

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

Završni projekt

Tangram

Tomislav Horina, Edi Ibriks, Edita Kulović, Petra Penzer, Barbara Prkačin

Mentor: *Goran Igaly*

Zagreb, veljača, 2016.

Sadržaj:

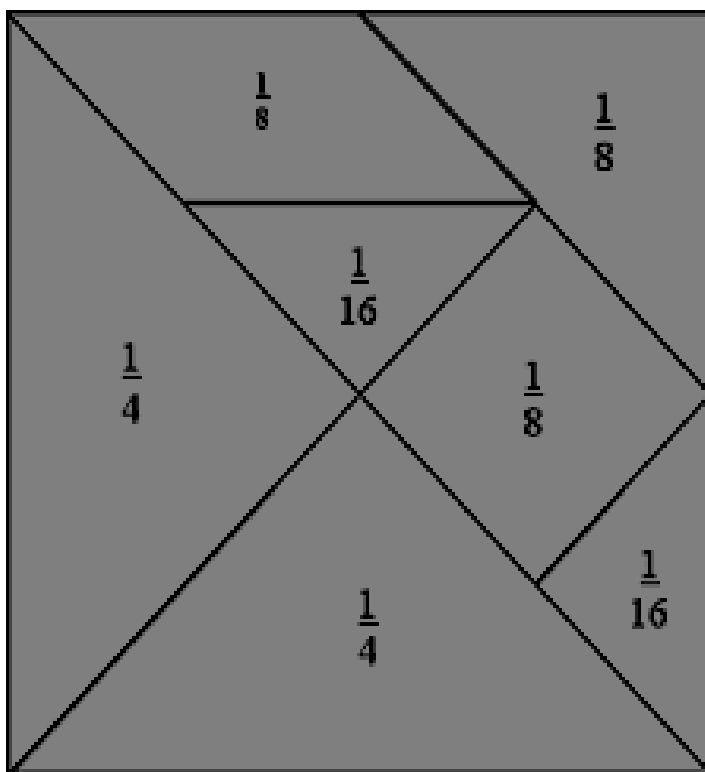
1	Tangram	3
2	Opis igre.....	4
3	Implementacija	6
4	Literatura.....	10

1 Tangram

Tangram je slagalica (zagonetka) porijeklom iz Kine koja je u Europu i Ameriku stigla početkom 19. stoljeća.

Originalni tangram je kvadrat (Slika 1) podijeljen na sedam dijelova tako da se dobiju:

- dva velika jednakokračna pravokutna trokuta (hipotenuza 1, katete $\sqrt{2}/2$, površina $1/4$)
- jedan jednakokračni pravkutni trokut srednje veličine (hipotenuza $\sqrt{2}/2$, katete $1/2$, površina $1/8$)
- dva mala jednakokračna trokuta (hipotenuza $1/2$, katete $\sqrt{2}/4$, katete $1/2$, površina $1/16$)
- jedan kvadrat (stranica $\sqrt{2}/4$, površina $1/8$)
- jedan paralelogram (stranice $1/2$ i $\sqrt{2}/4$, površina $1/8$)



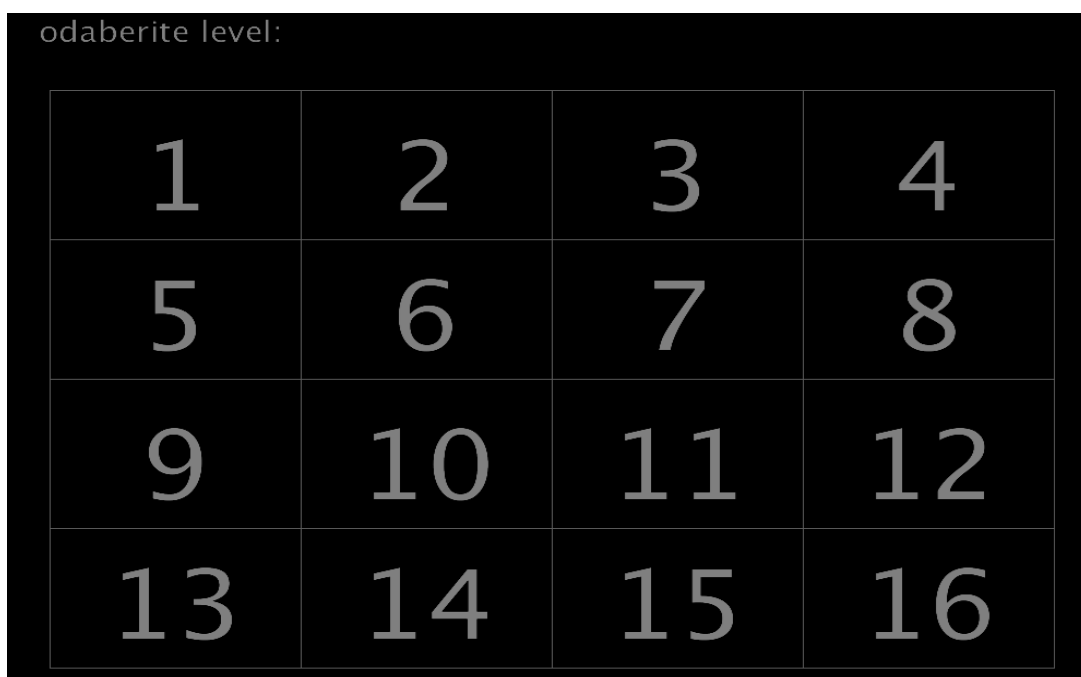
Slika 1: Originalni tangram je kvadrat podijeljen na sedam dijelova

Igra se sastoji u tome da se od dijelova sastavljaju, unaprijed zadani, različiti likovi i figure. Osnovna pravila, koja se moraju poštivati, su:

- Uvijek se mora upotrijebiti svih sedam dijelova
- Dijelovi se postavljaju jedan do drugog, ne smiju preklapati
- Dijelovi se po potrebi mogu preokrenuti na drugu stranu

2 Opis igre

Igra je koncipirana na način da je igraču ponuđen izbor između više levela koje može igrati (Slika 2). Kad igrač izabere određeni level iscrtava se bijela pozadina koja predstavlja zadani oblik i sedam dijelova u različitim bojama (Slika 3) kojima je potrebno ispuniti oblik prema pravilima igre. Kada se popuni zadani oblik level je uspješno odrađen. Dijelovi koje igrač koristi mogu se rotirati po potrebi koristeći desni klik miša. Također, igraču se mjeri vrijeme potrebno da bi uspješno odradio izabrani level. Na početni meni moguće se vratiti u bilo kojem trenutku pomoću tipke m.



Slika 2: Početak igre: Izbor levela



Slika 3: Bijela pozadina koju treba popuniti i sedam dijelova u različitim bojama

Popis levela:

1. Kvadrat
2. Trokut
3. Pješčani sat
4. Pravokutnik
5. Četverokut
6. Šesterokut
7. Paralelogram
8. Slovo D
9. Slovo A
10. Slovo M
11. Slovo C
12. Slovo T
13. Patka
14. Ribica 1
15. Pas
16. Ribica 2

3 Implementacija

Za izradu projekta koristili smo Processing zbog jednostavnosti i praktičnosti.

Na početku se iscrtava meni koji nudi izbor između 16 levela (Slika 2). Za iscrtavanje menija se koriste jednostavne naredbe *line()* i *text()*.

```
void loadMenu(){
    fill(255);
    textSize(30);
    textAlign(LEFT);
    text("odaberite level: ", 30, 30);
    stroke(220);
    line(40, 80, 960, 80);
    line(40, 235, 960, 235);
    line(40, 380, 960, 380);
    line(40, 535, 960, 535);
    line(40, 680, 960, 680);
    line(40, 80, 40, 680);
    line(960, 80, 960, 680);
    line(270, 80, 270, 680);
    line(500, 80, 500, 680);
    line(730, 80, 730, 680);
    textAlign(CENTER);
    textSize(100);
    text("1", 155, 205);
    text("5", 155, 350);
    text("9", 155, 505);
    text("13", 155, 650);
    text("2", 385, 205);
    text("6", 385, 350);
    text("10", 385, 505);
    text("14", 385, 650);
    text("3", 615, 205);
    text("7", 615, 350);
    text("11", 615, 505);
    text("15", 615, 650);
    text("4", 845, 205);
    text("8", 845, 350);
    text("12", 845, 505);
    text("16", 845, 650);
}
```

Za odabir levela se brine funkcija *levelFinder()* koja provjerava poziciju miša i postavlja broj levela na odabrani ovisno o poziciji.

```
void levelFinder(){
    if (mouseX > 40 && mouseX < 270){
        if (mouseY > 80 && mouseY < 235){ state=1; level=1; time2=millis();}
        if (mouseY > 235 && mouseY < 380){ state=1; level=5; time2=millis();}
        if (mouseY > 380 && mouseY < 535){ state=1; level=9; time2=millis();}
        if (mouseY > 535 && mouseY < 680){ state=1; level=13; time2=millis();}
    }
    else if (mouseX > 270 && mouseX < 500){
        if (mouseY > 80 && mouseY < 235){ state=1; level=2; time2=millis();}
    }
}
```

```

    if (mouseY >235 && mouseY < 380){state=1;level=6;time2=millis();}
    if (mouseY >380 && mouseY < 535){state=1;level=10;time2=millis();}
    if (mouseY >535 && mouseY < 680){state=1;level=14;time2=millis();}
  }
  else if (mouseX > 500 && mouseX <730){
    if (mouseY >80 && mouseY < 235){state=1;level=3;time2=millis();}
    if (mouseY >235 && mouseY < 380){state=1;level=7;time2=millis();}
    if (mouseY >380 && mouseY < 535){state=1;level=11;time2=millis();}
    if (mouseY >535 && mouseY < 680){state=1;level=15;time2=millis();}
  }
  else if (mouseX > 730 && mouseX <960){
    if (mouseY >80 && mouseY < 235){state=1;level=4;time2=millis();}
    if (mouseY >235 && mouseY < 380){state=1;level=8;time2=millis();}
    if (mouseY >380 && mouseY < 535){state=1;level=12;time2=millis();}
    if (mouseY >535 && mouseY < 680){state=1;level=16;time2=millis();}
  }
}

```

Kad se odabere level iscrtavaju se zadani oblici i bijeli lik koji treba popuniti. Oblike možemo rotirati desnim klikom i razmještati ih dok ne dobijemo ciljani lik. Kada se prekrije površina lika na način da je preostalo manje od 5000 bijelih piksela level je završen. Koristimo histogram koji računa količinu piksela svih boja na ekranu. Funkcija *levelFinished()* provjerava količinu bijelog pixela.

```

boolean levelFinished(){
  int[] hist = new int[256];
  // Calculate the histogram
  for (int i = 0; i < 1080; i++) {
    for (int j = 0; j < 720; j++) {
      int bright = int(brightness(get(i, j)));
      hist[bright]++;
    }
  }
  if (hist[255]<5000)
    return true;
  else return false;
}

```

Dok igrač igra u desnom donjem kutu se prikazuje tekst koji navodi trenutni level te se mjeri vrijeme koje je potrebno igraču da dovrši igru. Po uspješnom završetku levela ispisuje se poruka i trajanje igre (Slika 4). Ako igrač ne može dovršiti odabrani level postoji opcija povratka na početni meni pritiskom tipke **m**.

Prošli ste level 15

Potrebno vrijeme: 50.385

Slika 4: Završetak levela

Sljedeći dio koda prikazuje kako se prikazuju pozadine levela. U slučaju 0 prikazuje se zadnji završeni level , a u ostali 1, 2, ... , 16 prikazuju se pozadine oblika.

```
void loadLevel(int l){
    switch(l){
        case 0: {
            background(0);
            fill(255);
            textAlign(CENTER);
            textSize(100);
            stroke(15);
            text("Prošli ste level " + str(level-1) ,width/2, height/2);
            textSize(50);
            text("Potrebno vrijeme: " + time3, width/2,height/2+100);
            break;
        }
        case 1: { // kvadrat
            fill(255);
            beginShape();
            vertex(10,330);
            vertex(330,650);
            vertex(650,330);
            vertex(330,10);
            endShape();
            break;
        }
        case 2: {
            ...
        }
        ...
    }
}
```


Mjerenje vremena smo napravili ovako:

```
void timer(){
    fill(255);
    textSize(30);
    textAlign(LEFT);
    text("Level: " + level + " Vrijeme: " + ((millis()-time2)/1000) , 500 , 690);
}
```

Funkcija millis() vraća koliko je prošlo od pokretanja programa. U time2 je spremljeno vrijeme kada smo odabrali level.

4 Literatura

- [1] <https://en.wikipedia.org/wiki/Tangram>
- [2] <http://www.tangram-channel.com/tangram-solutions/>
- [3] <http://openprocessing.org/sketch/20158>
- [4] <https://processing.org/>