

Pravila

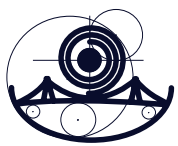
Studenti su dužni odabrati *jednu* od navedenih tema projekta i svoj odabir javiti na lborozan@mathos.hr do kraja lipnja 2019. Projekt se smatra položenim nakon odrađenog praktičnog dijela i izlaganja. Izlaganje je moguće odraditi nakon završenog praktičnog dijela i dogovara se putem već navedene e-mail adrese. Također, moguće je definirati vlastiti projekt u dogovoru s izvođačima kolegija.

Zadatak 1 (Tetris).

Implementirajte igru Tetris (<https://en.wikipedia.org/wiki/Tetris>) za jednog igrača u programskom jeziku JACK. Vaša igra mora sadržavati sljedeće elemente.

- Prilikom pokretanja igre, igrač bira težinu igre na proizvoljnoj skali. Odabrana težina igre utječe na bodovanje i brzinu padanja pojedinih blokova.
- Korisničko sučelje treba u svakom trenutku igre prikazivati težinu, broj trenutno ostvarenih bodova te nadolazeći blok.
- Igrača ploča sastoji se od proizvoljnog broja redova proizvoljne dimenzije koji se protežu od dna do vrha ekrana u koje će padati blokovi.
- Mora postojati barem 5 različitih vrsta blokova (ne nužno istih kao u standardnoj igri Tetris).
- Prilikom padanja blokova, moguće ih je micati tipkama u lijevo i desno (npr. strelice lijevo i desno) te rotirati (npr. strelice gore i dolje).
- Igra završava nakon što je pao blok čiji se dio nalazi u zadnjem redu igrača ploče. Ostaje prikazano bodovno stanje i težina.

Osim gore navedenih elemenata, dodatno će se bodovati izrada generatora nasumičnih brojeva koji određuje kojim redosljedom će blokovi padati te sustav za crtanje pojedinih objekata na zaslonu koji se brine da se prilikom crtanja objekata odrađuju samo one operacije koje su zaista nužne.



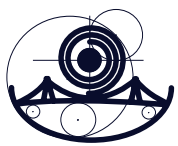
Zadatak 2 (Snake).

Implementirajte igru Snake

([https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))) za jednog igrača u programskom jeziku JACK. Vaša igra mora sadržavati sljedeće elemente.

- Prilikom pokretanja igre, igrač bira težinu igre na proizvoljnoj skali. Odabrana težina igre utječe na bodovanje i brzinu kretanja zmije.
- Igra se odvija na polju proizvoljnih dimenzija. Smatramo da je polje zatvoreno, tj. igrač gubi ako njegova zmija "udari" u rub polja.
- Zmija kreće na sredini polja na kojem se periodički stvaraju blokovi koje zmija jede tako da joj se glava nađe na njihovom mjestu. Kada zmija pojede blok njezina duljina raste. Pripazite da se blokovi koje zmija može pojesti ne stvore u njezinom tijelu.
- Korisničko sučelje treba u svakom trenutku pokazivati bodovno stanje (ovisi o težini i broju pojedenih blokova) i težinu.
- Glava zmije konstantno se kreće naprijed brzinom koja ovisi o težini. Njezin smjer se određuje strelicama (gore, dolje, lijevo i desno).
- Igra završava kada se glava zmije "sudari" s rubom polja ili bilo kojim dijelom svog tijela. Ostaje prikazano bodovno stanje i težina.

Osim gore navedenih elemenata, dodatno će se bodovati izrada generatora nasumičnih brojeva koji određuje koordinate pojavljivanja blokova koje zmije može pojesti. te sustav za crtanje pojedinih objekata na zaslonu koji se brine da se prilikom crtanja objekata odraduju samo one operacije koje su zaista nužne.

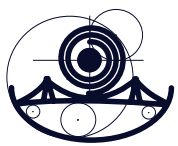


Zadatak 3 (Križić-Kružić).

Implementirajte igru Križić-Kružić (<https://en.wikipedia.org/wiki/Tic-tac-toe>) za jednog igrača u programskom jeziku JACK. Vaša igra mora sadržavati sljedeće elemente.

- Prilikom pokretanja igre, igrač bira hoće li igrati prvi ili drugi.
- Igra se odvija na polju dimenzije 3×3 u koja igrač i računalo (AI) naizmjenično stavljaju svoje znakove. Znakovi ne moraju nužno biti "X" i "O".
- Računalo svoje poteze mora igrati barem heuristički (ne nužno optimalno).
- Igrač svoj potez igra putem tipkovnice. Igračev potez se ponavlja dok potez koji je odigrao ne bude valjan.
- Igra završava s pobjedom jednog igrača koji je u cijeli redak, stupac ili dijagonalu upisao svoj znak. Rezultat može biti i neriješen ukoliko su ispunjena sva polja bez postojanja pobjednika. Nakon završetka igre ispišite poruku o ishodu igre.

Osim gore navedenih elemenata, dodatno će se bodovati izrada računalnog igrača koji svoje poteze igra optimalno (svaka igra završava bez pobjednika ili pobjedom računala) te sustav za crtanje pojedinih objekata na zaslonu koji se brine da se prilikom crtanja objekata odrađuju samo one operacije koje su zaista nužne.

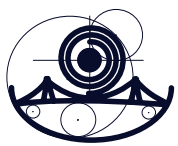


Zadatak 4 (Tekst editor).

U programskoj jeziku JACK implementirajte jednostavni program za uređivanje teksta (https://en.wikipedia.org/wiki/Text_editor). Program mora zadovoljavati sljedeće zahtjeve.

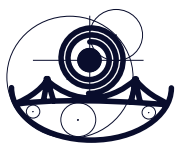
- Program ima dva načina rada: pisanje i uređivanje koj se izmjenjuju pritiskom na proizvoljnu tipku (koja nije slovo, razmak ili znamenka).
- Kada je program u načinu za pisanje, moguće je unositi slova i druge znakove (znamenke, razmak, novi red, itd.) na poziciju kursora. Također, moguće je pomicati kursor strelicama (gore, dolje, lijevo i desno) te brisati znakove koji se nalaze iza kursora. Prilikom pisanja pripazite na prelamanje teksta u novi red.
- Kada je program u načinu za uređivanje, redove na kojima se nalazi kursor moguće je obrisati te poravnati u lijevo, desno ili centrirati. Tipke koje je potrebno pritisnuti kako biste izvršili ove naredbe definirajte sami.

Osim gore navedenih elemenata, dodatno će se bodovati izrada sustava za prenošenje riječi u novi red tokom pisanja ukoliko one ne stanu u trenutni red (riječi su definirane znakovima razmaka) te sustav za crtanje pojedinih objekata na zaslonu koji se brine da se prilikom crtanja objekata odrađuju samo one operacije koje su zaista nužne.



Zadatak 5 (JACK kompajler).

Prema zahtjevima danima u prezentacijama u predlošku implementirajte kompajler koji kod napisan u jeziku JACK pretvara u kod virtualnog stroja. Kompajler je moguće implementirati u bilo kojem programskom jeziku, no preporuča se korištenje gotovog Python predloška koji je odrađen na vježbama.



Zadatak 6 (Floating point jedinica).

U jeziku HDL nadopunite HACK procesor s jedinicom za rad s 16-bitnim brojevima s pomičnim zarezom (floating point jedinica) opisanim u IEEE 754 standardu (https://en.wikipedia.org/wiki/IEEE_754). Procesor mora udovoljavati sljedećim zahtjevima.

- Instrukcijski set mora biti proširen s aritmetičkim operacijama zbrajanja, oduzimanja i negiranja (promjena predznaka) brojeva s pomičnim zarezom.
- Instrukcijski set mora biti proširen s operacijama veće, manje i jednako za brojeve s pomičnim zarezom.
- Floating point jedinica mora vraćati output koji je konzistentan s outputima HACK aritmetičko-logičke jedinice (npr. zr i ng zastavice).

Dodatno će se vrednovati dodavanje operacija množenje i dijeljenje u instrukcijski set. Više o 16-bitnom formatu brojeva s pomičnim zarezom moguće je pronaći na https://en.wikipedia.org/wiki/Half-precision_floating-point_format.