InfiniteObjects Reference

Version 1 WIP

For InfiniteObjects v0.0.1-SNAPSHOT

2013

Table of contents

Overview 3

iWGO configuration format 4

Expressions 5

Mathematical expression 5

Annex 6

iWGO configuration character and formatting key 6

Notes 6

# Overview

InfiniteObjects uses a number of components to represent the various steps and concepts necessary to generate complex world generator objects. They are three main components, each having any number of types, which provide specific functionality. InfiniteObjects includes some basic types for each component. The list of components and provided types may be found bellow. Types can be added by external plugins, and may be used exactly like the provided ones.

1. Material setter
   1. Simple
   2. Random simple
   3. Inner-outer
   4. Random inner-outer
2. Condition
   1. Cuboid
   2. Sphere
3. Instruction
   1. Shape
   2. Block
   3. Repeat

Generation of a world generator object (which in this document will be referred to as a WGO) is done in three steps.

1. Evaluate the main variables
2. Evaluate each condition, terminating if any return false
3. Execute each instruction

Defining an InfiniteObjects WGO (which in this document will be referred to as an iWGO) is done using a YAML configuration. YAML is used as many Minecraft sever owners maybe familiar with it. It is also easy to read, and the Bukkit and SpoutAPI APIs provide an API for it.

# iWGO configuration format

The following is the basic format for the iWGO YAML configurations. The key for the meaning of the characters and formatting used can be found at the end of the document. This format doesn’t include any information specific to component types.

name: [name]

*variables:*

*[name]: [****value****]*

*{…}*

setters:

[name]:

type:

properties:

(properties)

*{…}*

*conditions:*

*[name]:*

*shape: [shape]*

*mode: [“include” | “exclude"]*

*size:*

*(size)*

*position:*

*x: [****x****]*

*y: [****y****]*

*z: [****z****]*

*check:*

*- [material]*

*{…}*

*{…}*

instructions:

[name]:

type: [type]

*variables:*

*[name]: [****value****]*

*{…}*

(properties)

*{…}*

# Expressions

Expressions are used to provide dynamic numeric quantities to the various properties. They are two types of expressions: mathematical expressions, a generic way to define values through scalars, variables, operators and functions; and random number functions, used to provided faster calculation of random numbers and can only be used for generating a random integer or floating point number between a maximum and a minimum.

## Mathematical expressions

Mathematical expressions can be as simple as a one scalar, or very complex, with various functions and variables. Evaluation of these expressions is done using the exp4j library, modified for InfiniteObjects. As part of these modifications, constants and better support for custom functions and operators have been added. A number constant, functions and operators are provided by the library and InfObjects. Please refer to the tables bellow for the complete lists.

|  |  |
| --- | --- |
| Operators | |
| Operator | Symbol |
| Addition | + |
| Subtraction | - |
| Multiplication | \* |
| Division | / |
| Negation | ‘ or - |
| Remainder | % |
| Power | ^ |
| Operation priority | () or [] or {} |

|  |  |
| --- | --- |
| Constants | |
| Operator | Expression |
| *π* (Pi) | PI |
| *e* (Euler’s number) | E |

|  |  |
| --- | --- |
| Functions | |
| Name | Expression |
| Floor | floor(x) |
| Ceiling | ceil(x) |
| Absolute | abs(x) |
| Square root | sqrt(x) |
| Fast square root | fsqrt(x) |
| Logarithm (base *e*) | log(x) |
| Sine | sin(x) |
| Cosine | cos(x) |
| Tangent | tan(x) |
| Arcsine | asin(x) |
| Arccosine | acos(x) |
| Arctangent | atan(x) |
| Random integer within [min, max] | ranI(min, max) |
| Random floating point within [min, max[ | ranF(min max) |

Mathematical expressions are written the standard linear way.

'ranI(5,10) + floor( 5 \* sin(PI / 2)) % 3

Operator precedency is respected. Scientific notation is also supported.

(14 \* 10 ^5) \* (3 \* 10 ^ -3)

can be noted as

14e5 \* 3e-3

A single scalar is a valid mathematical expression.

42

## Random number functions

When only random numbers are required, a special notation can be used. This will lead the iWGO loader to use a different higher performance way of handling the random number generation. The notation is similar to the mathematical expression random number functions and the resulting number is identical for the same “min” and “max” arguments.

|  |  |
| --- | --- |
| Random number functions | |
| Function | Expression |
| Random integer within [min, max] | ranI=min-max |
| Random floating point within [min, max[ | ranF=min-max |

ranI=5-10

ranF=0-1

These cannot be mixed with mathematical expressions.

# Annex

## iWGO configuration character and formatting key

* Elements in italics are optional
* Bold elements must be mathematical expressions
* Elements between brackets are to be defined by the user, unless between quotes. The “|” character means that one, but not both, of the quoted terms must be used.
* Elements between parentheses represent a section of the configuration, which cannot be defined as it depends on the type of component.
* An ellipsis between braces mean that the key above at the same rank and its children are to be repeated (unless in italics). For lists, it means that more items can be added to the list

## Notes

* “InfObjects” and “InfiniteObjects” can be used interchangeably, and refer to the same software and project, the official name being the second.