

## Dokumentacija za Tetris

### UVOD

Projekt Tetris je vrlo poznata retro igra koja je izašla na Sovjetskoj konzoli Elektronika 60. Od tada igra je imala puno varijacija. U ovom projektu cilj je bio pogoditi točno kakav je bio originalni Tetris, što je uspješno izvedeno. Poslije, kada je još bilo vremena za nadograditi projekt, dodao sam 2 nova mode-a koji su većinom bili za zezanciju. U originalnom Tetrisu bodovanje je postignuto "clear"-anjem redova, s time da je 1 red "clear"-an jednako 100 bodova. Ako, pak, igrač uspije "clear"-at 4 reda odjednom postiže 800 umjesto 400 bodova. Ovo je moguće isključivo "I" tetrominom (naziv za oblike od kocaka koji padaju i koje se u Tetrisu kontrolira. Postoje "O", "S", "Z", "T", "I", "L" i "J" te nisu izrađeni od više od 4 kvadrata.). Ako igrač uspije na sljedećem potezu postići jednako rezultat, postiže 1200 umjesto 800 bodova. Ostali mode-ovi su "Chernobyl mode" i "Fast mode", mogu se koristiti zasebno ili u kombinaciji. U Fast mode-u glazba je brža te je brzina ubrzavanja kocaka ubrzana. U normalnome Tetrisu svaki put kada je postignut "clear" ubrzava se pad tetromina. U fastmode-u količina njihovog ubrzavanja poveća se 10 puta (s 1% na 10%). U Chernobyl mode-u izgled tetromina je izobličen te je teško vidjeti kako su okrenuti te koji je to točno tetromino, ako ih niste zapamtili po boji. Ovo otežava igru, ali se lako naviknuti na ovakav izgled kocaka i nakon samo 10-ak rundi lako je prepoznati kako su okrenuti i koji su to točno tetromini. Napravljeno je prilično brzo i lijeno od Leonarda Šimunovića (imam projekt u JavaScriptu sa strane te mi se nije dalo napraviti potpuno ulašten i savršen projekt, ali ga nije bilo suviše teško napraviti i radi točno kako je i očekivano (čak ima i onaj grafički glič gdje se kocke preklapaju ako ste izgubili koji je i u originalnoj verziji)).

### DETALJAN OPIS RADA

Glavni kod nalazi se u "while not kraj:" petlji. Puno sam toga radio in-line jer sam znao da ovo neće biti "polished" projekt od početka. Igra prvo započinje s "while start:" petljom u kojoj se odvija izbornik koji je kontroliran tipkovnicom. Bio sam suviše lijen napraviti pravi GUI engine s tipkama i kontroliranjem mišem te sam otišao na starinski kontroliran tipkovnicom, čak i paše s tadašnjim izbornicima koji su bili slični. U ovom izborniku koristimo eventove od pygame-a kako bismo detektirali pritisak tipaka na tipkovnici. Na ovaj se način izmjenjuju modeovi i pokreće igra. U glavnoj petlji nalazi se "if new:" koji stvara novi nasumični tetromino na početku igre i svaki put kada onaj stari padne. U njemu se također poziva i funkcija gameover() koja provjerava ako je igra gotova (to se događa kada se novostvoreni tetromino preklapa s nekim starijim postavljenim u trenutku stvaranja.) te, ako je kraj igre, svira za kraj te zatvara program. još jedna provjera koja se događa u samom loopu je provjera za pad. Svakih "jumptime" sekundi trenutni tetromino će pasti za 1 blok. Ovo se ubrzava sa svakim clear-om. Također provjerava ako se već postavljeni blok nalazi ispod tetromina pomoću funkcije check(). Ova će funkcija, ako ispod ikojeg od aktivnih tetromina postoji postavljeni tetromin, postaviti trenutni tetromin te pokrenuti "if new:" opet. Zatim idemo na eventove. Eventovi se događaju kada program "osjeti" user-input. Prvo što igrač može učiniti je kliknuti "x" u gornjem desnom kutu prozora. U tom će se slučaju program zaustaviti i glavni prozor zatvoriti. Ako igrač drži "S" na tipkovnici, "jumptime" će se smanjiti i tetromino će ubrzano padati. Ako pak pritisne "A" ili "D" dogoditi će se provjera ako lijevo ili desno od tetromina ima postavljenih blokova, ako ima tetromino se kreće u jednu stranu. Ako se pritisne "Q" ili "E" pokreće se funkcija "checkrot" koja simulira zaokret u neku stranu te provjerava ako se simulirani blokovi nalaze na prostoru koji već zauzimaju drugi blokovi. Ako ima mjesta za zaokret, dogoditi će se

rotacija kod tetromina koji je aktivan pomoću funkcije "move". Svaki put kada se dogodi "if new:" dolazi i do provjere "clear". To je funkcija koja provjerava ako ima redova za clear-at, clear-a ih te dodaje bodove igraču. Na kraju svega ovoga sve ispisuje zadnji blok koda na dnu. Na ekran se crtaju svi oblici te "score" od igrača. U Fast mode-u pad "jumptime" im drastičnije promjene dok u Chernobyl mode-u border aktivnog tetromina raste na 50 i teško je vidjeti gdje se točno nalazi i kako je okrenut tetromino. To je ponanta mode-a, a ne pogreška.

## TEHNIČKE OSOBINE

Zbog drastičnih promjena u programu koje radi pygame, nemoguće je napraviti executable na klasični način, a na poseban način (gdje sam radimo svoj executable uz pomoć nekog drugog library-a) nisam uspio napraviti. Zbog ovog razloga za funkcionalnost programa potrebno je instalirati python i pygame (upute za pygame nalaze se u readmi file-u). Vezano za tehničke potrebe računala potreban je procesor snage Intel-a i3 ili jače, malo manj od 1GB slobodnog prostora, ikoji ram i grafička kartica te Winodws 7 ili novije, Mac OS X 10.11 ili novije te Linux RHEL 6/7, a raditi će i noviji Ubuntu. Za izradu projekta nije se koristilo ništa osim Python-a 3.7.2 i Pygame-a 1.9.6 (koristili su se i library-i time, random i sys, ali oni su uklopljeni u Python 3.7.2).