

ТИМСКИ ПРОЕКТ ПО ВИЗУЕЛНО ПРОГРАМИЊЕ

Тема: Race of Death

Изработиле:

Мерџан Бајрами 181060

Хамза Јашари 185062

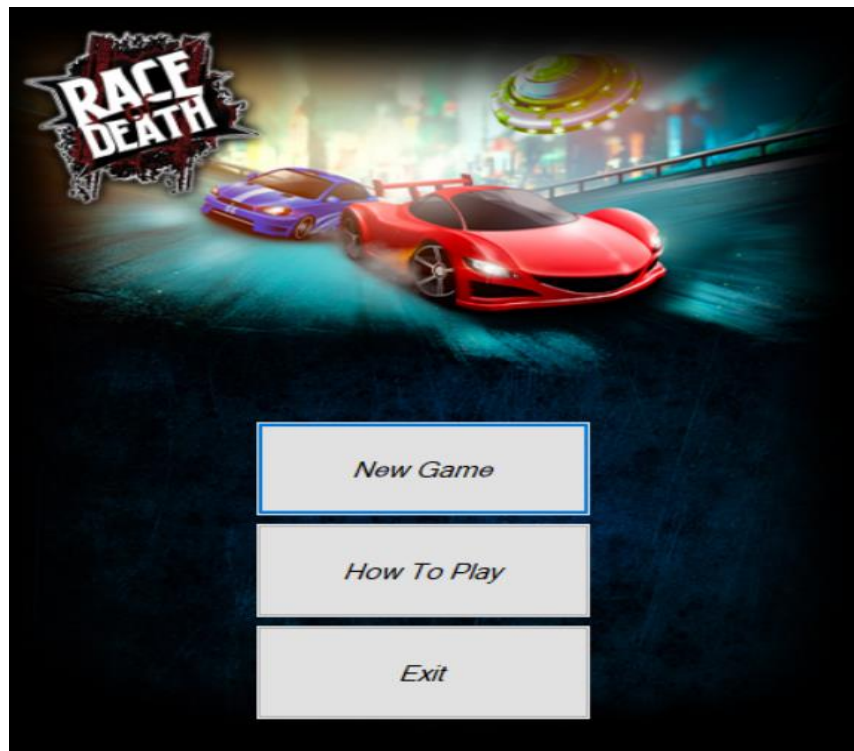
Мухамет Османи 181093

1. Опис на апликацијата

Апликацијата што ја развивавме е 2D игра, трка со автомобили која се вика Race of Death. Играта не е статична (нема одредени нивоа) туку зависи од самиот играч, колку сака да игра и да постигнува добар резултат. Исто така ги има сите функционалности на една 2D игра.

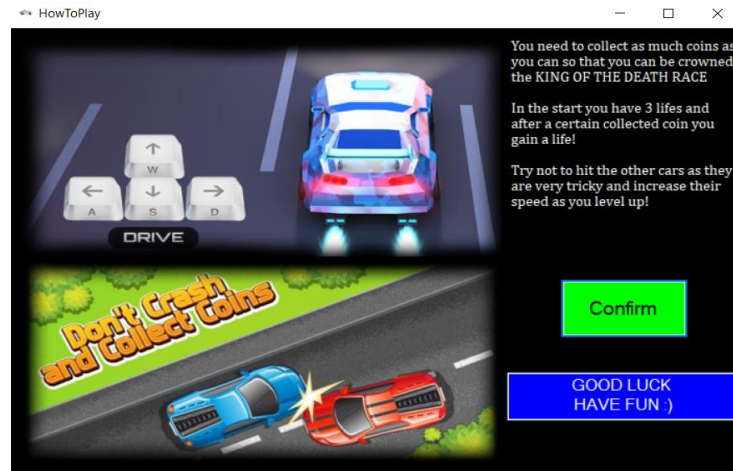
2. Упатство за користење

Во почеток има главен прозорец(слика 1) во кој има 3 копчиња што доведуваат до нови прозореци(форми). Со клик на копчето New Game играта започнува, но соодветно е прво да се прочитат правилата на играта.



2.1 Правила и упатства

Во почетниот прозорец по притискање на копчето How To Play се отвара нов прозорец(слика2) во кој е опишано целта на играта, правилата на самата игра и после тоа играчот може да почне со играта.

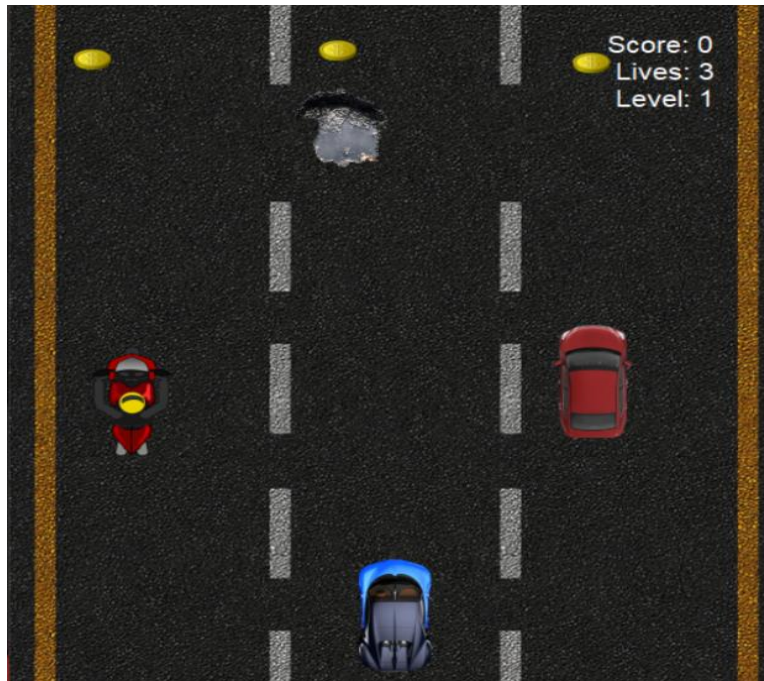


Правилата се едноставни и лесни за паметење. Играта има нивоа кои се неограничени. Почнувате со 3 животи, можете да возете со W; A; S; D или keyUp, keyDown, keyRight, keyLeft. Играчот треба да ги избегнува непријателските автомобили и дупките покрај патот, бидејќи тие им одземаат еден живот.

2.2 Детален опис на играта

Покрај патот ќе се соочите со монети кои треба да ги соберете. Монетите ви помагаат на два начина, односно првиот ви го одредува вашиот резултат, додека вториот начин на помагање е откако ќе соберете 10 монети ќе добиете нов дополнителен живот, но под услов да имате помалку од 3 животи.

Играта е дизајнирана на тој начин, што колку повеќе ја играте, толку потешко станува за да ги завршите нивоата. Ова се случува затоа што откако ќе се помине првото ниво моторот се менува со камион а со тоа станува потешко да се земат монетите. Исто така после секое минато ниво се зголемува брзината на непријателските автомобили и станува потешко да се вози (не се удираа со непријателски автомобили).



3. Преставување на проблемот

Променливите и сите функции кои се употребени содржаат коментари каде детално се објаснуваат. Кодот е организиран во повеќе класи и со повеќе функции, во главната форма Form1 се употребуваат објекти од овие класи.

- **Класата Player:** Во оваа класа се декларирани променливите за ниво-то, животите и get set методите. Исто така во ова класа се чуваат функциите кои овозможуваат движење на колата(на играчот) во сите насоки. Во maxwidth и maxheight променливите се чувани димензиите на самата форма.

```

Race of Death
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

public void moveRight()
{
    if (car.getX() < maxwidth - 35)
    {
        car.setX(car.getX() + car.getspeed());
    }
}

1 reference
public void moveLeft()
{
    if (car.getX() >= 40)
    {
        car.setX(car.getX() - car.getspeed());
    }
}

1 reference
public void moveUp()
{
    if (car.getY() >= 10)
    {
        car.setY(car.getY() - car.getspeed());
    }
}

1 reference
public void moveDown()
{
    if (car.getY() < maxheight - 10)
    {
        car.setY(car.getY() + car.getspeed());
    }
}

```

- **Класата Background:** Ова класа ја создадовме само за сликата во позадина да може да се повторува и за апликацијата да тече непречено. Backgroundmove () функцијата ја овозможуваа тоа.
- **Класата Car:** Ја создадовме ова класа за да ги чува податоците за непријателските автомобили. Конструкторот прима: типот на автомобилот дефиниран со enum, брзината која е променливо и почетните координати (X,Y). DrawImage() методот служи за цртање на колите.

Колизијата меѓу колата на играчот и непријателските коли или пак дупката ја решивме употребувајќи вградената класа во C# Rectangle, така што во овој код кога колата ќе се приближуваа до другите коли ќе се повикат другите функции кои ќе се справаат со колизијата.

```
1 reference
private void Start_Tick(object sender, EventArgs e)
{
    Invalidate();

    Rectangle playerRec = new Rectangle(player.getCar().getX(), player.getCar().getY(),
        player.getCar().getWidthcar(), player.getCar().getHeight());
    foreach (Car enemy in enemyCars)
    {
        Rectangle enemyRec = new Rectangle(enemy.getX(), enemy.getY(), enemy.getWidthcar(),
            enemy.getHeight());

        //Check if a collision has accured
        if (playerRec.Intersects(enemyRec))
        {
            updateCarPosition(enemy);
            collision();
        }
    }
    Rectangle pitRec = new Rectangle(pit.getX(), pit.getY(), pit.getWidth(), pit.getHeight());
    if (playerRec.Intersects(pitRec))
    {
        pit.setY(-130);
        int x = rand.Next(panelmid.Location.X, panelmid.Location.X + panelmid.Size.Width - 60);
        pit.setX(x);
        collision();
    }
    background.backgroundmove(backflag);
    background1.backgroundmove(backflag);
    Start.Enabled = true;
}
```

Истата логика ја спроведовме и за монетите кои треба да ги собере играчот. Во рамките на формата поставивме и 4 тимери кои служат со движење на другите коли, дупките и монетите и самата игра.

Collectcoin() – функцијата осигурува дека во моментот кога играчот ќе земе монета ќе му се зголеми резултатот а исто така проверува дали условот е исполнет за да зголеми нивото.

```
//Function to collect the coins and to call the function updateLevel() when the condition is m
1 reference
public void collectCoin(Coin myCoin)
{
    player.increaseScore();
    //in every 10 coins collected increase the level
    if (player.getCoins()%10==0)
    {
        updateLevel();
    }
    if(player.getCoins()%10==1)
    {
        updateLevellb.Visible = false;
    }
    myCoin.setY(-50);
    //myCoin.setX(rand.Next(0, this.ClientSize.Width));
    updateLabel();
}
```

GenerateCoins() – Во ова функција употребувавме податочна структура листа од тип од класата Coin, тука ги позициониравме монетите рандом.

```
//Generate coins after they're taken or run out of Form Window
1 reference
public void generateCoins()
{
    coins = new List<Coin>();
    int x, y;
    x = rand.Next(panelleft.Location.X, panelleft.Location.X + panelleft.Size.Width - 120);
    y = rand.Next(-20,0);
    Coin coin1 = new Coin(x, y);
    x = rand.Next(panelright.Location.X, panelright.Location.X + panelright.Size.Width - 120);

    Coin coin2 = new Coin(x,y);

    x = rand.Next(panelmid.Location.X, panelmid.Location.X + panelmid.Size.Width - 120);
    Coin coin3 = new Coin(x,y);
    coins.Add(coin1);
    coins.Add(coin2);
    coins.Add(coin3);
}
```

updateLevel() – Со повикување на ова функција на играчот му се дава уште еден живот, но ова функција се повикува само доколку играчот успеал да освои најмалку 10 монети. После тоа се зголеми ниво-то, му се намалуваат вкупните монети(резултатот) за 20 и се зголеми брзината на непријателските коли за рандом број.

```
//After the player has collected certain coins he gains a life only if he has less than 3 lifes(starting condition)
//Also increases the speed of the enemy for each level that he completes
1 reference
public void updateLevel()
{
    if(player.getLives() < 3)
    {
        player.setLife(player.getLives() + 1);

        /*if (player.getCoins() < 20)
        {
            player.setCoins(0);
        }*/
        player.setCoins(player.getCoins() - 20);
    }
    player.setLevel(player.getLevel()+1);
    updateLevellb.Text = String.Format("Congratzz! \n New level is unlocked");
    updateLevellb.Visible = true;
    truckFlag = true;
    enemy1.setspeed(rand.Next(enemy1.getspeed(), enemy1.getspeed() + 15));
    /* enemy2.setspeed(rand.Next(enemy2.getspeed(), enemy2.getspeed() + 14));*/
    enemy3.setspeed(rand.Next(enemy3.getspeed(), enemy3.getspeed() + 13));
}
```