
Igra 7

IGRA:

- Pogađanje tajne reči - klasična igra vešalica
- Program uzima random reč iz fajla koju predstavlja igraču u vidu crtica i broji njegove promašaje crtajući čikicu
- Kada je čikica skroz iscrtan, kraj je igre i igrač je izgubio, a ako je reč pogođena igrač je pobedio.

DISKUSIJA:

- Čitanje iz fajla svih reči i biranje random reči, čuvanje u listi, pa biranje random indeksa
- Funkcija za svaki poseban izgled čikice, ne za deo tela, nego za celu sliku sa posebnim dodatim delom tela
- Funkcija za ažuriranje reči koja se pogađa ukoliko smo neko slovo pogodili
- Crtice za svako slovo - indeksi će se poremetiti jer ćemo imati i po jedan spejs između slova
- Moramo pamtititi slova koja se promaše, ako igrač ponovo ukuca slovo koje je već pogodio ili promašio, moramo da to preskočimo, da se da ponovo unos
- Glasovi koji u našem pismu podrazumevaju dva znaka u latinici, moraju biti izostavljena pri biranju reči i popunjavanju fajla sa mogućim rečima
- Reči koji imaju spejs u sebi mogu da budu dozvoljena ali da se doda npr spejs koji će da se odmah unese (da se veštački pogodi kao taj karakter spejs)
- Glavni program igre - igramo dok ne okinemo kraj igre
 - Detekcija pobede - ako u reci koja se pogađa vise nema _
 - Detekcija gubitka - ako imamo trenutno promašaja koliko je max dozvoljeno, tj kada se iscrta skroz čikica

TEST PRIMER:

* * H A N G M A N * *

Rec koja se pogađa je: _ _ _ _ _

Pogađana slova:

Unesi novo slovo: a

Pogodio si!

Trenutni hangman:

=====

```

||           |
||
||
||
||
||

```

```

||
||
||
||
||
||=====
Rec koja se pogađa je: _ a _ _ _ _
Pogađana slova: a,
Unesi novo slovo: t
Promasio si!
Trenutni hangman:
=====
||          |
||         /  \
||        \ _ /
||
||
||
||
||
||
||
||
||
||=====
Rec koja se pogađa je: _ a _ _ _ _
Pogađana slova: a, t
Unesi novo slovo:

```

IMPLEMENTACIJA:

```

# funkcije za cikicu
def ceoHangman():
    print("=====")
    print("||          | ")
    print("||         /  \ ")
    print("||        \ _ / ")
    print("||       \\ | / ")
    print("||       \\|/ ")
    print("||          | ")
    print("||          | ")
    print("||         /  \ ")
    print("||        _/    \ _ ")
    print("|||")
    print("|||=====")

ceoHangman()

```

NAPOMENA 1: Ne zaboraviti pozvati funkciju na kraju. Prvo raditi kao obicne printove pa posle izdvojiti u funkciju

NAPOMENA 2: Ovaj hangman mora da bude potpun i da fino radi, tj da se lijepo ispisuje. Ovaj dio koda kopiramo 6 puta i pravimo ostale funkcije prateci da izbrisemo po jedan dio tijela.

NAPOMENA 3: Ovaj nivo slozenosti hangmana je optimalan. Da imamo prste ili oci, to bi dodatno uslozilo kod kasnije, ali nije nemoguće.

NAPOMENA 4: \ je specijalan karakter pa se dodatnom \ mora skinuti njena specijalnost

```
# dodavanje ostalih funkcija za iscrtavanje cikice
```

```
def jednaNogaHangman():
    print("=====")
    print("||      | ")
    print("||      / \\ ")
    print("||      \\_/_ ")
    print("||      \\ | / ")
    print("||      \\|/ ")
    print("||      | ")
    print("||      | ")
    print("||      / ")
    print("||     _/ ")
    print("||")
    print("||=====")
```

```
def obeRukeHangman():
    print("=====")
    print("||      | ")
    print("||      / \\ ")
    print("||      \\_/_ ")
    print("||      \\ | / ")
    print("||      \\|/ ")
    print("||      | ")
    print("||      | ")
    print("||      ")
    print("||      ")
    print("||")
    print("||=====")
```

```
def jednaRukaHangman():
    print("=====")
    print("||      | ")
    print("||      / \\ ")
    print("||      \\_/_ ")
    print("||      \\ | ")
    print("||      \\| ")
    print("||      | ")
    print("||      | ")
    print("||      ")
    print("||      ")
    print("||")
    print("||=====")
```

```

def teloHangman():
    print("=====")
    print("||      | ")
    print("||      / \ ")
    print("||     \|_ / ")
    print("||      | ")
    print("||      | ")
    print("||      | ")
    print("||      | ")
    print("||      ")
    print("||      ")
    print("||")
    print("||=====")

```

```

def glavaHangman():
    print("=====")
    print("||      | ")
    print("||      / \ ")
    print("||     \|_ / ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||")
    print("||=====")

```

```

def prazanHangman():
    print("=====")
    print("||      | ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||      ")
    print("||")
    print("||=====")

```

```

prazanHangman()
glavaHangman()
teloHangman()
jednaRukaHangman()
obeRukeHangman()
jednaNogaHangman()
ceoHangman()

```

```
# Razgranavanje i pozivanje tačno jedne funkcije za iscrtavanje u skladu sa brojem promašaja
```

```
def iscrtajTrenutniHangman(brojPromasaja):  
    print("Trenutni hangman: ")  
    if brojPromasaja == 0:  
        prazanHangman()  
    elif brojPromasaja == 1:  
        glavaHangman()  
    elif brojPromasaja == 2:  
        teloHangman()  
    elif brojPromasaja == 3:  
        jednaRukaHangman()  
    elif brojPromasaja == 4:  
        obeRukeHangman()  
    elif brojPromasaja == 5:  
        jednaNogaHangman()  
    elif brojPromasaja == 6:  
        ceoHangman()  
    else:  
        print("Greska!")
```

```
# testiranje te funkcije  
iscrtajTrenutniHangman(5)  
iscrtajTrenutniHangman(-1)  
iscrtajTrenutniHangman(2)
```

DODATNE FUNKCIJE U IGRI:

- Funkcija za ispis naslova i pravila (samo jednom se poziva)
 - Neke od tih pravila će biti implementirana kasnije u igri, pa više neće biti kao pravila (npr postojanje spejsa u reči koja se pogađa)
- Ispis pobede (samo jednom se poziva)
- Ispis kraja (samo jednom se poziva)
- Ispis reči i slova (u svakoj iteraciji)
- Pogađanje slova u reči (u svakoj iteraciji)

IMPLEMENTACIJA:

```
# ispis početka i pravila
```

```
def pocetakNatpis():  
    print("*****")  
    print("**          H A N G M A N          **")  
    print("*****")  
    print("Pravila:")  
    print("1. Slova sa dva karaktera ne bi trebalo da se koriste (nj,  
lj, dz)")
```

```
print("2. Ne sme da postoje dve reci")
print("3. Reciproci su napisane malim slovima")
print("4. Da se ne koriste slova koja u latinici imaju kvakice")
print("5. Pogadja se slovo po slovo")
print("6. Reciproci se zadaju na latinici")
print("*****")
```

```
# ispis pobede, jedna od tri ponudene opcije
```

```
import random
```

```
def ispisPobede():
```

```
    lista_poruka = ["BRAVOOOO! POBEDIO SI!", "POKIDAO SIIII, SVAKA  
ČAST!", "KIDAŠ U OVOJ IGRI! BRAVO!"]
```

```
    print("*****")
```

```
    print(lista_poruka[random.randint(0, len(lista_poruka)-1)])
```

```
    print("*****")
```

```
# ispis gubitka, jedna od tri ponudene opcije
```

```
def ispisGubitka(tajnaRec):
```

```
    lista_poruka = ["VISE SRECE DRUGI PUT :(", "IZGUBIO SI!! REC JE: "+  
tajnaRec + ". NISI MOGAO DA JE NADJES", "ZA MALO! IZGUBIO SIII!"]
```

```
    print("*****")
```

```
    print(lista_poruka[random.randint(0, len(lista_poruka)-1)])
```

```
    print("*****")
```

```
# ispis reči i slova u svakoj iteraciji
```

```
def ispisReciISlova(recKojaSePogadja, nizSlova):
```

```
    print("Rec koja se pogadja je:", recKojaSePogadja)
```

```
    print("Slova koja su pogadjana:", nizSlova)
```

```
# funkcija za pogađanje slova
```

VERZIJA 1: jedno slovo

```
def pogodiSlovo(pogadjanaRec, slovo, tajnaRec):
```

```
    i = tajnaRec.find(slovo)
```

```
    if i != -1:
```

```
        string1 = pogadjanaRec[0:i*2]
```

```
        string2 = pogadjanaRec[i*2+1:]
```

```
        pogadjanaRec = string1 + slovo + string2
```

```
    return pogadjanaRec
```

VERZIJA 2: vise slova u reci

```
def pogodiSlovo(pogadjanaRec, slovo, tajnaRec):
```

```
    for i in range(len(tajnaRec)): # i = 0, 1, 2, 3, 4, 5
```

```
        if slovo == tajnaRec[i]: # a == a, a==n, a==t, a==e, a==n,
```

```
            string1 = pogadjanaRec[0:i*2]
```

```
        string2 = pogadjanaRec[i*2+1:]
        pogadjanaRec = string1 + slovo + string2
    return pogadjanaRec
```

```
# funkcija za pogađanje - testirati
```

```
pogodiSlovo("_ _ _ _ _ ", m, "malina")      # "m _ _ _ _ "
```

```
pogodiSlovo("_ _ _ _ _ ", a, "ananas")     # "a _ a _ a _ "
```

DODATNE FUNKCIJE U IGRI:

- Funkcija za učitavanje reči iz fajla
- Prikaz algoritma zadatka
- Glavni program
 - Promenljive koje nam trebaju
 - Naredbe grananja i slučajevi
 - Objašnjenje jedne iteracije igrice, šta se ponavlja u jednom potezu a šta je van poteza

```
# učitavanje reči iz fajla
```

tajnaRec.txt

```
antena
malina
automobil
lap top
otorinolaringolog
```

python fajl

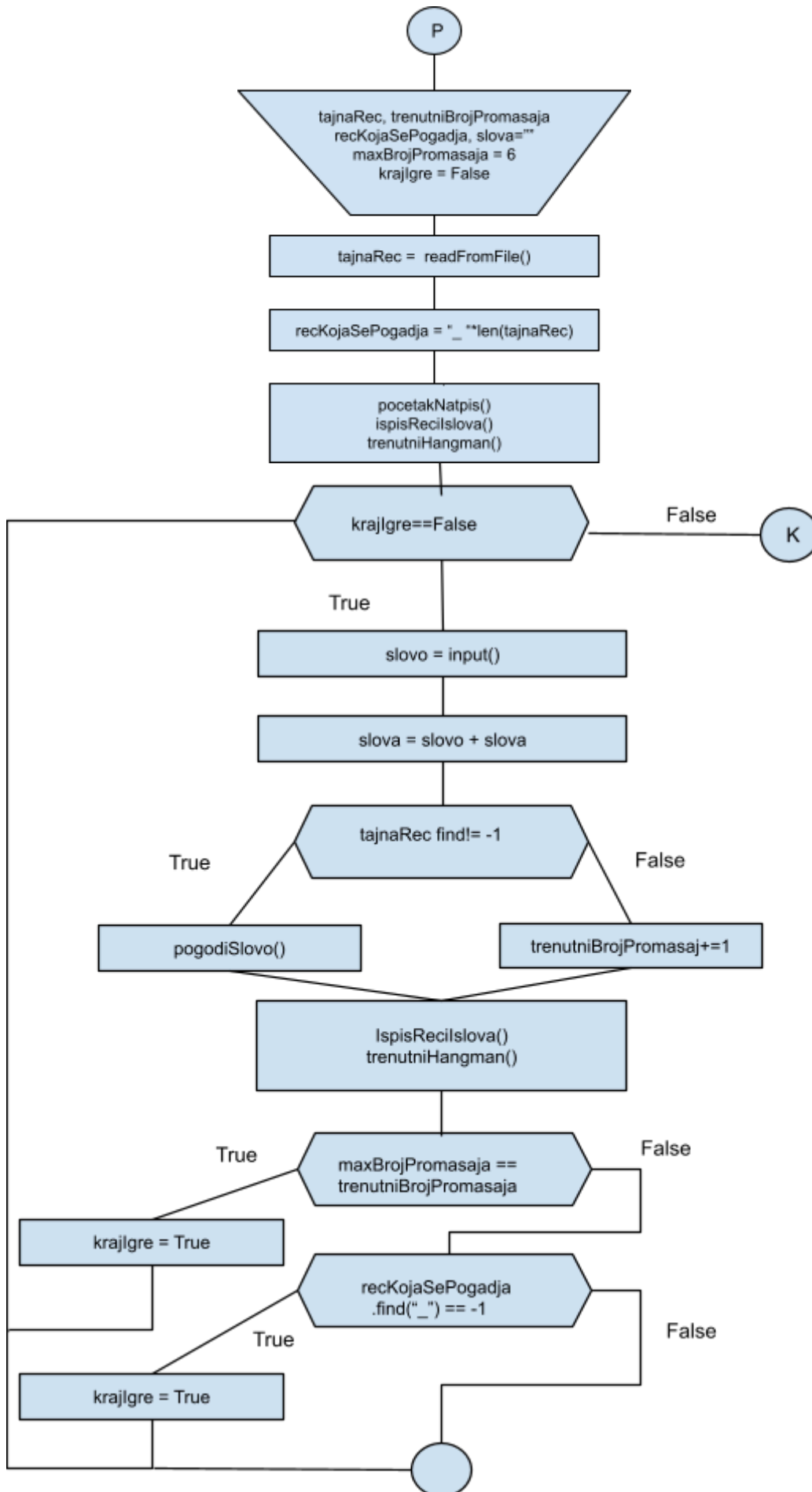
```
def ucitajTajnuRec():
    fajl = open("tajnaRec.txt", "r") # w-write a-append
    reci = fajl.readlines() # to je niz
    fajl.close()

    # izbacivanje poslednjeg slova iz svakog elementa
    novaLista = []
    for rec in reci:
        novaLista.append(rec[:-1])

    # izabrati random rec
    broj = random.randint(0, len(reci)-1)

    # vratiti izabranu rec
    return novaLista[broj]
```

ALGORITAM:



IMPLEMENTACIJA:

```
# promenljive koje nam trebaju
```

```
rec = ucitajTajnuRec()
rec_koja_se_pogadja = len(rec) * "_"

broj_promasaja = 0
max_broj_promasaja = 6
slova = "" # slova koja korisnik unosi
krajIgre = False
```

```
# početak
```

```
pocetakNatpis()
ispisReciISlova(rec_koja_se_pogadja, slova)
iscrtajTrenutniHangman(broj_promasaja)
```

```
# potezi
```

```
while krajIgre == False:

    # unos slova

    # obrada slova - imamo ili nemamo pogodak

    # obrada kraja igrice

    if rec_koja_se_pogadja.find("_") == -1: # nema _ u reci! pobeda!
        krajIgre = True
        ispisPobede()

    if broj_promasaja == max_broj_promasaja:
        krajIgre = True
        ispisGubitka(rec)
```

GLAVNA LOGIKA IGRE:

- Ažuriranje svih promenljivih tokom jedne iteracije
- Popravka funkcija tako da se pravila iz naslova igrice skidaju (nije neophodno sve da se uradi, koliko ima vremena)
- Smisliti napredne funkcionalnosti i implementirati ih (ako ima vremena) ili bar dati ideju

IMPLEMENTACIJA IGRICE:

```
# glavna logika igre
```

```
while krajIgre == False:
```

```

# unos slova
slovo = input("Pogadjaj jedno slovo: ")
# obrada slova - imamo ili nemamo pogodak

slova = slova + slovo
if rec.find(slovo) != -1: # slovce postoji u reci!
    rec_koja_se_pogadja = pogodiSlovo(rec_koja_se_pogadja, slovo,
rec)
else:
    broj_promasaja += 1

ispisReciISlova(rec_koja_se_pogadja, slova)
iscrtajTrenutniHangman(broj_promasaja)

# obrada kraja igrice
if rec_koja_se_pogadja.find("_") == -1: # nema _ u reci! pobeda!
    krajIgre = True
    ispisPobede()

if broj_promasaja == max_broj_promasaja:
    krajIgre = True
    ispisGubitka(rec)

```

TESTIRATI IGRICU

- Šta još može da se doda, šta može da se popravi

POPRAVKE IZ PRAVILA

```

# kukice od latiničnih slova
...
def ucitajTajnuRec():
    f = open("tajnaRec.txt", "r", encoding="utf8")
    rec = f.readlines()

...
# Po defaultu se radi open u ASCII dekodirajućem sistemu, ali na ovaj način
mu možemo reći da se koristi UTF, gdje onda postoje i naša slova sa
kvakicama

# ne dozvoljavamo da se koriste već pogodana slova
...
while krajIgre == False:
    slovo = input("Pogadjaj jedno slovo: ")
    while slova.find(slovo) != -1:
        slovo = input("To slovo je već pogodano. Unesi novo slovo: ")
    slova = slova + slovo

    if rec.find(slovo) != -1: # slovce postoji u reci!
...

```

```

# ažuriranje spacea u učitanoj reči
...
pocetakNatpis()
rec = ucitajTajnuRec()
rec_koja_se_pogadja = len(rec) * "_ "
rec_koja_se_pogadja = pogodiSlovo(rec_koja_se_pogadja, " ", rec)

broj_promasaja = 0
. . .

# dozvaljavanje i malih i velikih slova

...

def prebaciUMalaSlova(rec):
    nova_rec = ""
    for slovo in rec:
        if slovo != " ":
            if slovo in "ČĆŠŽĐ":
                slovo = chr(ord(slovo)+1)
                nova_rec = nova_rec + slovo
            elif slovo in "čćšžđ":
                nova_rec = nova_rec + slovo
            elif daLiSiMaloSlovo(slovo):
                nova_rec = nova_rec + slovo
            else:
                nova_rec = nova_rec+chr(ord(slovo)-ord("A")+ord("a"))
        else:
            nova_rec = nova_rec + slovo
    return nova_rec

def daLiSiMaloSlovo(slovo):
    number = ord(slovo)
    if number >= ord("a") and number <= ord("z"):
        return True
    else:
        return False
. . .

pocetakNatpis()
rec = ucitajTajnuRec()
rec = prebaciUMalaSlova(rec)
rec_koja_se_pogadja = len(rec) * "_ "
. . .

slovo = input("Pogadjaj jedno slovo: ")
while slovo.isalpha() == False:
    slovo=input("Morate uneti slovo, malo il veliko. Probajte ponovo:")

```

```
slovo = prebaciUMalaSlova(slovo)
```

```
while slova.find(slovo) != -1:
```

```
...
```

NAPREDNE VERZIJE:

```
# dozvoljavanje da se pogodi cela rec
```

```
...
```

```
slovo = input("Pogadjaj jedno slovo: ") # "malina" m -> find:0;f->-1
```

```
    if len(slovo) == 1:
```

```
        pogadjanaSlova = pogadjanaSlova + slovo + ", "
```

```
        if tajnaRec.find(slovo) != -1:
```

```
            recKojaSePogadja = pogodiSlovo(tajnaRec, slovo,  
recKojaSePogadja)
```

```
        else:
```

```
            trenutnoPromasaja += 1
```

```
    else:
```

```
        if slovo == tajnaRec:
```

```
            for element in tajnaRec:
```

```
                recKojaSePogadja = pogodiSlovo(tajnaRec, element,  
recKojaSePogadja)
```

```
            else:
```

```
                trenutnoPromasaja+=1
```

```
    #pobeda
```

```
    if recKojaSePogadja.find("_") == -1:
```

```
# racunanje vremena tokom resavanja reci i upis u rekord fajl
```

```
# dodati na pocetak fajla
```

```
from datetime import datetime
```

```
dt = datetime.now()
```

```
ts_pocetak = datetime.timestamp(dt)
```

```
#napraviti funkciju za ispis vremena
```

```
def racunajVreme():
```

```
    dt = datetime.now()
```

```
    ts_kraj = datetime.timestamp(dt)
```

```
    start_ts = datetime.fromtimestamp(ts_pocetak)
```

```
    end_ts = datetime.fromtimestamp(ts_kraj)
```

```
    # Difference between two timestamps in seconds
```

```
    delta = end_ts - start_ts
```

```
    vreme = int(delta.total_seconds())
```

```
    print('Igrao si partiju:', vreme, "sekundi")
```

```
# pozvati funkciju i pri pobedi i pri gubitku
```

```
    if rec_koja_se_pogadja.find("_") == -1: # nema _ u reci! pobeda!
```

```
        krajIgre = True
```

```
        ispisPobede()
```

```
        racunajVreme()
```

```

    if broj_promasaja == max_broj_promasaja:
        krajIgre = True
        ispisGubitka(rec)
        racunajVreme()

# funkcija za upis rekorda u fajl
def upisiVremeUfajl(vreme, ime):
    Fajl = open("rekord.txt", "r")
    Imena = Fajl.readlines()
    Fajl.close()
    indeks = -1
    for i in range(len(Imena)):
        Rec = Imena[i].split(":")
        if vreme <= int(Rec[0]):
            indeks = i
    rec = "{}:{}".format(vreme, ime)
    if indeks == -1:
        Imena.append(rec)
    else:
        Imena.insert(indeks, rec)
    sviRekordi = ""
    for el in Imena:
        sviRekordi += el # vec imaju enter
    Fajl = open("rekord.txt", "w")
    Fajl.write(sviRekordi)
    Fajl.close()

# Pozivanje ove funkcije na pravo mesto u funkciji racunajVreme()

vreme = int(delta.total_seconds())
print('Igrao si partiju:', vreme, "sekundi")
print("Unesi svoje ime za upis rekorda")
Ime = input()
upisiVremeUfajl(vreme, Ime

# dodati ispis najbolja tri vremena negde na pocetak programa

```