

Laporan Tugas 1

Struktur Data dan Analisa Algoritma

Mochammad Hilmi Rusydiansyah
5024211008

September 21, 2022

1 Pendahuluan

Laporan ini dibuat untuk melengkapi tugas pertama dalam mata kuliah Struktur Data dan Analisa Algoritma Kelas (A). Dalam tugas 1 ini meliputi proposal pembuatan game, source code implementasi array abstract dalam bentuk oop, dan source code sistem game pesawat yang sudah dimodifikasi arah geraknya.



Figure 1: Logo C++

2 Penjelasan Game

Dalam game ini, anda akan berperan sebagai penjaga kebun binatang. Suatu saat terjadi kebocoran zat kimia yang terletak di dekat kebun binatang. Kebocoran tersebut mengakibatkan hewan-hewan terkontaminasi zat kimia dan menjadi ganas, sehingga banyak dari hewan-hewan tersebut yang lepas dari kandang. Kondisi menjadi kacau, anda sebagai penjaga kebun binatang pun harus berusaha menghindari kejaran sambil memberi makan hewan-hewan yang ganas tersebut.

3 Cara Bermain

- Mekanisme game yang digunakan adalah tembak-tembak dengan perspektif kamera Top-down
- Pemain harus menembakkan peluru ke arah target yang terus bergerak mengejar pemain itu sendiri. Dengan kata lain, pemain harus melempar makanan ke arah hewan-hewan yang terus berusaha mengejar pemain itu sendiri
- Apabila peluru makanan tepat mengenai hewan sasaran, maka hewan tersebut akan berhenti mengejar dan sebaliknya memakan peluru makanan tadi
- Pemain harus berusaha mendapatkan skor tertinggi dengan bertahan selama mungkin
- Pemain memiliki bar darah, apabila hewan berhasil mencapai dan menyerang pemain, darah pemain akan berkurang
- Pemain dinyatakan kalah apabila bar darah telah habis
- Akan ada drop item seperti darah, peluru, dan lain-lain untuk meningkatkan keseruan dalam bermain

Source Code Array Abstract with OOP

```
1 #include <stdio.h>
2
3 class ClArray{
4 private:
5     float data[100];
6     int Size;
7     int Length;
8
9 public:
10    ClArray(){
11        Size = 100;
12        Length = 0;
13    }
14
15    void Append(float Vi){
16        if(Length < Size){
17            Length++;
18            data[Length-1]=Vi;
19        }
20    }
21
22    void Display(){
23        for(int i=0; i<Length; i++){
24            printf("%f, ",data[i]);
25        }
26        printf("\n");
27    }
28
29    void Insert(int index, float vi){
30        if(ClArray::Length < ClArray::Size){
31            for(int i=ClArray::Length; i>index; i--){
32                ClArray::data[i] = ClArray::data[i-1];
33            }
34            ClArray::data[index] = vi;
35            ClArray::Length++;
36        }
37    }
38
39    void Delete(int index){
40        if((ClArray::Length > 0) && (index >= 0)){
41            for(int i=index; i<ClArray::Length; i++){
42                ClArray::data[i] = ClArray::data[i+1];
43            }
44            ClArray::Length--;
45        }
46    }
47
48    int Search(float vi){
49        int result = -1;
50        for (int i=0; i<ClArray::Length; i++){
51            if(ClArray::data[i] == vi){
```

```

52         result = i;
53         break;
54     }
55 }
56 return result;
57 }
58
59 float Get(int index){
60     return ClArray::data[index];
61 }
62
63 void Set(int index, float vi){
64     ClArray::data[index] = vi;
65 }
66
67 float Max(){
68     float maxVal = ClArray::data[0];
69     for(int i=1; i<ClArray::Length; i++){
70         if(ClArray::data[i] > maxVal){
71             maxVal = ClArray::data[i];
72         }
73     }
74     return maxVal;
75 }
76
77 float Min(){
78     float minVal = ClArray::data[0];
79     for(int i=1; i<ClArray::Length; i++){
80         if(ClArray::data[i] < minVal){
81             minVal = ClArray::data[i];
82         }
83     }
84     return minVal;
85 }
86
87 void Reverse(){
88     float temp;
89     for(int i=0, j=ClArray::Length-1; i<ClArray::Length/2; i++, j--){
90         temp = ClArray::data[i];
91         ClArray::data[i] = ClArray::data[j];
92         ClArray::data[j] = temp;
93     }
94 }
95
96 void Shift(char c, int n){ // c for 'l' or 'r', n for how many steps
97     // of the shift algorithm
98     float temp;
99     if((c == 'l') || (c == 'r')){
100         for(int i=0; i<n; i++){ // looping for how many one-step
101             // shift will be performed
102             temp = ClArray::data[0]; // save the first data index
103             for(int j=0; j<ClArray::Length-1; j++){
104                 ClArray::data[j] = ClArray::data[j+1];

```

```

103         }
104         ClArray::data[ClArray::Length-1] = temp; // insert the
temp
105     }
106 }
107     else if((c == 'r') || (c == 'R')){
108         for(int i=0; i<n; i++){ // looping for how many one-step
shift will be performed
109             temp = ClArray::data[ClArray::Length-1]; // save the
last data index
110             for(int j=ClArray::Length-1; j>0; j--){
111                 ClArray::data[j] = ClArray::data[j-1];
112             }
113             ClArray::data[0] = temp; // insert the temp
114         }
115     }
116     else{
117         printf("\nERROR !!! invalid parameter...\n");
118     }
119 }
120 };
121
122 int main(){
123     class ClArray d;
124     d.Append(1);
125     d.Append(3);
126     d.Append(5);
127     d.Append(7);
128     d.Display();
129     printf("\n===== \n");
130
131     printf("\nInsert -1 di index ke 2\n");
132     d.Insert(2, -1);
133     d.Display();
134     printf("\nDelete data di index ke 2\n");
135     d.Delete(2);
136     d.Display();
137     printf("\n===== \n");
138
139     printf("\nSearch data yang bernilai 3\n");
140     int n = d.Search(3);
141     if(n == -1){
142         printf("Data tidak ditemukan\n");
143     } else{
144         printf("Data bernilai 3 ada di index ke : %d\n", n);
145         d.Display();
146
147         printf("\nDelete data di index ke %d\n", n);
148         d.Delete(d.Search(3));
149         d.Display();
150     }
151     printf("\n===== \n");
152 }

```

```

153     int idx = 2;
154     printf("\nAmbil nilai dari index ke %d ---> ", idx);
155     printf("%f\n", d.Get(idx));
156     printf("\nEdit data pada index ke %d menjadi bernilai 21\n", idx);
157     d.Set(idx, 21);
158     d.Display();
159     printf("\n===== \n");
160
161     printf("\n");
162     d.Display();
163     printf("\nNilai maksimum : %f", d.Max());
164     printf("\nNilai minimum : %f\n", d.Min());
165     printf("\n===== \n");
166
167     printf("\nTambah beberapa data...\n");
168     d.Append(9);
169     d.Insert(2, 55);
170     d.Insert(3, 11);
171     d.Append(99);
172     d.Display();
173     printf("\nReverse array\n");
174     d.Reverse();
175     d.Display();
176     printf("\n===== \n");
177
178     printf("\n");
179     d.Display();
180     printf("Shift 1 step to the left\n");
181     d.Shift('l', 1);
182     d.Display();
183     printf("Shift 3 step to the right\n");
184     d.Shift('r', 3);
185     d.Display();
186     printf("\n===== \n");
187
188     return 0;
189 }

```

Source Code for Edit Plane Movement Flows

```
1 #include<graphics.h>
2 #include<conio.h>
3 #include<dos.h>
4 #include<stdio.h>
5 #include<cmath>
6
7 #define PI 3.14159265359
8
9 class Pesawat{
10 private :
11     float x;
12     float y;
13     float sc; // scale
14     int Xmax ; // screen x max
15     int Ymax; // screen y max
16     int sty; // determine plane speed y
17     int stx; // determine plane speed x
18     float body[5][2] = {{0,0}, {1,1}, {0.5,5}, {-0.5,5}, {-1,1}}; // for
drawing body plane
19     float wing[4][2] = {{3,1}, {3,2}, {-3,2}, {-3,1}}; // for drawing
wing plane
20
21 public:
22     Pesawat(){
23         sc = rand()%20 + 5;
24         Xmax = getmaxx();
25         Ymax = getmaxy();
26         sty = rand()%20 + 4;
27         stx = rand()%20 + 4;
28         x = rand()%Xmax + 10;
29         y = rand()%Ymax + 10;
30     }
31
32     void Posisi(float xi, float yi){
33         x = xi;
34         y = yi;
35     }
36
37     void Skala(float scli){
38         sc = scli;
39     }
40
41     void MoveUp(){
42         y = y - sty;
43         if (y < 0){
44             y = y + Ymax;
45         }
46     }
47
48     void MoveRight(){
49         x = x + stx;
```

```

50     if (x > Xmax){
51         x = x - Xmax;
52     }
53 }
54
55 void RotatePesawat(float degree){
56     // convert degree to radian
57     float rad = degree * (PI/180);
58
59     // rotating body
60     for(int i=0; i<5; i++){
61         float temp[2]; // for storing one-point coordinate
62         // assign to temp
63         for(int j=0; j<2; j++){
64             temp[j] = body[i][j];
65         }
66         // convert cartesian to polar (r, theta)
67         float r = sqrt( pow(temp[0],2) + pow(temp[1],2) );
68         float theta = atan2(temp[1], temp[0]);
69
70         // adding theta by degree(radian) inputted
71         theta = theta + rad;
72
73         // convert polar to cartesian again and store it to temp
74         temp[0] = r * cos(theta); // as x
75         temp[1] = r * sin(theta); // as y
76
77         // return temp to body_array
78         for(int j=0; j<2; j++){
79             body[i][j] = temp[j];
80         }
81     }
82
83     // rotating wing
84     for(int i=0; i<4; i++){
85         float temp[2]; // for storing one-point coordinate
86         // assign to temp
87         for(int j=0; j<2; j++){
88             temp[j] = wing[i][j];
89         }
90         // convert cartesian to polar (r, theta)
91         float r = sqrt( pow(temp[0],2) + pow(temp[1],2) );
92         float theta = atan2(temp[1], temp[0]);
93
94         // adding theta by degree(radian) inputted
95         theta = theta + rad;
96
97         // convert polar to cartesian again and store it to temp
98         temp[0] = r * cos(theta); // as x
99         temp[1] = r * sin(theta); // as y
100
101         // return temp to wing_array
102         for(int j=0; j<2; j++){

```



```

103         wing[i][j] = temp[j];
104     }
105 }
106
107
108 void DrawPesawat(){
109     // drawing body of plane
110     line((body[0][0]*sc + x), (body[0][1]*sc + y), (body[1][0]*sc + x
111 ), (body[1][1]*sc + y));
112     line((body[1][0]*sc + x), (body[1][1]*sc + y), (body[2][0]*sc + x
113 ), (body[2][1]*sc + y));
114     line((body[2][0]*sc + x), (body[2][1]*sc + y), (body[3][0]*sc + x
115 ), (body[3][1]*sc + y));
116     line((body[3][0]*sc + x), (body[3][1]*sc + y), (body[4][0]*sc + x
117 ), (body[4][1]*sc + y));
118     line((body[4][0]*sc + x), (body[4][1]*sc + y), (body[0][0]*sc + x
119 ), (body[0][1]*sc + y));
120
121     // drawing wing of plane
122     line((wing[0][0]*sc + x), (wing[0][1]*sc + y), (wing[1][0]*sc + x
123 ), (wing[1][1]*sc + y));
124     line((wing[1][0]*sc + x), (wing[1][1]*sc + y), (wing[2][0]*sc + x
125 ), (wing[2][1]*sc + y));
126     line((wing[2][0]*sc + x), (wing[2][1]*sc + y), (wing[3][0]*sc + x
127 ), (wing[3][1]*sc + y));
128     line((wing[3][0]*sc + x), (wing[3][1]*sc + y), (wing[0][0]*sc + x
129 ), (wing[0][1]*sc + y));
130 }
131 };
132
133 int main()
134 {
135     initwindow(500,500);
136     Pesawat P;
137     P.Skala(20);
138     Pesawat M[20];
139     float xh,yh;
140     char c;
141
142     // harus dijalankan cuma sekali
143     P.RotatePesawat(90);
144     for(int i=0; i<20; i++){
145         M[i].RotatePesawat(90);
146     }
147
148     do{
149         cleardevice();
150         if(kbhit()){ // keyboard hit
151             c = getch(); // get button of pressed keyboard
152             if (c == 27) break;
153         }
154
155         xh = mousex();

```

```

147     yh = mousey() ;
148     P.Posisi(xh, yh);
149     P.DrawPesawat();
150
151     // move pesawat ke atas
152     // for(int i=0; i<20; i++){
153     //     M[i].MoveUp();
154     //     M[i].DrawPesawat();
155     // }
156
157     // move pesawat ke kanan
158     for(int i=0; i<20; i++){
159         M[i].MoveRight();
160         M[i].DrawPesawat();
161     }
162
163     delay(50); // delay 50ms each iteration
164 } while(1);
165
166 getch();
167 closegraph();
168 return 0;
169 }

```