

Flappy Bird

Проект по Визуелно Програмирање

Бојан Секулоски 191263

1. Цел

Проектот преставува рекреација на доста позната игра наречена “Flappy Bird”. Целта на играта е птицата да се лета низ мали дупки, што е можно подалеку. Доколку играчот се удри во препреките, падне на земјата, или прелета многу високо, играта завршува и играчот мора да почне одпоново. За време на играњето се собираат поени и брзината на играта се накачува со тек на време. Оригиналната игра беше позната по тоа колку е тешка и напорна и тоа во целост е следено во овој проект.

2. Решение

Решението се наоѓа во единствената класа во проектот, Form1. Разделување на кодот на повеќе класи е непотребно и би предизвикал прајќање на варијабли од класа до класа секоја милисекунда, непотребна трансакција. Текот на играта е воден од движењето на препреките на лево, од кои има 2 пара по 4. Откако ќе излезат од екранот, истите се вратени назад на различни висини, правејќи ја играта бесконечна. Единственото движење на играчот е по Y оската контролирано од едноставна симулација на гравитација и функција за скок. Функцијата се активира на секој клик на левото копче на глумчето или со користење на стрелката за нагоре. Многу малце податоци се чуваат и тоа, сегашниот резултат, како и највисок резултат, сочуван во локална датотека. Играта е всушност многу едноставна и затоа се користат само примитивни типови. Играта исто така е балансирана преку deltaTime променлива чија цел е да ја води брзината на играта и да ја држи брзината идентична на сите уреди со цел да нема разлика поради моќноста на уредите.

3. Функции

1. ApplyGravity

ApplyGravity е функцијата која овозможува летање на птицата. Таа е најбитната функција бидејќи ја претставува суштината на играта. Функцијата е многу едноставна. На почеток ја сетира променливата за забрзување. На неа и го додава производот од променливата за гравитација и променливата deltaTime. Доколку ја изоставиме втората променлива, на послаби уреди забрзувањето ќе биде помало, додека на посилни ќе биде поголемо. Понатаму има лимит, доколку забрзувањето ја стигне вредноста 0.6, тука ќе застане и нема веќе да се накачува. И на крај ова забрзување се додава на сегашната позиција на птицата на Y оската.

```
private void ApplyGravity()
{
    velocity += gravity * deltaTime;
    if (velocity > 0.6)
    {
        velocity = 0.6;
    }
    bird.Top += (int)(velocity * deltaTime);
}
```

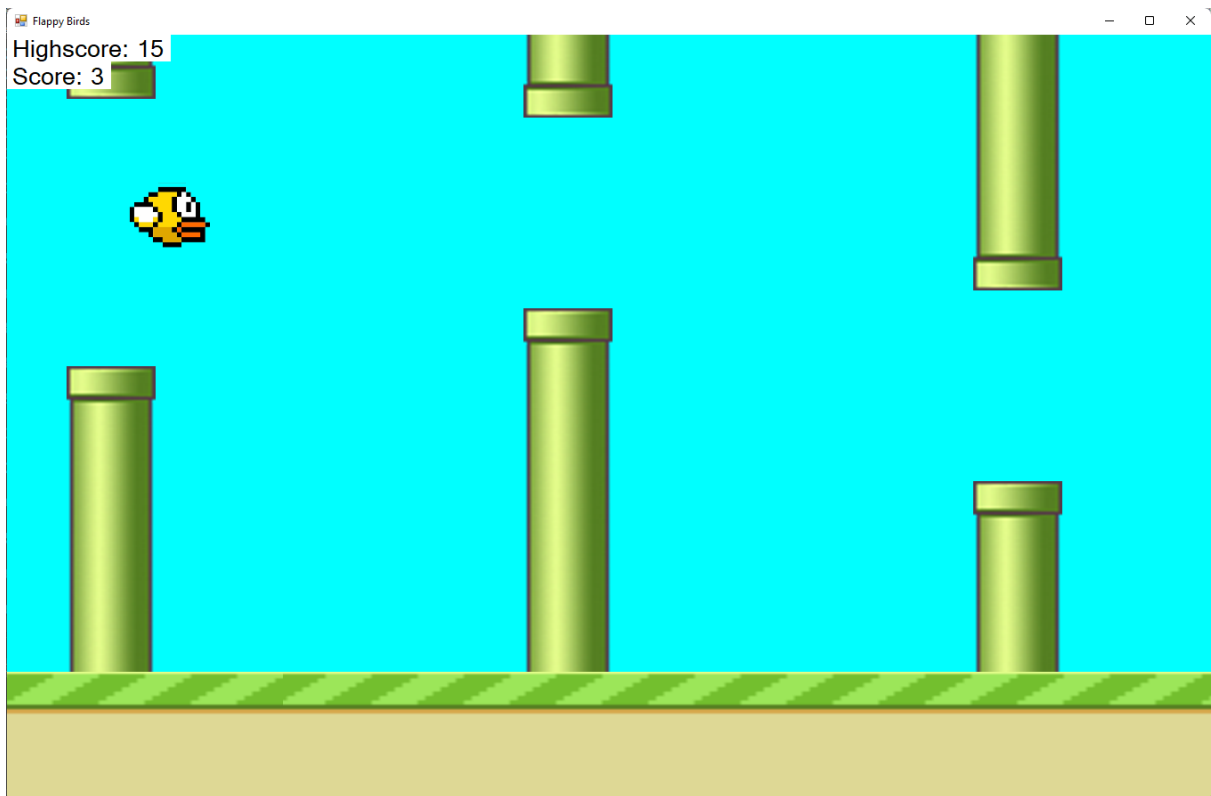
2. MovePipes

Втората главна функција ги придвижува препреките. Секој тик на тајмерот, таа одзема од позицијата на сите препреки (3 од нив не се наведени тука, кодот е идентичен) со помош на speed променливаја која ја одредува брзината на играта. Доколку позицијата на препреката стане помала од -180, таа е вратена назад на различна висина и се накачуваат поените. Со ова играта станува бесконечна.

```
private void MovePipes()
{
    pipeTop1.Left -= (int)(speed * deltaTime);
    pipeBottom1.Left -= (int)(speed * deltaTime);
    if(pipeBottom1.Left < -180)
    {
        pipeBottom1.Left = 1800;
        pipeTop1.Left = 1800;
    }
}
```

```
newHeight = random.NextDouble() * 248 + 276;  
pipeBottom1.Top = (int)newHeight;  
pipeTop1.Top = (int)newHeight - 714;  
UpdateScore();  
}  
...
```

4. Упатсво и пример



Горенаведената слика претставува пример од изгледот на играта. Во горниот ќош се наоѓаат сегашниот резултат и највисокиот резултат а остатокот од екранот е покриен од препреките и птицата. Играта се игра со помош на левото копче на глумчето или стрелката за нагоре, со што птицата скока нагоре и се “движи напред”.