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 Coordinate system

Origin point is left upper corner. `x` is vertical axis, `y` is horizontal axis.

Therefore `[0,500]` describes upper right corner, while `[500,0]` describes lower left corner.

3 Reserved words

Color literals: `black white red green blue yellow`

Boolean literals: `true false`

Pre-defined functions: `line rectangle text`

Constant declaration keyword: `let`

4 Language constructs

4.1 Expressions

Expressions are delimited using semicolon.

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

Expressions are following constructs:

- constant declaration
- function call
- scope

4.2 Comments

Single line:

```
// this is comment
```

Part-line / multiline:

```
/* multiline comment */
```

Comments can be nested:

```
/* /* */ */
```

```
/* */ */
```

```
/* /* */
```

Anything inside comments shouldn't break compilation.

4.3 Scopes

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

5 Architecture

5.1 Parser

5.2 Abstract Syntax Tree

5.3 Evaluation

remove comments, eval variables, ifs and for loops

5.4 ZUMA IR

5.5 Translation

ZUMA IR to SVG model

5.6 Generate SVG

6 Performance goals

1ms - good

10ms - acceptable

100ms - bad

1000ms - unacceptable