

# Why Use Treenimation?

By Mike Hahn

Copyright © 2008 Treenimation.net

Date Last Edited: 31-Jul-2008

# Table of Contents

<b>Why Use Treenimation?.....</b>	<b>1</b>
<u>Introduction .....</u>	<u>1</u>
<u>Easy to Learn .....</u>	<u>1</u>
<u>Powerful .....</u>	<u>3</u>
<u>Treenimation Business Model .....</u>	<u>3</u>

# Why Use Treenimation?

## Introduction

[ [Home](#) ]

Treenimation is a software tool which enables you to create multiplayer board games, as well as games with 2D animation (and ultimately 3D animation). You can log on to Treenimation.net and play these games with other Treenimation users. Treenimation is based on a powerful yet easy to learn, object-oriented scripting language called Treescript. Non-programmers can create drag-and-drop games, and both novice and advanced programmers can use Treescript to add functionality to these games.

What sets Treescript apart from other languages is its dual-syntax capability, which defaults to prefix mode (all operators come before their operands), and also gives you the option of infix mode (binary operators come in-between their operands). When prefix mode is enabled, Treescript resembles Lisp, but when infix mode is enabled, it looks a lot like Java (Treescript is based on Java).

## Easy to Learn

The Treescript language is based on Java, but unlike Java, only parentheses are used for grouping (no square brackets or curly brackets), and statements/declarations are separated with semicolons. When in the code editor, the user is always in one of 3 code entry modes: free form, structure editor, and code menu. The question mark (?) is used to enter code menu mode. The Escape key toggles between free form and structure editor modes. Structure Editor mode is (possibly) unavailable when the syntax mode is infix (binary operators come in-between their operands). Syntax mode defaults to prefix (all operators come before their operands), and code entry mode defaults to structure editor.

## Structure Mode Commands

A bottom-level token (e.g. a keyword, identifier, operator, or constant) or an entire list is often highlighted. Using the Shift key in conjunction with the Up/Down Arrow keys, it is possible to select more than one token/list at a time.

- **Esc** – toggle between Free Form and Structure Editor modes
- **Up Arrow** - go to previous list element
- **Down Arrow** - go to next list element
- **Left Arrow** - go to parent list
- **Right Arrow** - go to first child element (if none, display text cursor following current bottom-level token)
- **Shift+Up/Down Arrow** - select a range of tokens/lists
- **Printable Char.** - incrementally select valid matching menu item (if any)
- **Backspace** - undo insertion of previous printable char.
- **Delete** - delete current token/list
- **Enter** - display text cursor, insert space after cursor (or insert result of incremental menu selection)
- **Space** - display text cursor, insert space before cursor (or insert result of incremental menu selection)
- **Ctrl+Enter** – if at end of line, append blank line (otherwise break line into 2 lines)

## Code Menu Commands

A popup menu above or below text cursor (and including text cursor) is displayed. The contents of this menu include all valid code elements in the context of the text cursor (ignoring anything after the text cursor). If the current menu item refers to a list, the entire list is highlighted (defaults to light gray if background color of whitespace is white).

- **Question Mark** - toggle between Code Menu and Free Form/Structure Editor modes
- **Esc** - show/hide code menu
- **Up Arrow** - move selection up (scroll up after pressing Esc)
- **Down Arrow** - move selection down (scroll down after pressing Esc)
- **Left Arrow** - go to parent code menu
- **Right Arrow** - go to lower-level code menu, if any
- **Enter** - go to lower-level code menu (if none, insert current menu item, go to next menu item, or if none, go to parent code menu)
- **Space** - go to lower-level code menu (if none, insert current menu item, go to next menu item, or if none, go to parent code menu; exit Code Menu mode)
- **Printable Char.** - incrementally select matching menu item
- **Backspace** - undo operation of previous printable char.
- **Page Up** - page up after pressing Esc
- **Page Down** - page down after pressing Esc
- **Shift Arrow** – only used if current menu item is repeated, such as a statement in a block, a declaration, or a parameter in a parameter list
  - **Shift Up Arrow** – select previous instance of current menu item
  - **Shift Down Arrow** - select next instance of current menu item
  - **Shift Left Arrow** – insert above current menu item
  - **Shift Right Arrow** - insert below current menu item
  - **Semicolon** – toggle parent list: multi-line/single-line

## Keyboard Aid

This feature eases code entry by enabling the user to enter commonly used characters which are relatively hard to type with more easily-typed characters.

- **Hyphen:** press apostrophe ('). Use the double-quote (") for string literals.
- **Code Menu Mode:** press slash (/). Use the question mark (?) to enter divide-by (/).
- **Parentheses:** press comma (,) for the open parenthesis, and period (.) for the close parenthesis. Use the close parenthesis to enter a decimal point.
- **Hyphen (alternate):** when entering an identifier, hold down the Shift key and while doing that, press a letter key. This will enter a hyphen (-) followed by a lowercase letter.

Keyboard Aid is always disabled inside comments and string literals.

## RAD-style program development

The Treenimation Integrated Development Environment (TIDE) is similar to Visual Basic or Delphi, in which the user selects components from the component palette, drops them on the game window, and uses the Object Inspector to modify their design-time properties. The code editor is used to enter all program code, including event handlers.

## Powerful

Treescript is simple enough for beginner programmers to learn, yet powerful enough for professional game programmers to use as a prototyping tool. When the syntax mode is set to infix, Treescript code strongly resembles Java (with a touch of Object Pascal added for good measure).

Using multiplayer-enabled game components such as Card (playing card, chess piece, letter tile, etc.), Card-stack, Board-grid, and Table-grid, development of multiplayer board games is well within the grasp of programmers and non-programmers alike.

Treenimation includes a Level Editor and a Vector Editor (a "vector" is another name for a static/animated object, which may contain other vectors). The Level Editor lets you create worlds, which are inhabited by both static and animated vectors. Some game genres made possible using Treenimation include role-playing games and arcade-style (action games).

## Treenimation Business Model

Treenimation is free for all XO Laptop users. Windows users can play/create Treenimation games without charge for a 30-day trial period. After the trial period has expired, Treenimation game players must pay an annual membership fee in order to play games in multiplayer mode, and Treenimation game designers must pay the same membership fee in order to play games created by other game designers in multiplayer mode. Anyone can play games in single-computer mode or email-based mode for free, and anyone can create and play their own games in multiplayer mode for free. The annual membership fee is \$15 US.

The fact that the membership fees paid by Windows users subsidizes the operation of the Treenimation game server, allowing XO Laptop users to play for free, encourages Windows users to become members. In addition to selling subscriptions, Treenimation.net will make money from advertising. In the future, Treenimation.net will become a registered charity, helping XO Laptop users in developing countries acquire literacy and numeracy skills through playing board games. The more advanced XO users will learn programming skills as well.