Flappy Bird Pre-Alpha Costin Costache

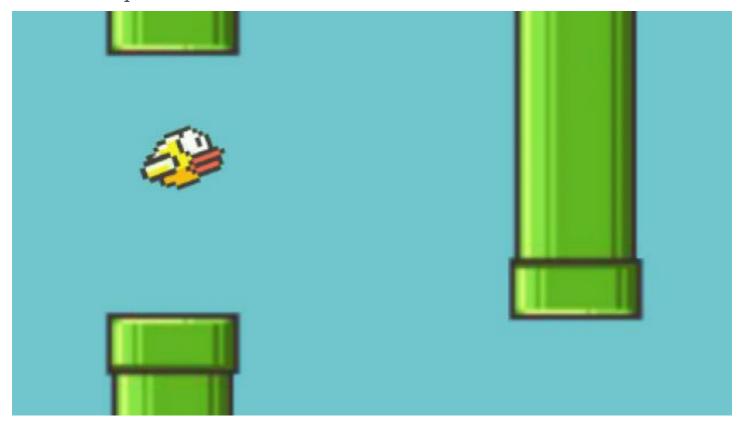
Grupa #253

Prof. coord:

- Stupariu Mihai Sorin
- Iolu Alexandra

Descriere

Tema aleasa este o reprezentare simplificata a celebrului joc Flappy Bird, care a avut peste 1 miliard de descarcari :



Scopul jocului este simplu: trebuie sa ajungi cat mai departe fara sa te lovesti de obstacole. Daca pierzi, jocul re-incepe, progresul nefiind salvat.

Instructiuni joc

Click stanga - zboara in sus Click dreapta - zboara in jos

Logica jocului

Pentru ca este o versiune simplificata, exista doar un singur nivel, scopul fiind ca pasarea sa ajunga din partea stanga a ecranlui in partea dreapta. Pentru acest lucru, se seteaza coordonatele de inceput, x_tr fiind cordonata de pe axa X a punctului din stanga jos a patratului, iar y tr de pe axa Y.

```
double x_tr = -500.0;
double y_tr = -150;
```

si conditiile de victorie / infrangere:

```
void youLost()
{
     glutDestroyWindow(1);
     cout << "You lost";
     //cin >> x_tr;
}
```

```
void youWon()
{
```

```
glutDestroyWindow(1);
cout << "CONGRATZ, YOU WON !!!";
cin >> x_tr;
}
```

Obstacolele din joc sunt definite astfel:

```
struct obstacles {
    // type 1 => sus
    // type 0 => jos
    double x_left, x_right, height, type;
} obstacles;
```

Deplasarea se face constant catre dreapta, acest lucru neputand fi modificat de catre jucator.

```
x_tr += 2;
```

Cand se apasa un click, se intampla 2 lucruri:

- se schimba unghiul la care este rotita pasarea, pentru a indica directia in care se indreapta
- se schimba orientarea pe axa Y

```
void mouse(int button, int state, int x, int y)
{
    switch (button) {
```

```
case GLUT RIGHT BUTTON:
       if (state == GLUT DOWN)
       {
           alpha = -x tr move speed;
       }
       glutIdleFunc(miscas);
       break;
   case GLUT_LEFT_BUTTON:
       if (state == GLUT_DOWN)
          alpha = x tr move speed;
       }
       glutIdleFunc(miscad);
       break;
   default:
     break;
   }
void miscad(void)
   y_tr += alpha;
   rotate_angle = 5;
   glutPostRedisplay();
}
```

Verificarea coliziunilor se face astfel:

```
for (jj = 1; jj <= nr_obstacles; jj++)</pre>
    {
        if (x_tr + precision_pixel >= obstacles[jj].x_left &&
 x_tr <= obstacles[jj].x_right)</pre>
         {
             if (obstacles[jj].type)
             {
                 if (y_tr + precision_pixel >= obstacles[jj].h
eight)
                 {
                      youLost();
                 }
             }
             else
             {
                 if (y_tr <= obstacles[jj].height)</pre>
                 {
                      youLost();
```

Bibliografie:

• suport curs

- exemple model moodle
- https://msdn.microsoft.com/en-us/library/