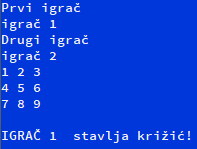
Dokumentacija križić-kružić programa u Pythonu

Kako je Python najpopularniji programski jezik među programerima, može se koristiti za puno stvari koje pomažu ljudima, pa tako i za igrice. Odlučih napraviti kod za poznatu igricu pod nazivom križić-kružić koja se igra pomoću papira i kemijske. Igra traje kratko te se zbog toga koristi puno papira i tinte, a može se iskoristiti samo jednom. Upravo zbog tih mana je video igrica najbolja jer se može igrati beskonačno dugo bez potrošnje papira ili tinte, a jedino što nam treba za to je kompjuter i na njemu kod. Kao učenik srednje škole, nemam puno iskustva u programiranju pa se služim isključivo jednostavnim kodom koji je u skladu s programom koji je zadalo ministarstvo.

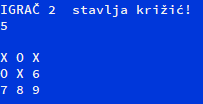
**Opis rada**

Prvo igrači unose svoja imena, a program ih automatski prikazuje velikim tiskanim slovima dalje u programu radi bolje preglednosti i slikovnog poboljšanja:

  
Programom se korisnik služi vizualno jer je polje prikazano u skladu s proporcijama u pravoj igri:



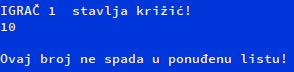
Tijekom igre igrači unose brojeve od 1 do 9 koji predstavljaju jedno od polja na igraćem polju:



Tijekom igre se također broj koji predstavlja polje promijeni u X ili O što daje još bolju preglednost igri:



Kada neki igrač unese broj koji ne postoji, program ga upozorava:



Kada neki igrač unese broj koji je zauzet, program ga upozorava:



Kada neki igrač unese vrijednost koja nije broj, program ga upozorava:



Nakon što su sva polja ispunjena i nema pobjednika, program automatski prikazuje rezultat:



Nakon što program detektira jednu od pobjedničkih kombinacija, automatski prepoznaje koji ju je igrač stvorio te obavještava igrače na kraj igre:

 ili 

Nakon igre program pita žele li igrači početi opet te da odgovore s da ili ne:



**Tehničke informacije**

**Definicija**

Cijeli projekt je unutar jedne definicije kojom se definira igra. Na taj način je lakše napraviti ju u cjelosti te se može lakše ponovno pokrenuti.

Ostali elementi igre su također definirani unutar definicije cijele igre, a oni uključuju polje, odabir brojeva, dva igrača te provjeravanje rješenja.

**Ostali elementi**

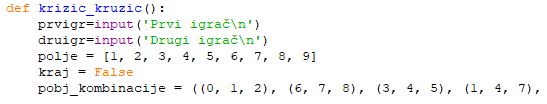
Ostali elementi su unos imena igrača, definiranje vrijednosti polja, pobjedničkih kombinacija i 'kraja', provjeravanje kraja, tj. kad treba završiti igru i kad treba nastaviti te kod za ponovno počimanje programa, tj. igre.

**Konfiguracija sustava**

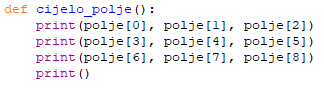
Za pokretanje programa potrebno je instalirati program Python na računalo: 

**Rad programa**

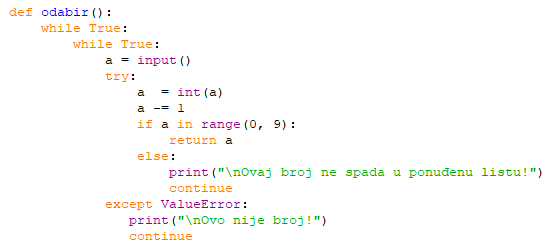
Nakon što je program započet definicijom, dvije vrijednosti koje predstavljaju igrače se unose od strane igrača. Polje je lista od devet brojeva koji su vizualno prikazani u tri reda i razdvojeni razmacima. Kraj je 'false' jer će nam kasnije trebati, a pobjedničke kombinacije su također navedene kao lista:

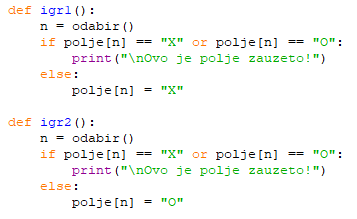
Nakon ovih vrijednosti definira se polje za vizualni prikaz, u tri reda te jedan dodatni za bolji pregled. Prva tri elementa liste su prikazani u prvom redu, druga tri u drugom i treća u trećem:



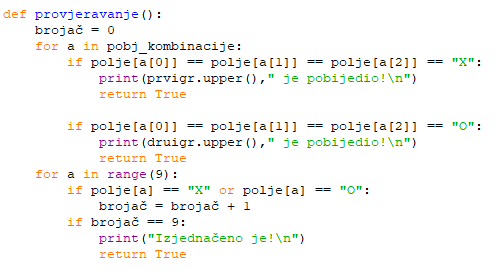
Također je definiran odabir koji provjerava je li unesena vrijednost u skladu s pravilima igre. Počinje s unosom vrijednosti koju onda provjerava tako što ju prvo pretvara u 'integer' te mu oduzima 1 budući da se prvom elementu liste ne dodjeljuje broj 1, već 0. Tada program provjerava je li taj broj u dometu 0 do 9. Ako je onda program ide dalje kao normalno, a ako nije program igrače obavještava da broj ne spada u ponuđenu listu. Ako unesena vrijednost nije broj, program, umjesto da javi pogrešku, nju 'očekuje' te kad ju dobije igrače obavještava da unesena vrijednost nije broj:



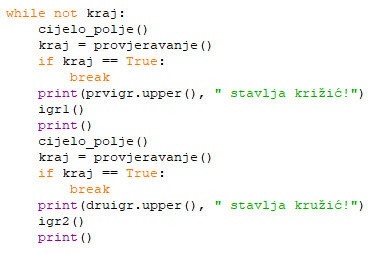
Igrači su definirani kako bi se provjerilo je li neko polje zauzeto, a ako nije mijenja vrijednost polja u vrijednost koja predstavlja određenog igrača:



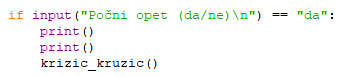
Nakon toga je definirano provjeravanje čiji je zadatak provjeriti skupove vrijednosti svaki put nakon što su unesene. Tu se nalazi i brojač koji nakon devet poteza od kojih niti jedna kombinacija nije pobjednička, javlja da je izjednačeno. Ako je uneseni broj a unutar jedne od pobjedničkih kombinacija, program provjerava jesu li ostala dva polja u toj kombinaciji okupirana istim znakon te ako jesu, ovisi o znaku, javlja da je igrač kojemu taj znak pripada pobijedio. Kada provjeravanje dođe do zaključka koji može biti i ništa od navedenog, provjeravanju se daje neka vrijednost. Ta vrijednost je 'True' ako je jedno od tri stvari točno ili 'False' ako nije:



Slijedeći dio koda odlučuje što će dalje i program ga vrti sve dok 'kraj' nije 'False'. Prikazuje modificirano cijelo polje, te izjednačava 'kraj' s 'provjeravanjem'. 'Kraj'-u je na početku koda dana vrijednost 'False'. Ako je provjeravanje 'True', 'kraj' isto poprima tu vrijednost.



Dok je 'kraj' 'False' program ponavlja ovaj dio koda te je prvo što napravi to da javi da je prvi igrač na redu. Prebaci na definiciju prvog igrača dok on ne završi unos te se tada sve vrijednosti mijenjaju. Program tada opet provjerava vrijednost 'kraj'-a te ako je ista, nastavlja s drugim igračem. Nakon toga se vrti u krug sve dok provjeravanje ne dobije vrijednost 'True'. Tada i 'kraj' dobiva vrijednost 'True' i 'while' petlja prestaje te nastavlja do zadnje 'if' petlje:



Ova 'if' petlja završava igru pitajući igrače hoće li početi ponovno. Igrači u novom redu unose 'da ili 'ne', ovisno o želji za ponovnom igrom. Ako igrači unesu 'da', program se resetira. U slučaju da igrači unesu bilo što drugo, poput 'ne', program se zatvori.