Дополнение 1.1

Примечание

Все что написано в предыдущей документации работает. Можно использовать любой формат файла, как старый, так и новый.

Часть 1 Старый формат

Новый функционал, который доступен в старом формате:

Новые операции:

< возвращает 1 если (операнд 1 < операнд 2) иначе 0;

> операция больше;

== 1 если операнды равны, иначе 0;

! отрицание. Если 0, то вернет 1, если что угодно кроме нуля вернет 0;

& логическое И. вернет 1, если оба операнда не ноль

| логическое ИЛИ. Вернет 1, если 1 из операндов не ноль

sig Сигмоида. Принимает на вход два аргумента: 1 – значение, 2 – максимальное значение. Возвращает число от нуля до единицы, замедляется при приближении к максимуму. Может быть полезна при расчете процентов.

Часть 2 Новый формат

Новый формат файла должен иметь расширение .rsr, иначе он не будет читаться.

Добавлен новый тип операнда – gv – означает глобальную переменную

Теперь файл поделен на три блока.

Первая строка файла название библиотеки, оно может быть любым.

Далее идет первый блок:

Каждый блок начинается с числа – количество глобальных операций.

Далее идут непосредственно сами операции в следующем формате:

Имя переменной количество операций

В значение глобальной переменной кладётся результат операции.

Пример:

20 // количество глобальных операций

Fimp 7 // имя переменной (Fimp) количество операций (7)

0 \* c 6.7 f 6 // сами операции

1 \* c 16 f 28

2 \* c 7.7 f 30

3 + v 0 v 1

4 + v 3 v 2

5 / v 4 c 30

6 = v 5 empty 0 // результат заносится в глобальную переменную Fimp

Limp 7 // имя переменной (Limp) количество операций (7)

0 \* c 5.2 f 11

1 \* c 170 f 8

2 \* c 59 f 32

3 + v 0 v 1

4 + v 3 v 2

5 / v 4 c 230

6 = v 5 empty 0

…

Глобальные переменные можно использовать в программе как угодно.

Второй блок:

Второй блок – это блок вычисления процента для каждого символа.

Начало блока также число – количество глобальных операций.

Далее Символ и количество операций для него. Все как и в первом блоке, за исключением того, что каждая глобальная переменная здесь это символ, который будет отображаться в программе, а значение это процент для этого символа. В этом блоке вы также можете использовать любые глобальные переменные, в самом простом случае это выглядит так:

F 1

0 = gv F% 0 0

L 1

0 = gv L% 0 0

…

Третий Блок:

Третий блок – блок вычисления результата для каждого символа (в некоторых алгоритмах результирующий символ вычисляется отдельно от процентов).

Структура такая же: сначала число – количество глобальных операций,

Затем сами глобальные операции, где глобальная переменная это символ, а значение переменной – это значение, которое будет использоваться для получения результата. В программе вычисляется максимальное значение, и в качестве результата выводится символ с максимальным значением.

В самом простом случае это выглядит так:

F 1

0 = gv resF 0 0

L 1

0 = gv resL 0 0

…

Имена функций

Для удобства пользования добавлена возможность пользоваться функциями по именам. Прежний вызов функций по номерам также работает.

Количество пикселей:

f pixelsCount::UP

f pixelsCount::Down

f pixelsCount::Right

f pixelsCount::Left

Lmin / Lmax:

f Lmin/max::CenterMass

f Lmin/max::LeftDown

f Lmin/max::RightDown

f Lmin/max::LeftUp

f Lmin/max::RightUp

f Lmin/max::LeftCenter

f Lmin/max::CenterUp

f Lmin/max::CenterCenter

Координаты:

f CoordinateX::CenterMass

f CoordinateX::LeftDown

f CoordinateX::RightDown

f CoordinateX::LeftUp

f CoordinateX::RightUp

f CoordinateX::LeftCenter

f CoordinateX::CenterUp

f CoordinateX::CenterCenter

f CoordinateY::CenterMass

f CoordinateY::LeftDown

f CoordinateY::RightDown

f CoordinateY::LeftUp

f CoordinateY::RightUp

f CoordinateY::LeftCenter

f CoordinateY::CenterUp

f CoordinateY::CenterCenter

f Circle::CenterUp

f Circle::RightUp

f Circle::LeftCenter

f Circle::CenterCenter

f Circle::CenterDown

Пример использования:

0 / f 2 f 3

Тоже самое:

0 / f pixelsCount::Right f pixelsCount::Left

Работают оба формата функций, пользоваться можно любым.