

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

РАЗРАБОТКА АЛГОРИТМА ДЛЯ ИГРЫ «КАМЕНЬ, НОЖНИЦЫ, БУМАГА»
Отчет по лабораторной работе №2 дисциплины
«Теория автоматов»

Выполнил студент группы ИВТ-21 _____/Рзаев А.Э./
Проверил преподаватель _____/Мельцов В.Ю./

2016 г.

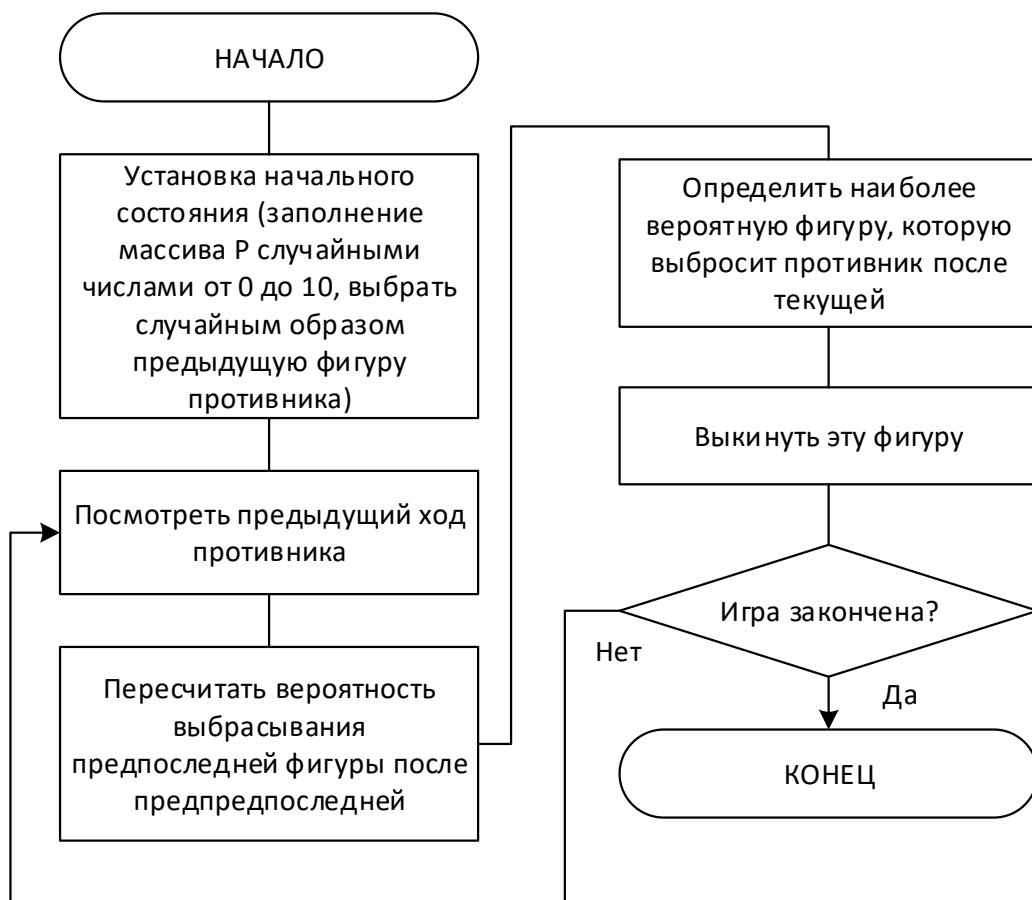
1 Постановка задачи

Необходимо разработать бота для игры «Камень, ножницы, бумага».

2 Разработка алгоритма

За основную идею алгоритма взят вероятностный подход. Алгоритм определяет, какая фигура вероятнее всего будет выброшена противником на следующем ходу, исходя из того, какая была выброшена на предыдущем ходу. Перед первым ходом наиболее вероятные переходы задаются случайным образом. За основу генератора псевдослучайных чисел взята функция `rand` из стандартной библиотеки C.

3 Схема алгоритма работы бота



4 Листинг программы

```
unit RPSBotImpl;

{$mode objfpc}{$H+}

interface

const

    UNKNOWN = 0;
    ROCK = 1;
    SCISSORS = 2;
    PAPER = 3;

    procedure setParams(setCount, winsPerSet: Integer);
    function choice(previousRivalChoice: Integer): Integer;
    procedure onCurrentGameEnd;

    // helper functions
    function produceChoice(rivalP: Integer): Integer;
    procedure srand(s: Longword);
    function rand: Integer;

implementation

var
    P: array [1..3, 1..3] of Longword;
    prevPrevRival, rnd: Integer;

//pseudorandom generator
const RAND_MAX = 32767;
var
    seed: Longword;

procedure srand(s: Longword);
var
    tmp: ^Longword;
begin
    New(tmp);
    seed := tmp^ xor s;
    Dispose(tmp);
end;

function rand: Integer;
var
    next, res: Longword;
begin
    next := seed;

    next := next * 1103515245;
    next := next + 12345;
    res := (next div 65536) mod 2048;

    next := next * 1103515245;
    next := next + 12345;
    res := res shl 10;
    res := res xor ((next div 65536) mod 1024);

    next := next * 1103515245;
```

```

    next := next + 12345;
    res := res shl 10;
    res := res xor ((next div 65536) mod 1024);

    seed := next;

    rand := res and RAND_MAX;
end;
//

function maxP(choice: Integer): Integer;
var
    i, m : Integer;
begin
    m := 1;
    for i := 2 to 3 do
        if P[choice][m] < P[choice][i] then
            m := i;
        maxP := m;
    end;

function produceChoice(rivalP: Integer): Integer;
begin
    if rivalP = ROCK then
        produceChoice := PAPER
    else if rivalP = SCISSORS then
        produceChoice := ROCK
    else
        produceChoice := SCISSORS;
end;

procedure setParams(setCount, winsPerSet: Integer);
var
    i, j: Integer;
begin
    srand(9931);

    prevPrevRival := rand mod 3 + 1;

    for i := 1 to 3 do
        for j := 1 to 3 do
            P[i][j] := (rand mod 10 + 1) xor rnd;
        end;
    end;

function choice(previousRivalChoice: Integer): Integer;
begin
    if previousRivalChoice = UNKNOWN then
        previousRivalChoice := rand mod 3 + 1;
    Inc(P[prevPrevRival][previousRivalChoice]);
    prevPrevRival := previousRivalChoice;
    choice := produceChoice(maxP(previousRivalChoice));
    Inc(rnd);
end;

procedure onCurrentGameEnd;
begin
end;

end.

```