

# **JZoneGamer**

## **Introduction**

By Mike Hahn

Copyright © 2009 JZoneGamer.com

Date Last Edited: 23-Feb-2009

# Table of Contents

<b>Overview .....</b>	<b>1</b>
<u>Introduction to JZoneGamer .....</u>	<u>1</u>
<u>Goals.....</u>	<u>1</u>
<b>Board Games .....</b>	<b>3</b>
<u>Board Game Codes .....</u>	<u>3</u>
<b>Animation.....</b>	<b>4</b>
<u>Low-Level Components .....</u>	<u>4</u>
<u>High-Level Components .....</u>	<u>4</u>
<u>Event Handling .....</u>	<u>4</u>
<b>Treescript .....</b>	<b>5</b>
<u>Treescript Overview.....</u>	<u>5</u>
<u>Language Features .....</u>	<u>5</u>

# Overview

## Introduction to JZoneGamer

[ [Home](#) ]

JZoneGamer is a software tool that lets you create multiplayer board games, as well as games with 2D, and ultimately 3D animation. You can log on to [jzonegamer.com](http://jzonegamer.com) and play these games with other JZoneGamer users.

Using a built-in scripting language called Treescrypt, developers can create almost any game imaginable. These games can then be uploaded to the JZoneGamer web site. Non-programmers can create drag-and-drop board games, and programmers can add functionality to these games.

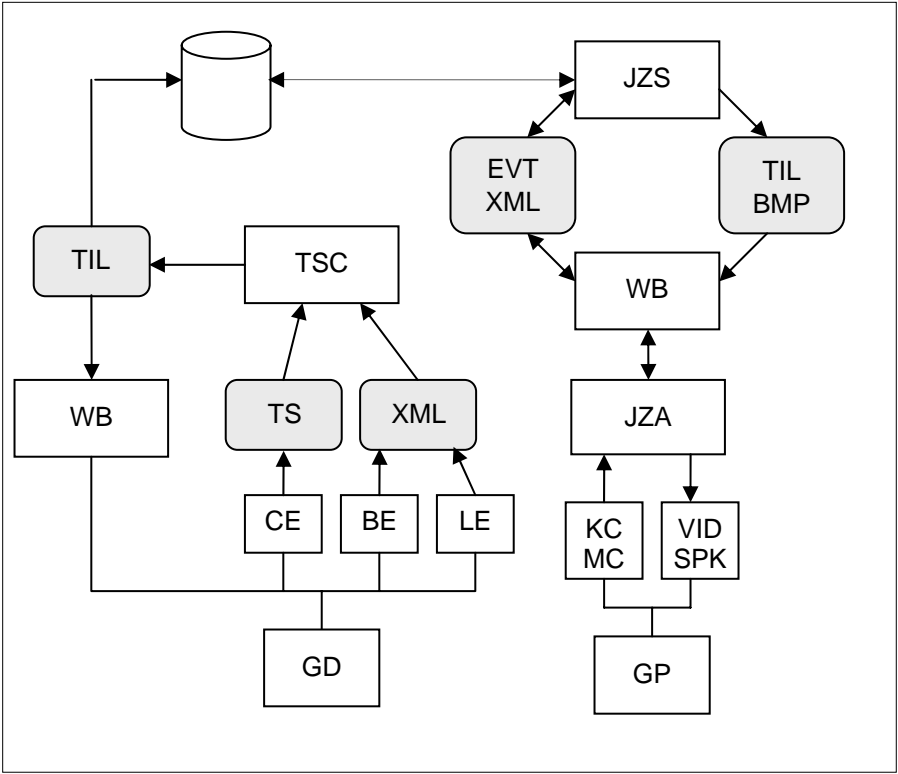
The primary goal of JZoneGamer is to enable both programmers and non-programmers to create games more easily than with a conventional RAD tool such as Visual Basic or Delphi. A secondary goal is to create in Treescrypt a simple yet powerful scripting language that is easier for a beginner to master than C#, Java or Object Pascal.

## Goals

The all-encompassing goal of JZoneGamer is ease of use. Here are the other goals, in descending order of importance:

1. Non-programming adults and teens can easily use JZoneGamer to create drag-and-drop multiplayer games, without being coached by JZoneGamer gurus and/or tech support.
2. Non-programming high school students and undergraduates can learn Treescrypt as a first programming language more easily than any other Graphical User Interface (GUI) programming tool.
3. Programming novices (who are already familiar with at least one other GUI programming tool) can master Treescrypt in one month or less, on their own, with little or no coaching from JZoneGamer gurus and/or tech support.
4. Advanced programmers can master Treescrypt as easily as falling off the proverbial log, unlike most existing GUI programming tools.
5. A vibrant JZoneGamer community blossoms, consisting of game designers and game players from all around the world.

# JZoneGamer Block Diagram



BE	Board Editor
BMP	Bitmaps
CE	Code Editor
EVT	Events
GD	Game Designer
GP	Game Player
JZA	JZone Applet
JZS	JZone Server
KC	Key Clicks
LE	Level Editor
MC	Mouse Clicks
SPK	Speaker
TIL	Treescript Intermediate Language
TS	Treescript
TSC	Treescript Compiler
VID	Video
WB	Web Browser
XML	Extensible Markup Lang.

# Board Games

## Board Game Codes

This web site describes using JZoneGamer to make multiplayer Scramble, backgammon and bridge. Non-programmers are restricted to No-Code (codeless programming) drag-and-drop type games. Code 1 (Automation) is the easiest type of game programming, which involves automating various aspects of the game-player environment and game-playing process. Code 2 (Move Constraints), or checking for illegal moves, is a little harder. Code 3 (Auto Move), playing against the computer, is the most challenging type of game programming.

**No-Code:** Codeless Programming (drag and drop)

**Code 1:** Automation of simple tasks

**Code 2:** Move Constraints (checking for illegal moves)

**Code 3:** Auto Move (playing against the computer)

# Animation

## Low-Level Components

JZoneGamer can be used to make web-based, multiplayer animated games, and these games are coded in a built-in scripting language called Treescrypt. Non-programmers can use the Level Editor and the Vector Editor to create simple animated games, and programmers can add functionality to these games.

These low-level components are the basic building blocks of animated games, otherwise known as "atoms."

**Coord:** Point on screen

**Rect:** Rectangle

**Quad:** Rectangle with fill color

**Ellipse:** Circle or oval

**Disc:** Ellipse with fill color

**Polyline:** Set of connected line segments

**Shape:** Set of connected line segments (polygon with fill)

**Bitmap:** Rectangular array of pixels (colored dots)

**Text:** Label containing text

## High-Level Components

**Vector:** collection of atoms and/or other vectors

**Window:** Rectangular area on screen, static or animated.

**User:** Game player

**Level:** Arena containing game action

**Game:** Collection of levels making up a complete game

**In-Progress:** Game in progress – web site includes list of these

## Event Handling

**Keyboard:** These events are handled at the user level.

**Mouse:** These events are handled at the vector level.

**Collision Detection:** These events are triggered when 2 atoms collide.

**Timer:** These events are triggered after every N frame changes.

# Treescript

## Treescript Overview

All JZoneGamer games are written in a built-in scripting language called Treescript. Treescript is a powerful object-oriented language which is designed with beginner programmers in mind. Pressing the question mark key (?) when in the code editor brings up a popup menu of choices valid in the context of the text cursor position. Optional Structure-Editor mode eases code entry for naïve users. The default operator/operand mode is prefix (operators come before their operands) as opposed to optional infix (binary operators come in between their operands). Treescript is based on Java, and when infix mode is selected, Treescript code strongly resembles Java, with a touch of Object Pascal thrown in for good measure.

## Language Features

- **Syntax:** Treescript is a subset of Java, although the syntax differs radically from Java, in that all operators are, by default, prefix (like Lisp) rather than infix, and parentheses are used for grouping.
- **Semicolons:** Statements and declarations are semicolon-delimited.
- **Case:** Treescript is case-sensitive: the convention for identifiers is all lower case, hyphens being used to separate parts of multi-word identifiers. Class names are an exception: the initial letter is upper case.
- **Keyboard Aid:** When enabled, this feature allows the user to enter a hyphen followed by a lower case letter by typing an upper case letter (but only when the text cursor is in the middle or at the end of an identifier). Alternatively, the user may enter a hyphen by typing a quote ('). Also, commas and periods are immediately converted into parentheses if desired (if the period key is used to enter ')', then the ')' key may be used to enter a decimal point). The user may toggle Code Menu mode by typing slash (/) instead of question mark (?). Keyboard Aid is always disabled inside comments and string literals.
- **Meta-Programming:** Treescript programs are lists, which can act as data for other Treescript programs. Structure editor mode lets newbies create simple event handlers without having to know the syntax of the Treescript programming language.