# JWagner language specification

### v0.1 alpha

Table of content:

1. Language overview
2. Language basics;
   1. Comments;
   2. Basic syntax;
   3. Melody element;
   4. Playback control;
3. Commands
   1. play
   2. go
4. Future tasks

## 1. Language overview

Wagner is music playing language. Currently it developed for Java VM only, so it now calling JWagner.

## 2. Language basics

### 2.a. Comments

Comments format is “#...” for one row comment and /\* … \*/ – block comments.

Examle:

**# this is comment**

**play 1; #this is another**

### 2.b. Basic syntax

In Wagner one command separated to another throw ‘;’ symbol. Command blocks has java (or C) style – {…}.

Example:

**{**

**#this is a code block**

**command;**

**anotherCommand;**

**}**

Spaces, tabs, carret returns be ignored.

### 2.c. Melody element

Any Wagner program must have a Melody block to start:

**melody NameOfMelody {**

**…**

**}**

In command block one or more commands can be called (see p. 3).

### 2.d. Playback control

To write Jwagner program it is important to understand, how melody be played.

Wagner melody playing by **tacts**, where one **tact** – atomic amount of time, smallest valid note or pause length. Tact size also is music tempo. And more complex music must consists more smallest tacts to provide complex melody.

When melody start playing, current tact is #0 tact. All commands (exclude commands changing current tact) calling in this place executed in current tact – tact #0. When program need to go forward, **go** command must be called. This command move current tact forward for 1 or more tacts.

Melody stops playing when all notes or other commands was executed on tact, where they stored. No need to call **go** command at the end of melody – this command designed for set position of other commands on melody.

For more info about language commands read p. 3.

## 3. Commands

### 3.a play

The **play** command playing a single note **in current tact**. Note will be turned off in tact {current tact + NoteLength}.

Format of the command:

**play {NoteNumber} {NoteLengthInTacts} {MIDIChannelN}**

Examples:

**play 70 1 1; # playing note with number = 70 (MIDI encoding) on 1 MIDI channel for 1 tact**

### 3.b go

The **go** command move current tact forward for specified number of tacts (if nothing specified, by default 1 tact).

Format of the command:

**go [numberOfTacts**];

Examples:

**go; # go forward for 1 tact**

**go 3; # go forward for 3 tacts**

## 4. Future tasks

In next version language be expanded and improved with this things:

1. Add **pause** command to do music program more good-readable – command emulates pauses in melody and in fact equals to go – to be discussed;
2. Procedures support;
3. Modules (in separate files) support;
4. Note numbers in play command changed to note symbol such as A#, B, C etc.;
5. All **playNote** command params will be set nonrequired but note number;
6. Add velocity support for playing note (to be discussed);
7. Add **config** command to set tact length;
8. Add support for standart note tact tables (3/4, 4/4, 2/2 etc.) – to be discussed.