УДК 004.93
Olga Scherbiena[1], Yuliya Bykova[2]
[1]student of group CST – 121 NU «Zaporizhzhia Polytechnic»
[2]senior teacher NU 'Zaporizhzhia Polytechnic'

WHAT MAKES VELATO AND PIET SO WHY UNIQUE

— Speak YOUR mind! You are as bad as Hamlet!

Do you think that is a quote from one of the Shakespeare's works? No, this is a piece of code on esoteric programming language.

Esoteric programming language (esolang) is a computer programming language created to experiment with weird ideas or as a joke. Esolang is not designed for serious functionality.

Esolangs can be divided into a few broad categories:

1. Minimalism: These languages use as few instructions as possible.

2. Weirdness: Some languages are created mainly for the purpose of being weird and difficult to program in.

3.Themed: Some languages are based on a theme that is not computer related.

4. Brevity: Many esoteric languages are designed to be as short as possible.

5. Jokes: Many esoteric languages are created purely as a joke.

For example, Velato and Piet are two thematic languages about music and drawing.

Velato is esoteric programming language, created by Daniel Temkin in 2009, which uses MIDI files as source code. Commands are determined by the intervals between notes. Velato offers musicians-programmers an unusual task: to compose a piece of music that encrypts the "text".

Velato language has online compiler that lets you experiment with veletta. While researching, we became interested in Schubert's "Mystic Symphony No. 8"and we decided to check if he suddenly left us a text message. So we downloaded the symphony, translated it into mdi format and… our theory failed. Then we checked the audio files that were created on velato (counting rhyme 99 bottles of beer). In this case, we made sure that the compiler works and Schubert did not leave any messages (figure 1).

Commands. All commands in Velato are expressed strictly by the pitch and order of notes. This means that time signature, key signature, measures, rests, repeat marks, even note durations are ignored by the code. Also, additional tracks can be added to the MIDI file, which will be ignored by the compiler.

Any notes from the end of a command to the beginning of the next are interpreted as an expression.

Variables:

1. Any note can be a variable as well as a command, determined only by syntax.

2. Variables can be assigned to any note.

3. Variables can only be called by a command that does not start with (have a command root note) the same note as the variable note. On the first page of the official site you can see the main commands in the form of a table (types, numbers, expressions and other).

The site provides a step-by-step guide to writing a "hello world" program. Unfortunately, the probability that you will write a program is very small.

At this stage, you can notice the first minus of the "Velato" language, it will be difficult for a person who does not have basic musical knowledge to write programs.

To write a program, you need a gmm file, which then needs to be converted to a midi file. If you follow this logic and use the resources that the official site offers, then you can only make a gmm file. Programs that convert a gmm file to midi do not run on the computer. This problem is solvable, but you will have to write a converter from a gmm file to midi yourself or a convenient compiler for (figure2).

This program is unique in its implementation, due to the use of compilers and the modern speed of computers, data decryption will take little time, but encryption a lot.

As for Piet, it is an esoteric programming language in which programs look like abstract paintings. The language is named after Piet Mondrian, who pioneered the field of geometric abstract art.

Piet uses 20 distinct colours. The 18 colours in the first 3 rows of the table are related cyclically in the following two ways:

1. Hue Cycle: red -> yellow -> green -> cyan -> blue -> magenta -> red;

2. Lightness Cycle: light -> normal -> dark -> light.

Piet code takes the form of graphics made up of the recognised colours. Individual pixels of colour are significant in the language.

The basic unit of Piet code is the colour block. A colour block is a contiguous block of any number of codes of one colour bounded by blocks of other colours or by the edge of the program graphic. Blocks of colour adjacent only diagonally are not considered contiguous. A colour block may be any shape and may have "holes" of other colours inside it, which are not considered part of the block.

Piet uses a stack for storage of all data values. Data values exist only as integers, though they may be read in or printed as Unicode character values with appropriate commands (figure 3).

List of basic commands with explanations:

1. push - add to the stack the number of particles from the set of blocks;

2. pop - a useful element that is thrown out of the glass;

3. subtraction, addition, division, multiplication, modification - arithmetic operations are detected with external top manifestations of the stack, the result is launched onto the stack;

4. not - reads the value of the top element in its stack, replaces it with zero (if the value is non-zero) or with one in other cases;

5. greater - improve the upper elements in the amount of two pieces and add 1 to the stack if the first is larger and 0 if the second is larger (figure 4).

Piet language has online compiler too.

Let's say we want to print "Piet". These are the ASCII values we want to print out: 80 105 101 116

When you open the compiler, you will see that there will be hints on each color. Red light is starting position.

1 px is 1 position at ASKIL table.

a) Step 1: we used starting position -> push -> out(char). Starting position, we used 80 times to print "P".

b) Step2: we used beginner position 105 time(to write "i") -> push -> out(char) (figure 5).

c) Step 3 and step 4 similar to step 1 and 2 (figure 6).

As a result, we got the word "Piet" and a rather interesting drawing.

Hence, Piet is pretty easy to learn and you don't need to have any special knowledge to encrypt information. Also Piet is a space for creative people who can draw a cool picture and encrypt the "text".

Data has been encrypted at all times from the ancients (in Egypt) up to World War 2 (Enigma machines and Christopher). We think these languages can be used to create a "Programmer Quest Room".

Also, data can be encrypted under one of the esoteric programming languages and under the Caesar font to ensure data security. It is considered that data

encryption sure develops our logic and thinking and this is the main application of exoteric languages.

Program output:

```
99 BOTTLES OF BEER ON THE WALL
99 BOTTLES OF BEER
TAKE ONE DOWN
PASS IT AROUND
98 BOTTLES OF BEER ON THE WALL

98 BOTTLES OF BEER ON THE WALL
98 BOTTLES OF BEER
TAKE ONE DOWN
PASS IT AROUND
97 BOTTLES OF BEER ON THE WALL

97 BOTTLES OF BEER ON THE WALL
97 BOTTLES OF BEER
TAKE ONE DOWN
PASS IT AROUND
96 BOTTLES OF BEER ON THE WALL

96 BOTTLES OF BEER ON THE WALL
96 BOTTLES OF BEER
```

Figure 1 – Counting rhyme 99 bottles of beer

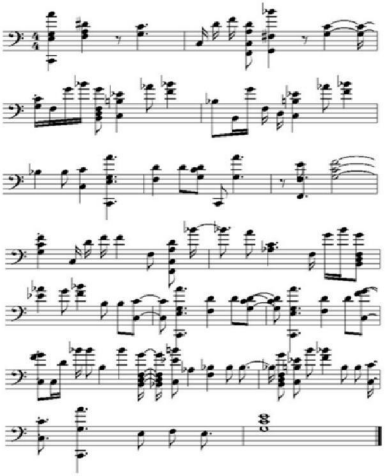

Figure 2 – «Hello, world!»

#FFC0C0 light red	#FFFFC0 light yellow	#C0FFC0 light green	#C0FFFF light cyan	#C0C0FF light blue	#FFC0FF light magenta
#FF0000 red	#FFFF00 yellow	#00FF00 green	#00FFFF cyan	#0000FF blue	#FF00FF magenta
#C00000 dark red	#C0C000 dark yellow	#00C000 dark green	#00C0C0 dark cyan	#0000C0 dark blue	#C000C0 dark magenta
#FFFFFF white			#000000 black		

Figure 3

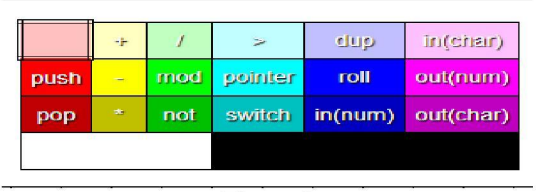

Figure 4

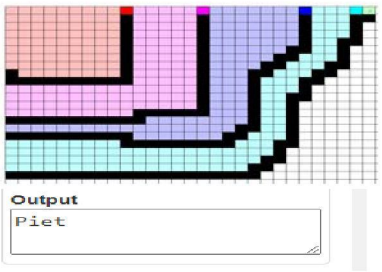

Figure 5

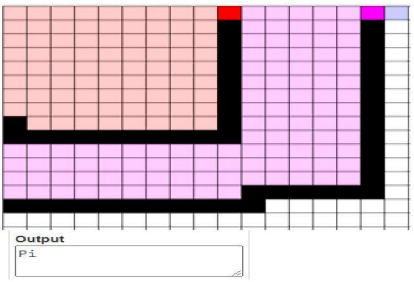

Figure 6 – Result