ZUMA

Language Specification

Contents

1 Datatypes

ZUMA is strongly typed.
Following datatypes can be created using literals:

1.1 Boolean

Boolean has one of values `true` or `false`.

1.2 Number

Number is a single precision floating point, i.e. f32: `1.5464`.

1.3 Point

Point is declared using two numbers inside square brackets like `[4.45,6.06]`.

1.4 Color

Color can be declared using sharp followed by hexadecimal value: `#ff00a1`. Additionally few basic colors can be declared by their name: `black`, `white`, `red`, `green`, `blue` or `yellow`.

1.5 Text

2 Language constructs

2.1 Expressions

Expressions are delimited using semicolon.

```
line start = [0,10] end = [25,50] color = #ff00a1;
```

Expressions are following constructs:
- constant declaration - function call - scope

2.2 Comments

Single line:
// this is comment
Part-line / multiline:
/* multiline comment */
let x = /* can be also in the middle of any expression */ 5;
Comments can be nested:
/* /* */ */ /* */ */ /* /* */
Anything inside comments shouldn't break compilation.

2.3 Scopes

Scope is delimited by `{` and `}`. There is list of expressions between braces. Scope is an expression.

3 Architecture

3.1 Parser

3.2 Abstract Syntax Tree

3.3 Evaluation

remove comments, eval variables, ifs and for loops

3.4 ZUMA IR

3.5 Translation

ZUMA IR to SVG model

3.6 Generate SVG