Poročilo za 1. del seminarske naloge- igrica Kača

Opis igrice

Kača (Snake) je klasična igrica, pogosto prednaložena na malce starejših mobilnih telefonih. Obstaja precej različic, sam pa sem sestavil meni najbolj znano, z nekaj dodatki.

Cilj igre je dobiti čim več točk, ki jih dobimo s pobiranjem hrane, s čimer se kača tudi daljša. Izogibati se moramo trku v steno, trku v ovire, ki nastajajo v določenih časovnih intervalih (korakih) za kačo in pa seveda trku v rep kače. Takšna je meni znana različica. Sam sem dodal še dve posebni vrsti hrane, prva, ki je rdeče barve, nam omogoča, da za 300 korakov lahko uničujemo ovire (črni kvadratki), druga, ki je modre barve, pa nam za 500 korakov omogoča prehod skozi steno. Trajanje teh sposobnosti se sešteva, če poberemo to vrsto hrane, ko je ta sposobnost še aktivna. Hrana, ki nam prišteva število točk in daljša kačo, je zelene barve.

Klasična Kača ima prednastavljeno velikost mreže, ki je po navadi takšna, da se prilega zaslonu mobilnega telefona, sam pa sem poleg izbire stopnje (vpliva na hitrost kače in število dobljenih točk) dodal še izbiro velikosti mreže in velikosti okna igrice.

Pod igralno površino je še statusna vrstica, ki nam kaže trajanje sposobnosti razbijanja ovir (EC), trajanje sposobnosti prehoda skozi steno in na sredini število točk. Igrica ima tudi menijsko vrstico, ki ima zavihka Datoteka in Pomoč, prvi nam odpre podmeni z zavihki Nova igra, Nastavi površino in Zapri, drugi pa nam odpre okno s pomočjo. Izbira Nastavi površino nam zapre trenutno okno in odpre uvodno okno z nastavitvami.

Ko končamo igro (se zaletimo v steno, oviro ali rep kače), se odpre majhno okno s številom točk in opozorilom za končano igro.

Reševanje

Zaradi bolj enostavnega pisanja in malo krajše kode sem uporabil razred Kaca, v katerem so vključene vse metode, potrebne za delovanje igrice. Poleg tega sem seveda vključil modul tkinter (za izdelavo grafičnega vmesnika) in modul random (za naključno pojavljanje hrane).

Zaradi večje jasnosti rešitve bom opisal vsako metodo posebej.

1. Konstruktor

V konstruktorju se nastavijo nekatere spremenljivke, ki so nujne za delovanje programa, kot na primer stopnja, dimenzija mreže in velikost okna, ki ostanejo nespremenjene, če ob pojavitvi uvodnega okna takoj pritisnemo enter, brez da bi kaj spreminjali. Poleg tega se v konstruktorju seveda ustvari uvodno okno. To vsebuje pozdrav, možnost nastavitve stopnje, ki je vezana na metodi stopnjaPlus() in stopnjaMinus() , velikosti okna, ki je vezana na metodo izberiRes() in dimenzije

mreže, ki je vezana na metodo izberiDimPremik(). Na to okno je vezana tipka enter, ki nam glede na izbrane nastavitve preko metode zacniIgro() odpre glavno okno z igralno površino.

Slika 1: Uvodno okno

2. stopnjaPlus(), stopnjaMinus()

Metodi sta vezana na gumba, ki nam povečujeta ali manjšata težavnost (hitrost) igre. Med gumboma je prikazana trenutno izbrana stopnja. Vrednosti lahko zavzamejo vrednosti od 1 do 6.

3. izberiRes()

Imamo možnost izbire treh različnih velikosti okna, 400 pikslov, 640 pikslov ali 800 pikslov. Prednastavljeno je izbrana velikost 640, ki je tudi rdeče obarvana. Če želimo izbrati drugo, lahko z miško kliknemo na željeno vrednosti, ki se nato obarva rdeče.

4. izberiDimPremik()

Izbiro dimenzije izvedemo tako, da z miško premaknemo drsnik na željeno vrednost. Prednastavljeno je drsnik na sredini (35 kvadratkov za stranico), lahko pa se premika od 10 do 60 kvadratkov.

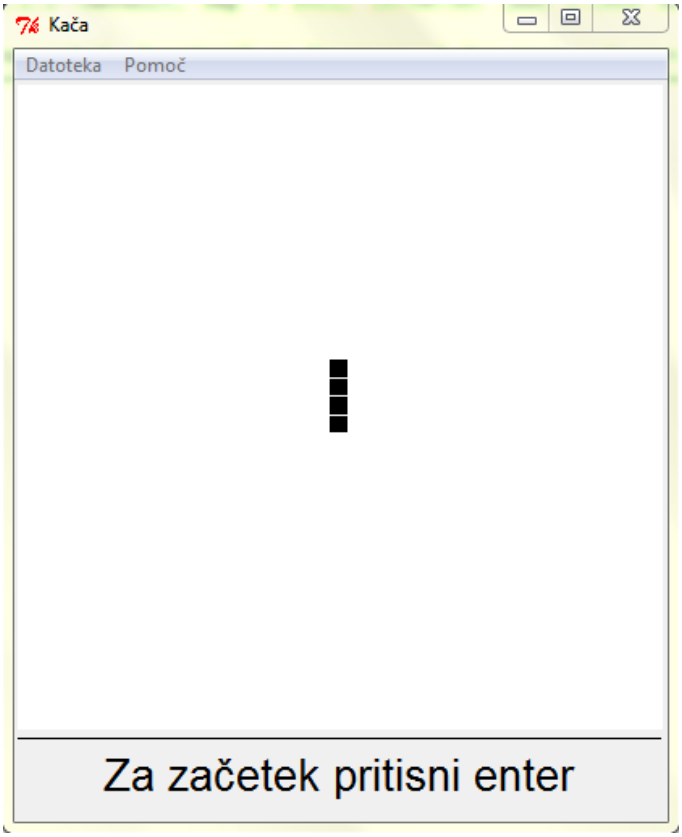
5. zacniIgro()

Če metodo kličemo s tipko enter, ko je aktivno uvodno okno, se ustvarijo seznami indeksev za hrano, kačo, ovire in power-upe. Poleg tega se ustvari tudi seznam grafičnih elementov, ki vsebuje vse

kvadratke v mreži. Kvadratki so v seznamu po vrsti od prvega zgoraj levo do zadnjega spodaj desno, pri čemer so dodani od leve proti desni, kot je prikazano spodaj na primeru:

0	1	2
3	4	5
6	7	8

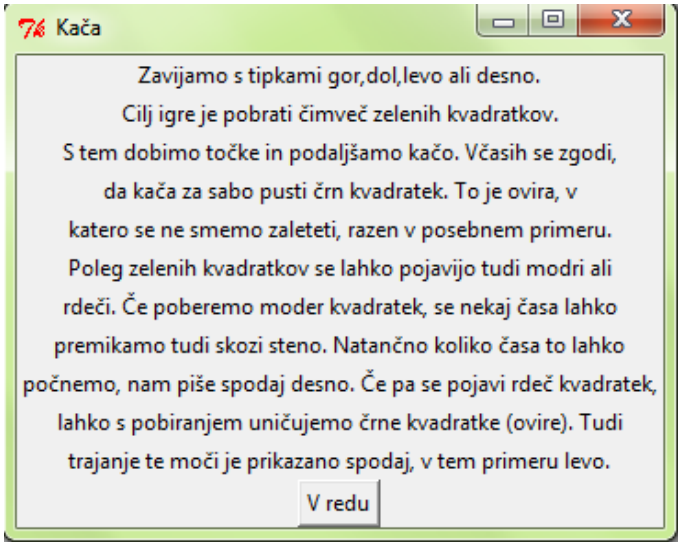
Ko so prazni kvadratki v seznami, se srednji štirje pobarvajo na črno, njihovi indeksi pa se dodajo v seznam kvadratkov kače. Poleg teh seznamov se seveda ustvari igralno okno z menijem, igralno površino in statusno vrstico.



Slika 2: Igralna površina

6. pomoc()

Metoda ustvari okno s tekstom, ki nam razloži nekaj osnovnih pravil igre in pove, s katerimi tipkami igramo.



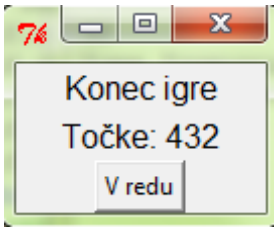
Slika 3: Pomoč

7. start()

Metoda je vezana na tipko enter (ko imamo odprto igralno okno). Kličemo jo lahko samo, če trenutno ne igramo. Ko pritisnemo enter, se kača začne premikati in pojavi se prvi zeleni kvadratek. Naključno se določi tudi število korakov, potrebnih za naslednji power-up. Spremeni se tudi tekst v statusni vrstici tako, da prikazuje, koliko časa nam ostane za oba power-upa in koliko točk smo že dobili.

8. konec()

Ta se izvede, če se zaletimo v steno, oviro ali rep kače. Metoda odpre pojavno okno s točkami in opozorilom, da je igra končana.



Slika 4: Okno ob koncu igre

9. narediHrano()

Kliče se, ko poberemo trenutni zeleni kvadratek. Naključno se izbere indeks kvadratka, nato pa preverimo, če je kvadratek že pobarvan, torej v enem od seznamov. Sicer bi bilo bolj varno uporabljati choice() na seznamu praznih kvadratkov, vendar se mi zdi to boljše, saj sam nisem prišel do tega, da bi bila površina tako zapolnjena, da bi morali indeks mesta generirati več kot dvakrat. Tudi potek igre mi je tako izgledal bolj tekoč, kot pa s seznamom praznih kvadratkov.

10. narediPowerUp()

Metoda je skoraj enaka prejšnji, le da naključno izberemo še eno od dveh vrst. Ko se kliče metoda, se tudi nastavi število korakov, potrebnih za naslednji power-up.

11. tockePlus()

Vsakič, ko poberemo zelen kvadratek, se kliče ta metoda, ki nam poveča točke za dolžino kače, pomnoženo s stopnjo težavnosti.

12. premik()

Metoda je razdeljena na štiri dele, odvisno od smeri, v kateri se premika kača. Za vsak del pa velja, da se najprej izračunata x in y koordinati glave kače po premiku. Nato se preveri, če se bo kača zaletela v steno. Če je trenutno aktiven power-up za prehod skozi steno, se bo kača pojavila na nasprotni strani tabele, sicer je konec igre. Če kača ni ob steni, se izračuna indeks kvadratka, na katerem bo ob premiku. Če je indeks v seznamu ovri,s e prav tako preveri, če je trenutno aktiven power-up za razbijanje ovir. V tem primeru se oviro ignorira, in odstrani iz seznama indeksov, sicer pa je spet konec igre. Če je pred kačo prazen kvadratek ali pa hrana, se kliče metoda premakni(). Čisto na koncu metode premik ta spet kliče sama sebe v časovnem zamiku 800 milisekund deljeno s stopnjo težavnosti.

13. trk()

Ta nastavi stanje igre na 2 (končana) in kliče metodo konec().

14. premakni()

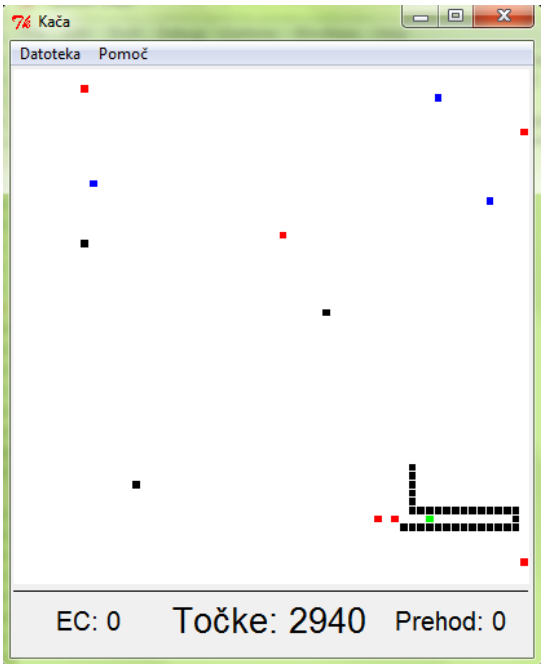
Ta deluje različno glede na to, kaj je pred kačo. Za parameter dobi indeks kvadratka pred njo. Če je to hrana, indeks zadnjega kvadratka kače ostane v seznamu kače in se ne spremeni, kvadratek s hrano pa se doda v ta seznam in prebarva na črno. Če je pred njo posebna vrsta hrane, se doda število korakov primernemu števcu, kvadratek pred kačo se pobarva na črno, njegov indeks se doda v seznam, zadnji kvadratek pa se odstrani in prebarva na belo. Če je število korakov deljivo s 500, se zadnji kvadratek ne prebarva, ampak se njegov indeks doda v seznam ovir.

15. gor(), dol(), levo(), desno()

Metode so vezane na smerne tipke. Glede na to, kam je trenutno obrnjena kača, se smer primerno spremeni glede na to, katero tipko pritisnemo. Če je kača obrnjena na primer levo, pritisnemo pa tipko desno, se ne zgodi nič. Seveda se spremeni samo niz self.smer, kača se v tej smeri premakne šele ob metodi premik().

16. spremeniPovrsino()

Do metode dostopamo preko menija. Uniči nam glavno okno in požene konstruktor, kar pomeni, da se spet pojavi uvodno okno.



Slika 5: Potek igre

Izjava

Izjavljam,da sem seminarsko nalogo opravil samostojno in da sem njen avtor. Zavedam se, da v primeru, če izjava prvega stavka ni resnična, kršim disciplinska pravila.