

Interaktivna multimedija, prvi sklop vaj

Naloga 1 – ogrevanje z vesoljsko ladjo

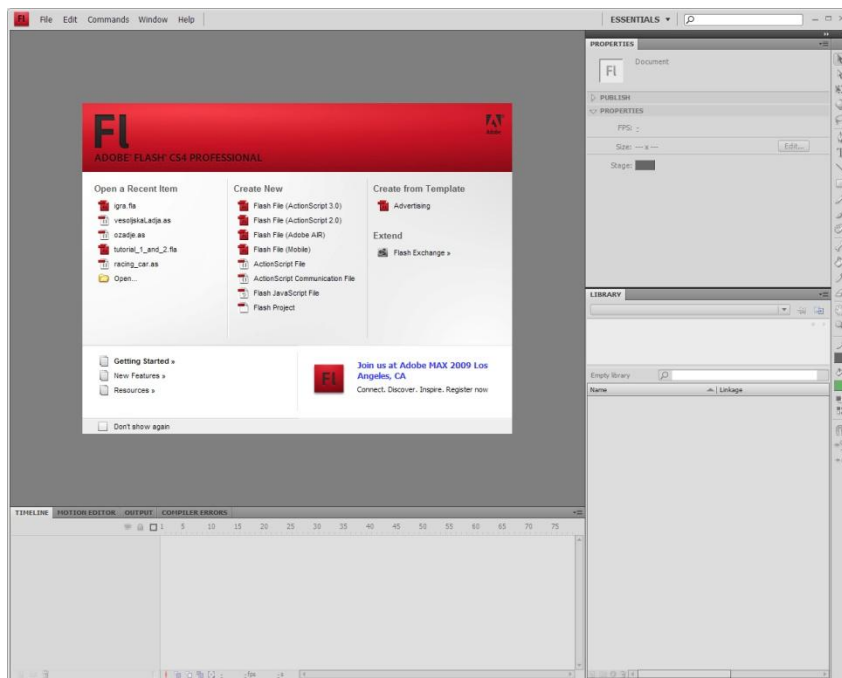
Na voljo imamo nedokončano igrico – vesoljsko ladjo, ki leti nad mestom. Mesto nad katerim leti ladja se samodejno premika v levo, kar ustvari iluzijo nenehnega premikanja ladje v desno. Ladjo lahko tudi dodatno premikamo s pomočjo smernih tipk na tipkovnici.

Mi bomo izboljšali premikanje ozadja in upravljanje z ladjo.

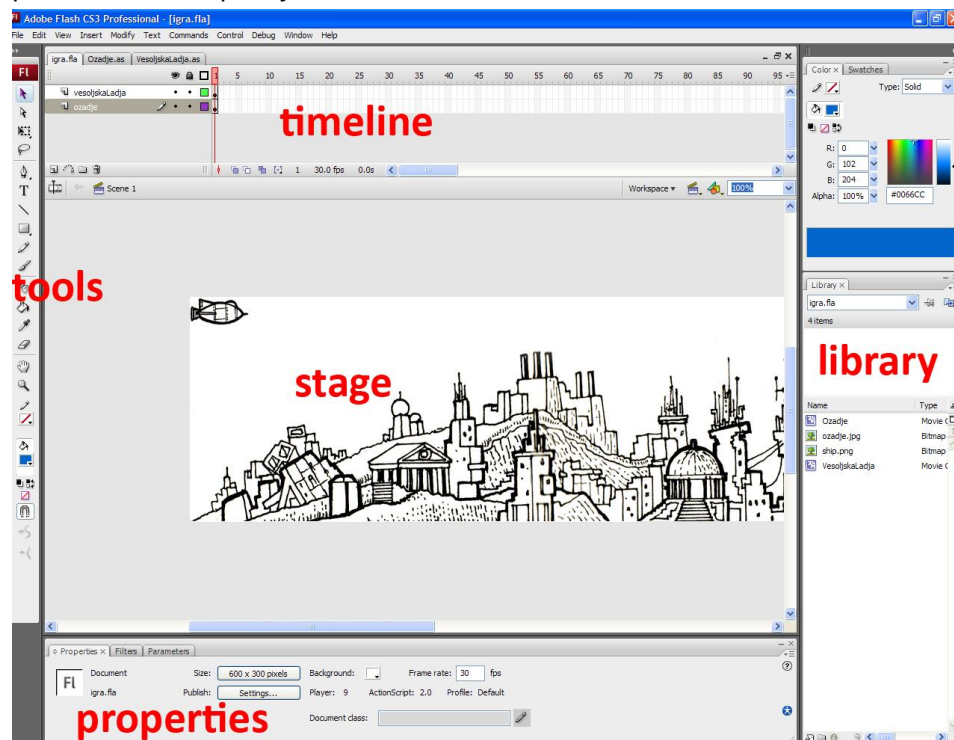


Uvod

Ko zaženemo Flash, se nam prikaže okno podobno spodnjemu:



Gremo na Open..., najdemo .fla datoteko od današnje vaje in jo odpremo. Sedaj Flash aplikacija izgleda podobno kot na spodnji sliki:



Kot lahko vidimo, je Flash GUI razdeljen na nekaj delov:

- **Stage** je osnovni prostor za ustvarjanje naše Flash igre. V ta del bomo uvažali grafične elemente, ki se bodo pojavljali v končni verziji naše igre. Velikost stage-a nam pove, koliko prostora na zaslonu bo zasedla naša igrice, ko jo objavimo (publish). To velikost lahko spremenimo v panelu properties.
- V panelu **properties** lahko spremenimo tudi barvo **stage-a**, slikovno hitrost (frame rate)naše igre, kasneje pa bomo lahko tu spreminjali in pregledovali lastnosti vsakega izbranega elementa.
- Panela **timeline** ne bomo veliko uporabljali, ta pride bolj v poštev pri animacijah. Mi bomo delali vglavnem v prvem framu.
- **Library** shranjuje vse grafične in zvočne elemente, ki jih uvozimo v datoteko igre.
- Orodja (**tools**) se nahajajo v ozkem orodnem traku, podobno kot pri Photoshopu. Gre za orodja, ki se uporabljajo pri risanju. Glede na to, da bomo naredili igrice iz že pripravljenih grafičnih elementov, teh orodij ne bomo veliko uporabljali.

Razred Ozadje

V Flashu odpremo še .as datoteko (Ozadje.as), ki vsebuje kodo za ozadje. Oglejmo si datoteko Ozadje.as:

```
1class Ozadje extends MovieClip
{
    var hitrostOzadja;

    2function onLoad()
    {
        hitrostOzadja = 5;
    }

    3function onEnterFrame()
    {
        4_x = _x - hitrostOzadja;
    }
}
```

1. ActionScript je objektno orientiran jezik in v tej igri bo vsak element član nekega razreda. Razredi vsebujejo spremenljivke, ki shranjujejo vrednosti in funkcije, ki te vrednosti spreminjajo. Razred, kateremu pripada naše ozadje, se imenuje *Ozadje*. *MovieClip* je razred, ki je že vgrajen v Flash in ima določene lastnosti in funkcije. **V tej vrstici smo napisali, da bomo ustvarili nov razred, ki bo podedoval tudi lastnosti razreda *MovieClip*.** V nadaljevanju smo definirali tudi spremenljivko *hitrostOzadja*. Ta spremenljivka bo hranila podatek o hitrosti premikanja ozadja.
2. Funkcija `onLoad()` je že vgrajena v razred *MovieClip*. Ta funkcija se izvede samo enkrat in sicer, ko *MovieClip* prvič vstopi na sceno (stage). Zato v to funkcijo pišemo kodo, ki se mora izvesti samo enkrat na začetku. **Mi smo določili hitrost premikanja ozadja: `hitrost = 5;`**

3. Funkcija `onEnterFrame()` je še ena funkcija, ki že obstaja v razredu `MovieClip`. Ta funkcija se izvede ob vsakem novem frameu, kar v našem primeru pomeni 30-krat v sekundi. To je zelo uporabna funkcija, saj lahko v njej sproti preverjamo kaj se dogaja z ozadjem in z njim upravljamo.
4. V funkciji `onEnterFrame()` bomo premikali ozadje in sicer za vrednost spremenljivke *hitrostOzadja*. Razred `MovieClip` ima vgrajene dve spremenljivki, ki v vsakem trenutku vsebujeta informacijo o položaju movie clipa: `_x` in `_y`. Gre za položaj v pixlih (slikovnih elementih). **Da ustvarimo iluzijo premikanja ladje v desno, moramo premikati ozadje v levo, torej moramo ozadju manjšati vrednost spremenljivke `_x`.** Ker to počnemo v funkciji `onEnterFrame()`, se bo to zgodilo 30 krat v sekundi.
5. Slika ozadja je precej široka, a je vseeno omejena in s takšnim premikanjem hitro pridemo do konca. Takrat se nam zgodi, da ozadje izgine in ladja leti po beli podlagi. To lahko rešimo na več načinov, mi smo se odločili, da ozadje zaciklamo v neskončni zanki. Ozadje je široko 2110 pixlov (slikovnih elementov), premikamo ga pa v negativni smeri. **Torej, ko pridemo do konca ozadja, ga postavimo spet na začetek.**

Razred *VesoljskaLadja*

V Flashu odpremo še .as datoteko (*VesoljskaLadja.as*), ki vsebuje kodo za vesoljsko ladjo.

Oglejmo si datoteko *VesoljskaLadja.as*:

```
1 class VesoljskaLadja extends MovieClip
{
    var hitrost;

    2 function onLoad()
    {
        hitrost = 10;
    }

    3 function onEnterFrame()
    {
        4 if( Key.isDown(Key.RIGHT) )
        {
            5 _x = _x + hitrost;
        }
    }
}
```

1. V tej vrstici smo napisali, da bomo ustvarili nov razred, ki bo podedoval tudi lastnosti razreda *MovieClip*. V nadaljevanju smo definirali tudi spremenljivko *hitrost*. Ta spremenljivka bo hranila podatek o hitrosti premikanja ladje.

2. Funkcija `onLoad()` je že vgrajena v razred `MovieClip`. Ta funkcija se izvede samo enkrat in sicer, ko `MovieClip` prvič vstopi na sceno (`stage`). Zato v to funkcijo pišemo kodo, ki se mora izvesti samo enkrat na začetku. **Mi smo določili hitrost ladje: `hitrost = 10`;**
3. Funkcija `onEnterFrame()` je še ena funkcija, ki že obstaja v razredu `MovieClip`. Ta funkcija se izvede ob vsakem novem frameu, kar v našem primeru pomeni 30-krat v sekundi. To je za nas najbolj uporabna funkcija, saj lahko tu sproti preverjamo kaj se dogaja z vesoljsko ladjico in z njo upravljamo.
4. Z vesoljsko ladjo upravljamo s pomočjo smernih tipk na tipkovnici: UP, DOWN, LEFT, RIGHT. To naredimo s pomočjo še enega razreda, ki je vgrajen v Flash: `Key`. Da preverimo, če je tipka pritisnjena, napišemo naslednji kos kode:
`Key.isDown(oznaka tipke)`.
V našem primeru preverjamo, če je bila pritisnjena smerna tipka RIGHT (desno).
5. Če pritisnemo tipko RIGHT, radi bi premaknili ladjo v desno. Razred `MovieClip` ima vgrajeni dve spremenljivki, ki v vsakem trenutku vsebujeta informacijo o položaju movie clipa: `_x` in `_y`. Gre za položaj v pixlih (slikovnih elementih). Da ladjo premikamo v desno, ji moramo večati vrednost spremenljivke `_x`. To naredimo tako, da ji **vsakič, ko uporabnik pritisne tipko RIGHT, povečamo vrednost in sicer za vrednost spremenljivke `hitrost`**. Večja kot je vrednost spremenljivke `hitrost`, hitreje se bo ladja premikala.