
Igra 5

IGRA CITY BUILDER - osnovni deo

- Igrač igra protiv računara
- Igra se sastoji u biranju građevine za građenje
- Svakog dana dobijaju se random zlatnici od 1 do 5
- Početna vrednost zlatnika je 50
- Tokom gradnje, za svaku građevinu se dobija sreća, a svaka košta određeni broj dana i zlatnika
- Pobednik je ko ima više sreće nakon 50 dana

DODATNE FUNKCIONALNOSTI

- Dolaze lopovi random dana i pljackaju grad, uzimaju odredjen broj zlatnika
- Od izgrađene građevine se dnevno dobija novac

DISKUSIJA

- Potrebno je da postoje liste u kojima čuvamo informacije o tome koliko koja zgrada čega donosi
- Potrebno je da se za računar random biraju zgrade i da se njegova gradnja simulira a mi našu igramo
- Potrebno je ispisivati izveštaj svakog dana šta se dešava i status zlata i sreće za igraca
- Sve informacije o gradjevinama čuvati u zasebnim listama, ali po indeksu u listi čuvati info koja informacija se odnosi na koju građevinu
- Pri izboru gradnje, treba pratiti da li ima dovoljno zlatnika i da li ima dovoljno dana za gradnju
- Svaki dan u kome se ne uspije zapoceti radnja zbog nekog od ta dva slucaja, tretira se kao izgubljeni dan

IMPLEMENTACIJA V1 [BEZ DANA]

```
import random

def staGradis():
    print("Izaberite sta gradite:")
    print("1 - ulica")
    print("2 - skola")
    print("3 - zgrada")
    print("4 - bolnica")
    print("5 - park")
    objekat = int(input())
```

```

    return objekat-1
def ispisPobednika():
    print("*****")
    print("Korisnik - sreca", srecaIgraca)
    print("Racunar - sreca", srecaRacunara)
    if srecaRacunara < srecaIgraca:
        print("Pobednik je korisnik.")
    elif srecaRacunara == srecaIgraca:
        print("Nemamo pobednika.")
    else:
        print("Pobednik je racunar.")

# promenljive
objekti = ["ulica", "skola", "zgrada", "bolnica", "park"]
kolikoKosta = [8, 13, 17, 19, 5]
sreca = [4, 6, 8, 5, 5]

srecaIgraca = 0
srecaRacunara = 0
zlatniciIgraca = 50
zlatniciRacunar = 50

# simulacijaRacunar
while True:
    gradnja = random.randint(0, 4)
    if zlatniciRacunar >= kolikoKosta[gradnja]:
        zlatniciRacunar -= kolikoKosta[gradnja]
        srecaRacunara += sreca[gradnja]
    else:
        break

# simulacija igraca
while True:
    gradnja = staGradis() # indeks
    if zlatniciIgraca >= kolikoKosta[gradnja]:
        zlatniciIgraca -= kolikoKosta[gradnja]
        srecaIgraca += sreca[gradnja]
    else:
        break
ispisPobednika()

```

OBJASNJENJA:

- Objasniti sta jos moze da se izdvoji u funkciju
- Ako se doda i vreme, koliko dugo traje svaka gradnja, igra se komplikuje

- Kod učenika koji su sporiji, dovoljno je da oni uradi samo ovu V1 verziju, da imaju to da radi, a onda zasebno da adaptiraju i da dodaju da postoje i vremena gradnje i lopovi.

IMPLEMENTACIJA V2 [sa DANIMA]

```
# korak 0 : tok i struktura rešenja
```

```
# neophodne funkcije za:  
#     za unos gradnje,  
#     ispis pobednika,  
#     simulaciju za racunar,  
#     simulaciju za igraca  
# neophodne promenljive za igracu  
#     liste  
# poziv fja:  
#     simulacija racunar  
#     simulacija igraca  
#     ispis pobjednika
```

```
# korak 1 : promenljive koje nam trebaju
```

```
# promenljive
```

```
objekti = ["ulica", "skola", "zgrada", "bolnica", "park"]  
kolikoKosta = [8, 13, 17, 19, 5]  
kolikoTraje = [3, 7, 8, 9, 5]  
sreca = [4, 6, 8, 5, 5]
```

```
ukupnoDana = 50  
srecaIgraca = 0  
srecaRacunara = 0  
zlatniciIgraca = 50  
zlatniciRacunar = 50
```

```
# korak 2: funkcija za gradnju, dodati na vrh fajla
```

```
# fja za unos gradnje
```

```
# ne mora da se doda provera sa whileom u proveru inputa u prvoj verziji, moze kasnije
```

```
def gradiNoviObjekat():  
    print("Izaberite sta gradite:")  
    print("1 - ulica")  
    print("2 - skola")  
    print("3 - zgrada")  
    print("4 - bolnica")  
    print("5 - park")  
    objekat = input()  
    while not objekat.isdigit() or not (0 < int(objekat) < 6):
```

```
    print("Neispravan unos. Unesi broj od 1 do 5")
    objekat = input()
return int(objekat)
```

```
# korak 3: pomocna funkcija za ispis pobednika
```

```
def ispisPobednika():
    print("*****")
    print("Korisnik - sreca", srecaIgraca)
    print("Racunar - sreca", srecaRacunara)
    if srecaRacunara < srecaIgraca:
        print("Pobednik je korisnik.")
    elif srecaRacunara == srecaIgraca:
        print("Nemamo pobednika.")
    else:
        print("Pobednik je racunar.")
```

```
# korak 4: glavni tok programa nakon promenljivih
```

```
# podebljano je šta se dodaje
```

```
...
```

```
zlatniciIgraca = 50
zlatniciRacunar = 50
```

```
srecaRacunara = simulacijaRacunar(zlatniciRacunar)
```

```
srecaIgraca = simulacijaIgrac(zlatniciIgraca)
```

```
ispisPobednika()
```

```
...
```

```
# simulacija za racunar
```

```
# definišemo tu funkciju na početku fajla u delu gde su funkcije
```

```
# možemo samo kostur funkcije da ispišemo a samu implementaciju sledeći  
čas
```

```
def simulacijaRacunar(zlatniciRacunar):
    daLiSeGradi = False # promenljiva preko koje cemo da pratimo kada
    pravimo novi objekat a kada se gradi zapoceti
    daniGradnje = 0
    srecaRacunara = 0 # ono sto ce funkcija da vrati

    for dan in range(1, 51, 1):
        # dodajemo random zlatnike

        if daLiSeGradi == False:
            # biramo objekat za gradnju
            # proverava se da li ima dovoljno zlatnika
            # proverava se da li ima dovoljno dana
            # zapocinje se gradnja
```

```

else:
    # gradnja je u toku
    # smanjuju se dani gradnje
    # za završenu gradnju, dodajemo sreću

return srećaRacunara

# simulacija za igrača
# definišemo tu funkciju na početku fajla u delu gde su funkcije
# možemo samo kostur funkcije da ispišemo a samu implementaciju sledeći
čas

def simulacijaIgrac(zlatniciIgraca):
    daLiSeGradi = False
    daniGradnje = 0
    srećaIgraca = 0
    for dan in range(1, ukupnoDana + 1, 1):
        # ispis za svaki dan koji je dan

        # korisnik dobije random zlatnike

        # da li se nešto gradi
        if daLiSeGradi == False:
            # treba da započnemo gradnju
            # treba da ispišemo šta se gradi, koliko traje
            # treba da proverimo da li imamo dovoljno dana
            # treba da proverimo da li imamo dovoljno zlatnika

        else:
            # gradnja je u toku plus
            # treba da se smanji gradnja za 1 dan
            # treba da se proverí ako je gradnja završena
            # ako jeste, da se doda sreća

        # svakog dana treba da se ispiše status dana koliko ima sreće i
        zlata trenutno
        # između svakog dana treba da se simulira pauza

    return srećaIgraca

# korak 5 : simulacija računara

```

```

def simulacijaRacunar(zlatniciRacunar):
    daLiSeGradi = False # promenljiva preko koje cemo da pratimo kada
    pravimo novi objekat a kada se gradi zapoceti
    daniGradnje = 0
    srecaRacunara = 0 # ono sto ce funkcija da vrati

    for dan in range(1, 51, 1):
        # dodajemo random zlatnike
        zlatniciRacunar += random.randint(1, 5)

        if daLiSeGradi == False:
            # biramo objekat za gradnju
            indeksObjekta = random.randint(0, 4)
            # proverava se da li ima dovoljno zlatnika
            # proverava se da li ima dovoljno dana
            # i zapocinje se gradnja
            if zlatniciRacunar > kolikoKosta[indeksObjekta] and
            kolikoTraje[indeksObjekta] < ukupnoDana-dan:
                zlatniciRacunar -= kolikoKosta[indeksObjekta]
                daLiSeGradi = True
                daniGradnje = kolikoTraje[indeksObjekta]
                print("Gradi se", objekti[indeksObjekta])
            else:
                print("Nemate dovoljno zlatnika ili dana gradnje.")
                print("Izgubili ste jedan dan gradnje.")
        else:
            # gradnja je u toku
            # smanjuju se dani gradnje
            # za završenu gradnju, dodajemo sreću
            daniGradnje = daniGradnje - 1
            if daniGradnje == 0:
                # završena gradnja, dodajemo sreću
                daLiSeGradi = False
                srecaRacunara += sreca[indeksObjekta]

    return srecaRacunara

```

```
# korak 6 : simulacija igrača
```

```

def simulacijaIgrac(zlatniciIgraca):
    daLiSeGradi = False
    daniGradnje = 0
    srecaIgraca = 0
    for dan in range(1, ukupnoDana + 1, 1):
        # ispis za svaki dan koji je dan
        print("***** DAN {}*****".format(dan))
        # korisnik dobije random zlatnike

```

```

zlatniciIgraca += random.randint(1, 5)
# da li se nesto gradi
if daLiSeGradi == False:
    # treba da zapocnemo gradnju
    indeksObjekta = gradiNoviObjekat() - 1
    # treba da ispisemo sta se gradi, koliko traje
    opis = "{} traje {} dana, kosta {} novcica i donosi {}
srece.".format(objekti[indeksObjekta], kolikoTraje[indeksObjekta],
kolikoKosta[indeksObjekta], sreca[indeksObjekta])
    print(opis)
    # treba da proverimo da li imamo dovoljno dana
    # treba da proverimo da li imamo dovoljno zlatnika
    if kolikoKosta[indeksObjekta] <= zlatniciIgraca and
kolikoTraje[indeksObjekta] < ukupnoDana-dan:
        zlatniciIgraca -= kolikoKosta[indeksObjekta]
        daLiSeGradi = True
        daniGradnje = kolikoTraje[indeksObjekta]
    else:
        # ili nema para ili nema dana
        if kolikoKosta[indeksObjekta] > zlatniciIgraca:
            print("Nemate dovoljno zlatnika.")
            print("Izgubili ste jedan dan gradnje.")
        else:
            print("Nemate dovoljno dana.")
            print("Izgubili ste jedan dan gradnje.")
else:
    # gradnja je u toku plus
    # treba da se smanji gradnja za 1 dan
    daniGradnje = daniGradnje - 1
    print("Gradi se:", objekti[indeksObjekta])
    print("{} / {}".format(kolikoTraje[indeksObjekta] -
daniGradnje, kolikoTraje[indeksObjekta]))
    # treba da se proveri ako je gradnja završena
    # ako jeste, da se doda sreca
    if daniGradnje == 0:
        # završena gradnja
        daLiSeGradi = False
        srecaIgraca += sreca[indeksObjekta]

# svakog dana treba da se ispiše status dana koliko ima sreće i
zlata trenutno
# između svakog dana treba da se simulira pauza
print("Zlatnici:", zlatniciIgraca)
print("Sreća:", srecaIgraca)
time.sleep(1)

```

```
return srecaIgraca
```

```
# korak 7 : igrati igricu
```

```
# korak 8 : dodati neku od dodatnih funkcionalnosti - dodatne gradjevine
```

```
# proširiti nizove sa još tri elementa, mogu učenici da smisle
```

```
# paziti na vrednosti u nizovima da se napravi da je nešto skupo a donosi puno sreće, a nešto jeftino da donosi manje sreće. Takodje, ako traje trace da je skuplje a ako traje više da je jeftinije.
```

```
# korak 9 : upad lopova na kraju dana, dodati random kojim se bira broj npr o 1 do 10 i ako se izabere paran ima lopova a ako ne, nema lopova Lopov odnosi 10 zlatnika npr, a ako nema onda se nosi onoliko koliko ima
```

```
# korak 10: dani gradnje da se ne gube ako nema novca ili dovoljno dana. Da se umesto fora radi while, pa da se broje iteracije samo one kod kojih se ili gradi ili se zapocela uspjesna gradnja. U while petlji mora onda postojati i provera ako je ostalo manje dana nego sto je minimalno trajanje gradnje neke gradjevine.
```