**Introduction**

EZ-ASCII is an effective programming language for displaying a 2D image with ASCII characters. It allows users to easily create and manipulate ASCII art. In the language, different characters defined in the ASCII set will be used to represent different intensities in images.

The primary purposes of the language include converting images of other format (.bmp, .jpg) to an ASCII format, constructing ASCII format images and performing various manipulations on existing ASCII images. For instance, users will be able to do intensity mapping, a mapping of character intensities to the actual characters they describe.

The language comes objects like canvas. A canvas represents a 2D array of intensities and takes in three parameters: width, height and granularity, of which granularity is an integer and represents the number of intensity levels.

**Primitive Types**

There are four primitive types in E-Z ASCII

boolean： Booleans only have values true and false

int: integers are signed and each has a size of 32 bit

char: a character from ASCII set, surrounded by single quotes ‘’.

string: a sequence of characters surrounded by double quotes “”.

**Arithmetic Operators**

Arithmetics operators are + (addition), - (subtraction), \* (multiplication), %(modulation). All of them are left associative and require their operands to be of the same primitive types.

**Comparison Operators**

Comparison operators are >(greater than), <(less than), == (equal), ~= (not equal), <= (less than or equal to) and >= (greater than or equal to). All of them are left associative and require their operands to be of the same primitive types.

**Arrow Operator**

Arrow operators are left arrow (<-) and right arrow (->). Left arrow is used for variable assignment canvas manipulation. Right arrow is used for canvas output.

**Logical Operators**

Logical operators are && (logical and), || (logical or), ~(logical not). && and || are left associative. They take two boolean type operands and return a boolean value. The operand on the right of ~ must be a boolean value and the return type is also a Boolean value.

**Comma Operator**

A pair of expressions separated by a comma “**,”** is evaluated left-to-right

**Braces**

Braces [] are used to enclose one or more expressions.