**BassWhacker**

Basswhacker is a music playing / training application written in AppGameKit.

Screen Layout

|  |  |
| --- | --- |
| Bass Stave, simplified. Only supports single notes (e.g. no joins) and rests.  Scrolls Horizontally. Just notes and lines.  Each note is one sprite. | |
| Tab display similar to Yousician. Does not support multiple notes.  Scrolls Horizontally.  Each note is one sprite and one text item. | |
| Bar which shows position, sets range, and can set position. | Control Area |

The control area has icons for faster, normal, slower playback speed, mute metronome, mute playback.

Input music is in raw text form, in the following formats. Anything after // on an input line is stripped. Case and spacing is ignored except commands must all be on the same line *and* a space is required between consecutive notes (e.g. not confusing 1 1 and 11). The file format is .bass . There is no interim conversion (as in the music trainer). The identifier is the file name base (without type and directory) , this should only contain alphanumerics and spaces.

|  |  |  |
| --- | --- | --- |
| **Syntax** | **Type** | **Description** |
| EADG | Command | Sets the current string (defaults to G) |
| /1 /2 /3 /4 | Same as EADG | Sets the current string. Number is the same as in the *Note* internal structure. |
| nn | Command | Plays note at fret nn, current string, one beat |
| $ | Rest | Single beat rest. |
| - | Modifier | Modifies note to ½ beat |
| = | Modifier | Modifies note to ¼ beat |
| . | Modifier | Modifies last note to 1.5 times its length. |
| o | Modifier | Modifies note to double its length. |
| | | Command | Mandatory, you cannot rely on automatic overflow. If the bar boundary is crossed without such, causes an error. |
| {n} | Command | Sets tempo or beats in a bar (if <= 12). Both are single values. |
| :[letter] | Command | Starts a definition called [letter]. The input stream is copied to the current definition verbatim (e.g. in the above format). There are a possible 26 definitions A-Z. |
| ; | Command | End definition, goes back to compiling music. |
| @[letter] | Command | Generate for definition [letter] |

**Music Data Structures**

**Song**

|  |  |  |
| --- | --- | --- |
| name$ | string | Song name (stripped part of file name) |
| barCount | integer | number of bars |
| beats | integer | beats per bar. |
| tempo | integer | tempo in beats / minute |
| bars[] | *Bar* | bars (chunk extended, do not use size) |

**Bar**

|  |  |  |
| --- | --- | --- |
| noteCount | integer | number of notes in bar |
| notes[] | *Note* | notes in bar (chunk extended, do not use size) |
| \_\_baseID | integer | base Sprite ID for this bar (owns n to n+999 sprites) |
| \_\_loaded | integer | True if sprite/text loaded for this bar. |

**Note**

|  |  |  |
| --- | --- | --- |
| stringID | integer | String number 1-4 (EADG) |
| fret | integer | Fret position, 0 = open, 1 = first fret etc. |
| mbLength | integer | note length in millibars (1 bar = 1000 units) |
| \_\_mbPosition | integer | note position in milliBars (1 bar = 1000 units) |