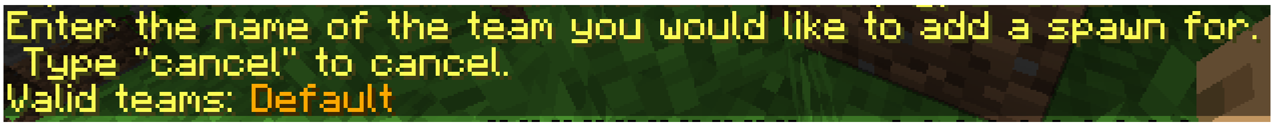

Type "spectator" to set the spectator spawn, or "cancel" to cancel.

Team Spawns

The next and final option is where you configure spawnpoints. Similar to the last two steps, when you type "spawn" in chat, a spawn point will be created at your location.


Type "spawn" add a spawn for a team. Once you are finished adding team spawns, type "done". If you wish to clear all spawns for a certain team and start over, type "clear".

Once you type "spawn", you will need to enter which team the spawn belongs to. In solo games, this will often just be "Default", however BattleArena will inform your of all the valid teams available.


Enter the name of the team you would like to add a spawn for. Type "cancel" to cancel.
Valid teams: Default

Once you have typed in the team name, you can continue adding as many spawns as you wish by continuing to type "spawn" in chat. Once you are done, type "done" in chat and the wizard will complete.

Now that the map is created, you can join it using the **/<arena> join <map>** command.

Editing a Map

Maps can be edited with the **/<arena> edit <map> <option>** command. Upon typing in the command, a list of modifiable options will show up in the tab completion. These will correspond to the documented options above, and upon executing the command, updating a specific option follows the exact same instructions.

Deleting a Map

Maps can simply be deleted using the **/<arena> delete <map>** command.

Map Storage

Maps are stored in the **plugins/BattleArena/maps** directory. Each arena will have its own directory containing the map configurations. These can also be modified by hand, and loaded in-game by running the **/ba reload** command.

Arena Configuration

Overview

BattleArena offers a significant amount of flexibility when it comes to configuring arenas.

All arenas in BattleArena are located in the **plugins/BattleArena/arenas** directory. All of these can be modified, deleted and new ones can be added. For the sake of this tutorial, we will be working with the **arena.yml** arena configuration. Each set of options is documented below.

Standard Options

- **name:** The name of the arena.
- **mode:** The mode of the arena. BattleArena only comes with the "Arena" mode built in, but extension plugins may add new modes.
- **aliases:** Command aliases to see available commands for this arena.
- **type:** The type of arena. Can either be Match or Event. See the plugin overview page for what differentiates these two.
- **modules:** A list of modules to enable in this competition. See the Modules and Other Tools page for more information on modules.

> NOTE: Changing these options for arenas with existing maps may cause the maps to no longer be linked to the arena. If this occurs, be sure to update the map arenas in **plugins/BattleArena/maps/<arena>** YAML files. You may need to rename the directory to be the same name as what is set in the **name** option above.

Team Options

Manages how to distribute players across teams along with options regarding the team size.

- **named-teams:** Whether teams should be named (i.e. Red, Blue, etc.) See the teams page for more information about teams.
- **team-size:** How many players this team fits. This can be a specific number, a range, or a minimum-inclusive range. See the following examples:
  - **team-size: 4**: Each team must have 4 players in order for a competition to start. Additionally, the team can *only* hold 4 players.

    - **team-size: 2-4:** Each team can hold between 2 and 4 players. The competition will start once each team has at least 2 players.
    - **team-size: 2+:** Each team can hold 2 or more players (minimum-inclusive). The only limit is the number of spawns set in the map.
- **team-selection:** How team selection is done in the competition. The following options are listed below:
    - **none:** No team selection. This is often used for solo competitions, where a player does not have a specific team and all other players are their enemies
    - **random:** BattleArena randomly assigns a player to a team.
    - **pick:** Players pick their team (usually during a countdown or waiting phase). This can be done using the **/<arena> team <team>** command. Server owners can also configure this to be done when the player clicks an NPC, sign or GUI button.
- **shared-spawn-points:** Whether team members are able to share spawn points. This is primarily useful for team games where all players on a team spawn at the same place.

```
team-options:
  named-teams: true
  team-size: 4
  team-amount: 4
  team-selection: pick
  shared-spawn-points: true
```

An example from **colosseum.yml** is shown above, which shows these options in use.

Lives

Controls how many lives players have in an arena. By default, this is disabled,

- **enabled:** Controls whether lives are enabled.
- **amount:** How many lives a player has by default.

Victory Conditions

Controls how a player or team may "win" a competition. Each victory condition has its own set of options which determine how a player may win. Multiple can be set per competition, and once one condition is met, the competition ends. An example is shown below for the **teams alive** and **time limit** conditions.

```
victory-conditions:
  teams-alive:
    amount: 1
  time-limit:
    time-limit: 5m
```

In this example, a team will win the competition if they are the last team standing. However, if after 5 minutes there is no winner, the competition will end and all remaining players will draw.

A full list of victory conditions can be found on the Victory Conditions Reference page.

Events & Actions

This is where the bulk of logic for BattleArena is configured. Events denote when certain "things" happen in an arena, such as when a player joins, spectates, leaves, wins, dies, etc. Upon these events, actions can be ran which control the competition behavior. These can be as simple as sending a message to the player, clearing their inventory, or kicking them from the arena.

An example from the **arena.yml** is shown below for the **on-join** and **on-leave** events.

```
events:
  on-join:
    - store{types=all}
    - change-gamemode{gamemode=adventure}
    - flight{enabled=false}
    - teleport{location=waitroom}
  on-leave:
    - clear-effects
    - restore{types=all}
```

As seen in the **on-join** event, when a player joins the arena, their state is stored. This includes their inventory, previous gamemode, health, attributes, experience, effects and last location. Their gamemode is also updated to adventure, flight is disabled for them, and they are teleported to the waitroom.

When the player leaves, as seen in the **on-leave** event, the player's previous state is restored. This will effectively restore everything that was stored when the **store** action was called.

A full list of events can be found on the Event Reference page. All actions can be found on the Action Reference page.

Options

The options section allows you to control mechanics while in the arena. The default options from the **arena.yml** is shown below.

```
options:
  - block-break{enabled=false}
  - block-place{enabled=false}
  - block-interact{enabled=false}
  - damage-entities{option=never}
  - keep-inventory{enabled=true}
  - keep-experience{enabled=true}
```

These are fairly self explanatory, and disable various mechanics such as block breaking and entity damage.

A full list of options can be found on the Option Reference page.

Phases

Perhaps one of the most powerful systems in the plugin, phases let you control exactly how a competition progresses. BattleArena comes built-in with four phases, which are all utilized in the **arena.yml**.

Each phase has a set of common options that are available for any phase. These are listed below:

- **next-phase:** The next phase to progress to.
- **allow-join:** Whether players can join during this phase. Highly recommended to set to **true** for the waiting phase.
- **allow-spectate:** Whether this map can be spectated during this phase.
- **options:** Options (see above) that are set for *this phase only*. This allows for more fine-tuning of options, such as enabling damage in an in-game state, or disabling block breaking during a waiting phase.
- **events:** Events (see above) that are set for *this phase only*. This allows for more fine-tuning of events, such as giving players items in **on-start** or teleporting them to a spawn location in **on-respawn**.

The initial phase is set in the config using the **initial-phase** option.

Waiting

This phase is simply a holding phase for players before a competition begins. Nothing really happens here, except the progression to the next phase once a player threshold is hit.

The example from **arena.yml** is broken down below:

```
phases:
  waiting:
    allow-join: true
    next-phase: countdown
    options:
      - damage-players{option=never}
```

This essentially means that during this phase, players cannot damage each other along with all the other options set in the main options section. Once the player threshold is hit, the game will move on to the countdown state.

Countdown

This phase is a generic countdown phase that runs a countdown timer, in which once concluded, will move on to the next state. This is often paired with a waiting state, before moving on to an ingame state.

Options:

- **revert-phase:** Whether the countdown should cancel and the competition should revert to the previous state if the conditions are no longer met (i.e. not enough players).
- **countdown-time**: How long the countdown should be.
- **sound:** What sound the play when a countdown number is broadcasted. Leave blank to disable.

The example from **arena.yml** is broken down below:

```
phases:
  countdown:
    allow-join: false
    allow-spectate: true
    revert-phase: true
    next-phase: ingame
    countdown-time: 5s
    options:
      - damage-players{option=never}
    events:
      on-complete:
        - teleport{location=team_spawn}
        - give-effects{effects=[speed 300 1]}
        - play-sound{sound=block.note_block.pling;pitch=2;volume=1}
```

In this example, the countdown phase will only last 5 seconds before progressing onto the ingame phase. Players can also not join during this phase, as **allow-join** is set to false. Once this phase is complete, players will be teleported to their team spawn, given speed, and a sound is played to notify them the competition is now in-game.

In-game

This phase is the phase in which victory conditions are checked upon, and when the actual game logic should run. The example from **arena.yml** is provided below:

```
phases:
  ingame:
    allow-join: false
    allow-spectate: true
    next-phase: victory
    options:
      - damage-players{option=other_team}
```

As seen here, the **damage-players** option has been updated to allow damaging players on other teams. The next phase is set to **victory**, so once the victory conditions are met, the competition will progress to the victory phase.

Victory

The final phase in the logic for a competition. This is called once victors have been determined for a competition.

Options:

- **duration:** How long the competition should remain in this phase before moving to the next phase.

The example from **arena.yml** is broken down below:

```
phases:
  victory:
    allow-join: false
    allow-spectate: false
    next-phase: waiting
    duration: 5s
    events:
      on-complete:
        - leave
      on-victory:
        - send-message{message=<green>Congrats, you won!</green>}
        - play-sound{sound=entity.player.levelup;pitch=1;volume=1}
      on-lose:
        - send-message{message=<red>Sorry, you lost!</red>}
        - play-sound{sound=block.anvil.place;pitch=0;volume=1}
      on-draw:
        - send-message{message=<yellow>It's a draw!</yellow>}
        - play-sound{sound=block.beacon.deactivate;pitch=0;volume=1}
```

During this phase, player joining is still disabled, along with spectators now. As seen in the events section though, upon the completion of the victory phase (after the 5 seconds), the player is removed from the competition with the **leave** action. All the victors are sent a message upon their victory, informing them they won the game. The losers are told they have lost the game.

In the case of a draw, the on-draw event will be ran, in which all players will be informed the competition ended in a draw.

Conclusion

These phase options are quite flexible, and not all are required. In the case of a Skirmish arena for example (provided by default in BattleArena), the game is always running and can be joined or left at any point. The game also never ends. Due to this, only the **ingame** phase is present.

Creating Custom Modes

Overview

This page covers creating a custom arena mode with some more advanced features. It largely picks up from the Arena Configuration page, so before continuing on with this section, it's recommended to familiarize yourself with the arena configurations. This is primarily a walk-through example.

In this example, we will be creating a 2v2 PvP game with up to 5 players on each team called Red vs Blue. Each player will have 3 lives, and the last remaining team alive wins. This will be a 10 minute game.

Standard Options

The first step will be configuring the standard options for the arena.

```
name: RedvsBlue
aliases: [rvb]
mode: Arena
type: Match
```

This sets the arena name to RedvsBlue, the aliases for the command to **/rvb**, the mode to Arena, and the type to Match.

Team Options

```
team-options:
  named-teams: true
  team-size: 2-5
  team-amount: 2
  team-selection: random
```

This will set the team options to use named teams, with a team size of between 2 - 5 players from earlier. This means the game will begin starting when there are at least 2 players on each team, but max out at 5 players on each.

Modules

For this mode, we will be enabling a few modules: **Classes, Team Colors**, and **Team Heads.**

In order to enable these, the following options have been added:

```
modules:
  - classes
  - team-heads
```

Lives

The lives configuration has been updated to be enabled, and give each player three lives.

```
lives:
  enabled: true
  amount: 3
```

Victory Conditions

The following victory conditions have been enabled: teams-alive and time-limit. We want the game to end when there is one team alive, but if after 10 minutes there is no victor, the game should end in a draw.

```
victory-conditions:
  teams-alive:
    amount: 1
  time-limit:
    time-limit: 10m
```

Events & Actions

For this mode, we will be using very similar events from the **arena.yml**, however some changes have been made since players will have multiple lives in this mode.

In the **on-death** event, the teleport and delay actions have been removed, leaving it as the following:

```
events:
  on-death:
    - clear-inventory
    - repsawn
```

Since the player has multiple lives, we don't want them being teleported back to the waitroom on death. However, if the player has exhausted all their lives, we also don't want them being teleported into the game. The following events below are used:

```
events:
  on-life-deplete:
    - delay{ticks=2}
    - give-effects{effects=[speed 300 1]}
    - teleport{location=team_spawn}
    - equip-class{class=warrior}
    - team-heads
  on-lives-exhaust:
    - delay{ticks=2}
    - teleport{location=waitroom}
```

As seen here, **on-life-deplete** is used when the lives are depleted. This is called upon death, but only when the player has lives remaining. We want to use this event here as it allows players to be teleported back to the game, only when they have lives to spare.

However, when the player runs out of lives, **on-lives-exhaust** is called. In this case, we want to teleport the player back to the waitroom.

More details about these events and others can be found on the Event Reference page.

> The delay option used above allows you to delay how long until the next action runs. In this case, we delay immediately since we want to ensure the player is fully respawned, and their death & lives tally have updated internally.

Options

For the options, we will be using the same options as **arena.yml**.

```
options:
  - block-break{enabled=false}
  - block-place{enabled=false}
  - block-interact{enabled=false}
  - damage-entities{option=never}
  - keep-inventory{enabled=true}
  - keep-experience{enabled=true}
  - class-equip-only-selects{enabled=true}
```

Phases

The phases for this game are the same as what is from **arena.yml,** with some minor additions.

The first option being changed is to the **countdown** phase, increasing the **countdown-time** from 5 seconds to 1 minute. Since the maximum players for this game is 10, but the minimum is 4, we want to give players not in the game a longer opportunity to join before starting the game. Additionally, the **allow-join** option has been set to **true**, since we want to allow joins during the countdown phase.

And finally, in the ingame phase, the following is added to the events section:

```
events:
  on-start:
    - team-heads
```

This will set the player's helmet to their team color, using the **team-heads** action.

Conclusion

And with this done, we now have a Red vs Blue arena! Run **/ba reload** or restart the server, and the arena will now exist on the server. A map for this arena can be created by following the Map Creation instructions, and joined with **/rvb join**.

A full YAML file for the Red vs Blue arena can be found below:

```
name: RedvsBlue
aliases: [rvb]
mode: Arena
type: Match
team-options:
  named-teams: true
  team-size: 2-5
  team-amount: 2
  team-selection: random
modules:
  - classes
  - team-heads
lives:
  enabled: true
  amount: 3
victory-conditions:
  teams-alive:
    amount: 1
  time-limit:
    time-limit: 10m
events:
  on-join:
    - store{types=all}
    - change-gamemode{gamemode=adventure}
    - flight{enabled=false}
    - teleport{location=waitroom}
  on-spectate:
    - store{types=all}
    - change-gamemode{gamemode=spectator}
    - flight{enabled=true}
    - teleport{location=waitroom}
  on-leave:
    - clear-effects
    - restore{types=all}
  on-death:
    - respawn
    - clear-inventory
  on-life-deplete:
    - delay{ticks=2}
    - give-effects{effects=[speed 300 1]}
    - teleport{location=team_spawn;random=false}
    - equip-class{class=warrior}
    - team-heads
  on-lives-exhaust:
    - delay{ticks=2}
    - teleport{location=waitroom}
options:
  - block-break{enabled=false}
  - block-place{enabled=false}
  - block-interact{enabled=false}
  - damage-entities{option=never}
  - class-equip-only-selects{enabled=true}
  - keep-inventory{enabled=true}
  - keep-experience{enabled=true}
initial-phase: waiting
phases:
  waiting:
    allow-join: true
    next-phase: countdown
    options:
      - damage-players{option=never}
      - class-equipping{enabled=true}
  countdown:
    allow-join: true
    allow-spectate: true
    revert-phase: true
    next-phase: ingame
    countdown-time: 1m
    options:
      - damage-players{option=never}
      - class-equipping{enabled=true}
    events:
      on-complete:
        - teleport{location=team_spawn}
        - give-effects{effects=[speed 300 1]}
        - play-sound{sound=block.note_block.pling;pitch=2;volume=1}
  ingame:
    allow-join: false
    allow-spectate: true
    next-phase: victory
    options:
      - damage-players{option=other_team}
    events:
      on-start:
        - equip-class{class=warrior}
        - team-heads
  victory:
    allow-join: false
```

```
allow-spectate: false
next-phase: waiting
duration: 5s
events:
  on-complete:
    - clear-effects
    - leave
  on-victory:
    - send-message{message=<green>Congrats, you won!</green>}
    - play-sound{sound=entity.player.levelup;pitch=1;volume=1}
  on-lose:
    - send-message{message=<red>Sorry, you lost!</red>}
    - play-sound{sound=block.anvil.place;pitch=0;volume=1}
  on-draw:
    - send-message{message=<yellow>It's a draw!</yellow>}
    - play-sound{sound=block.beacon.deactivate;pitch=0;volume=1}
```

Event Reference

Overview

This page contains all the events built in to BattleArena natively. Additional modules or extensions may add new events, which will be included on their documentation page.

Events List

Event	**Description**
on-start	Called when a phase starts. Best used inside of the **events** section of phases, since while this can be used globally, it'll call on *every* phase.
on-complete	Called when a phase completes. Best used inside of the **events** section of phases, since while this can be used globally, it'll call on *every* phase.
on-join	Called when a player joins a competition.
on-leave	Called when a player leaves a competition for any reason.
on-respawn	Called when a player respawns in a competition.
on-death	Called when a player dies in a competition.
on-life-deplete	Called when a player loses a life, but still has lives to spare.
on-lives-exhaust	Called when a player loses their final life and is considered "dead".
on-kill	Called when a player kills another player in a competition.
on-victory	Called when a player or team is victorious in a competition. This is only called for players who win the competition.
on-lose	Called when a player or team loses a competition. This is only called for players who lose in the competition.
on-draw	Called when a competition ends in a draw. This will be called for all players.

on-spectate	Called when a player spectates a competition.

Action Reference

Overview

This page contains all the actions built in to BattleArena natively. Additional modules or extensions may add new actions, which will be included on their documentation page.

Actions List

- **<, >** denotes a *required* option
- **[,]** denotes an optional option
- Options are separated using the semicolon (**;**)

Broadcast

- Description: Broadcasts a message to the specified audience
- Options
  - **<message>** The message to broadcast
  - **[audience]** Who the message will be broadcasted to. Can be **game** or **server** (default **game**)
  - **[type]** The type of message. Can be **chat, action_bar, title, subtitle** (default: **chat**).
- Syntax
  - broadcast{message=**<message>**;type=**[type]**}

Change Gamemode

- Description: Changes the player's gamemode
- Options
  - **<gamemode>** The gamemode to set the player to
- Syntax
  - change-gamemode{gamemode=**<gamemode>**}

Change Role

- Description: Changes the player's role
- Options
  - **<role>** The role to set the player to. Can be **playing** or **spectating**
- Syntax
  - change-role{role=**<role>**}

Clear Effects

- Description: Clears all of a player's effects
- Syntax
  - clear-effects

Clear Inventory

- Description: Clears a player's inventory
- Syntax
  - clear-inventory

Delay

- Description: Delays any subsequent actions by a specified duration
- Options
  - **<ticks>** The amount of time to delay the next action
- Syntax
  - delay{ticks=**<ticks>**}

Flight

- Description: Sets whether a play can fly
- Options
  - **<enabled>** Whether the player can fly
- Syntax
  - flight{enabled=**<enabled>**}

Give Effects

- Description: Gives potions effects to a player
- Options
  - **<effects>** The effects to give to the player
- Syntax
  - give-effects{effects=**<effects>**}
    - The effects syntax is in a list format, with root object being the effect name, and the parameters specified inside
    - Example: give-effects{effects=[speed{duration=300;amplifier=1},jump_boost{duration=300;amplifier=1}]}

Give Item

- Description: Gives an item to a player
- Options
  - **<item>** The item to give to the player

    - **[slot]** The slot to place the item in. If unspecified, it will be added to the next available slot
- Syntax:
    - give-item{item=**<item>**;slot=**[slot]**)
        - See the **Item Syntax** page for how to format items in this section

Health

- Description: Sets the player's health and hunger values
- Options
    - **<health>** The health to set for the player
    - **<hunger>** The hunger to set for the player
- Syntax
    - health{health=**<health>**;hunger=**<hunger>**}

Join Random Team

- Description: Makes a player join a random team. Only runs on players who have not selected a team already
- Syntax
    - join-random-team

Kill Entities

- Description: Kills all entities within the map bounds
- Options
    - **[excluded-groups]** The group of entities to exclude. If left empty, all entities are killed aside from players
        - The following options are allowed: **monster, animal, water_animal, water_ambient, water_underground_creature, ambient, axolotl**
- Syntax
    - kill-entities{excluded-groups=**<excluded-groups>**}
        - The kill-entities syntax is in a list format, with the separator being a comma
        - Example: kill-entities{excluded-groups=[monster,axolotl]}

Leave

- Description: Causes a player to leave the competition
- Syntax
    - leave

Play Sound

- Description: Plays a sound to the player
- Options
    - **<sound>** The sound to play

    - **[pitch]** The pitch of the sound. Values between 0 and 2 are allowed
    - **[volume]** The volume of the sound
- Syntax
    - play-sound{sound=**<sound>**;pitch=**[pitch]**;volume=**[volume]**}

Reset State

- Description: Resets the state of a player. This will clear all of their stats, and make them leave their current team (and rejoin a random one if applicable)
- Syntax
    - reset-state

Respawn

- Description: Causes a player to automatically respawn
- Syntax
    - respawn

Restore

- Description: Restores player data that was previously stored
- Options
    - **<type>** The type of data to restore.
        - The following options are available: **all, inventory, gamemode, health, attributes, experience, flight, effects, location**
- Syntax
    - restore{types=**<types>**}

Run Command

- Description: Runs a command
- Options
    - **<command>** The command to run
    - **[source]** The source of the command Can be **player** or **console** (default: **player)**
- Syntax
    - run-command{command=**<command>**;source=**[source]**}

Send Message

- Description: Sends a message to the player
- Options
    - **<message>** The message to send
    - **[type]** The type of message. Can be **chat, action_bar, title, subtitle** (default: **chat**).
- Syntax
    - send-message{message=**<message>**;type=**[type]**}

Store

- Description: Stores player data that should later be restored. Allows for instances where games may have their own inventories, but player inventories should be restored later
- Options
  - **<type>** The type of data to restore.
    - The following options are available: **all, inventory, gamemode, health, attributes, experience, flight, effects, location**
  - **[clear state]** Whether the state should be cleared. If false, this means that players will join with their previous inventory, gamemode, etc., or whatever is configured in the type option. However, the inventory that was used before the competition will be restored at the end, so if a player were to break a chestplate, the previous version of that chestplate is what would be restored at the end of the competition.
- Syntax
  - store{types=**<types>;**clear-state=[true|false]}

Teardown

- Description: Tears down a competition and removes it from starting again. Often used for event competitions, which close once the event has concluded
- Syntax
  - teardown

Teleport

- Description: Teleports a player to a location in a map
- Options
  - **<location>** The location to teleport the player. Can be **waitroom, spectator, team_spawn** or **last_location**
  - **[random]** Whether the teleport is randomized. Only used for the **team_spawn** option
- Syntax
  - teleport{location=<location>;random=**[random]**}

Option Reference

Overview

This page contains all the options built in to BattleArena natively. Additional modules or extensions may add new options, which will be included on their documentation page.

Options List

Option	**Description**	**Syntax**
block-break	Whether block breaking is enabled.	block-break={enabled=<true|false>}
block-place	Whether block placing is enabled.	block-place={enabled=<true|false>}
block-drops	Whether blocks will drop loot when broken.	block-drops={enabled=<true|false>}
block-interact	Whether players can interact with blocks (i.e. buttons).	block-interact={enabled=<true|false>}
hunger-deplete	Whether hunger should deplete.	hunger-deplete={enabled=<true|false>}
item-drops	Whether players can drop items while in a competition.	item-drops={enabled=<true|false>}
keep-inventory	Whether players keep their inventory when they die in a competition.	keep-inventory={enabled=<true|false>}
keep-experience	Whether players keep their experience when they die in a competition.	keep-experience={enabled=<true|false>}
team-selection	Whether players can switch teams.	team-selection{enabled=<true|false>}
damage-players	Whether players can damage other players.	damage-players={option=<never|other_team|always>}
damage-entities	Whether players can damage other entities.	damage-entities={option=<never|other_team|always>}

Victory Conditions Reference

Overview

This page contains all the victory conditions built in to BattleArena natively. Additional modules or extensions may add new actions, which will be included on their documentation page.

Victory Conditions List

Highest Stat

- Description: A victory condition that determines the winner based on the highest stat
- Options
  - **stat-name:** The name of the stat
  - **win-after:** The threshold of a stat value to achieve before a winner is determined (default: -1)
    - If this option is not specified, this condition can still be used, but the winner will be determined when otherwise a draw would have occurred (i.e. in **time-limit**)
  - **team-stats**: Whether to tally up all the stats of a team, rather than an individual player. (default: false)
- Example:

```
victory-conditions:
  highest-stat:
    stat-name: kills
    win-after: 20
    team-stats: false
```

Teams Alive

- Description: A victory condition that determines the winner when a certain number of alive teams are remaining
- Options
  - **amount:** The amount of teams alive
- Example:

```
victory-conditions:
  teams-alive:
    amount: 1
```

Time Limit

- Description: A victory condition that ends the competition after a specified amount of time. This will always result in a draw if a victor cannot be determined. When combined with **highest-stat**, at the end of this time, the victor will be the player with the highest number of the specified statistic
- Options
  - **time-limit:** The amount of time before the competition is ended
- Example:

```
victory-conditions:
  time-limit:
    time-limit: 5m
```