

Računarska grafika

Domaći zadatak #2 (2018/2019) : 3D grafika - JavaFX

Drugi domaći zadatak je iz oblasti 3D grafike i rešava se uz primenu grafičkih paketa biblioteke JavaFX. Studentima je ponuđeno više zadataka, od kojih treba da odaberu i samostalno izrade jedan. Sve elemente rešenja koji nisu specificirani postavkom, studenti definišu na bazi razumnih, profesionalno opravdanih pretpostavki. Osim realizacije traženih funkcionalnosti, u ocenu ulazi kvalitet i izgled grafičkog interfejsa. Postoji mogućnost nadgradnje nekih domaćih zadataka sa ciljem da prerastu u diplomski rad osnovnih studija ili master rad. Studenti zainteresovani za diplomski/master rad treba da se blagovremeno obrate predmetnom nastavniku. Uspešno realizovani zadaci A i B bi mogli da imaju i praktičnu vrednost. Postoji mogućnost da se kvalitetno realizovan diplomski ili master rad zasnovan na zadatku B, na nekomercijalnim osnovama, ponudi institucijama koje rade sa decom i omladinom ometenom u razvoju, u okviru projekta *Lite*.

Zadatak A

Gradski prevoz: Razvoj obrazovnog softvera za pomoć u obuci korišćenja gradskog prevoza

Cilj razvoja je obrazovni softver za simulaciju korišćenja usluga gradskog prevoza. Obučavani treba da nauči kako da obavi sve potrebne radnje poput bezbednog ulaska i izlaska iz vozila. Kretanje se ostvaruje kontrolom figure koju ne treba prikazivati. Od softvera realizovanog kao domaći zadatak se ne očekuje (foto)realističnost prikaza.

Od sredstava za prevoz treba ponuditi autobus i tramvaj, koji se biraju na početku programa. Nakon toga prikazuje se predefinisana scena iz gradskog okruženja koja sadrži jednu zatvorenu jednosmernu putanju duž ulica, za kretanje vozila. Putanja je okružena prizorima grada, poput stambenih zgrada, parkova, itd. Zgrade i stabla nije potrebno detaljno modelirati. Na primer, za zgrade je dovoljno da se predstavljaju kvadrima različitih dimenzija ukrašenih realnim slikama zgrada. Na putanji se nalazi nekoliko stanica na kojima se prevozno sredstvo zaustavlja. Obučavani ne može da se kreće van vozila ili stanice.

Scenario simulacije je sledeći: obučavani na stanici čeka da se vozilo zaustavi i otvori vrata, nakon čega ulazi u vozilo i vrata vozila se zatvaraju. Ukoliko ne zada komandu za ulazak u vozilo na vreme, vozilo odlazi i čeka se sledeće. U slučaju da se obučavani zadrži u zoni vrata, treba da dobije upozorenje da u toj zoni nije bezbedno zadržavanje. Nakon ulaska u vozilo, može da se kreće i da odabere mesto za sedenje.

Nije neophodno detaljno modelirati unutrašnjost i spoljašnjost vozila, ali je potrebno da prozori budu providni (tj. da kroz njih može da se vidi okolina) i da postoje sedišta u vozilu na koja obučavani može da sedne. Potrebno je obezbediti da obučavani ne može da prođe kroz zidove vozila i zatvorena vrata.

Studentima je prepušteno da sami odaberu izgled spoljašnjosti i unutrašnjosti vozila. Vozila mogu da budu niskopodna (bez stepenica). Osim pogleda iz "prvog lica" (tj. animirane figure) obezbediti još tri pogleda: pogled na spoljašnjost vozila (kamera se nalazi ukoso i iznad), pogled na unutrašnjost dotičnog vozila sa prednjeg i zadnjeg kraja vozila.

DIPLOMSKI RAD

Za diplomski rad, pored stavki predviđenih domaćim zadatkom, potrebno je realizovati sledeće funkcionalnosti:

1. Osim prevoznog sredstva, putanjom se kreću druga vozila (na primer automobili).
2. Scena koja sadrži više putanja kretanja vozila gradskog prevoza koje se ukrštaju (isključivo pod pravim uglom) i imaju zajedničke stanice na kojima obučavani može da preseda. Jednom putanjom se kreće samo jedna vrsta vozila gradskog prevoza. Ukrštanja putanja obezbediti semaforima. Svako vozilo treba obeležiti brojem linije na kojoj saobraća.
3. Stanice treba imenovati, a njihova imena prikazati na tabli istaknutoj u okviru stanice. Ispod imena stanice treba ispisati brojeve linija vozila koja se na toj stanici zaustavljaju.
4. Unutrašnjost vozila gradskog prevoza opremiti sa nekoliko displeja koji obaveštavaju putnike o nazivu stanice kojoj se vozilo približava.

MASTER RAD

Za master rad, pored stavki predviđenih domaćim zadatkom i stavki predviđenih za diplomski rad, potrebno je realizovati sledeće funkcionalnosti:

1. Posebna aplikacija – Editor scene i linija gradskog saobraćaja. Potrebno je omogućiti raspoređivanje zgrada, definisanje ulica, stanica i tramvajskih šina (koje ne moraju biti položene duž ulica). Zgrade se biraju iz predefinisane biblioteke objekata. Nije predviđeno kretanje animirane figure van stanice ili vozila pa nije potrebno dodavati trotoar ulicama. Modeli drugih vozila koja se kreću definisanim putanjama se biraju na slučajan način iz predefinisane biblioteke objekata, a korisnik bira željenu gustinu saobraćaja tako što za svaku putanju zadaje broj vozila koja se njom kreću. Svi elementi se pamte u XML fajlovima lekcija.
2. Režim testiranja snalaženja u gradskom saobraćaju: obučavani dobija zadatak da stigne do određene stanice, polazeći od slučajno izabrane stanice. Obučavani sa sobom nosi plan grada kome u svakom trenutku može da pristupi. Obučavani najpre treba da se korišćenjem plana grada informiše gde se nalazi i gde treba da stigne. Obučavani tada treba da označi stanicu gde se trenutno nalazi i stanicu na koju treba da stigne. Tek nakon uspešne identifikacije polazne i ciljne stanice, obučavani se upućuje na čekanje prevoza. Boduju se radnje vezane za korišćenje prevoznog sredstva: bezbedan ulazak u vozilo, ponašanje u vozilu i bezbedan izlazak iz vozila. Posebno se boduje korektan izbor saobraćajnih linija, presedanje na odgovarajućim stanicama i stizanje na određenu stanicu. Bodovi se upisuju u fajl (formata koji se može jednostavno uvesti u *MS Excel*) i/ili u bazu podataka (eventualno, preko veb-servisa). Bodovanje može biti prikazano ispitaniku, a ne mora. Posebna mala aplikacija ili deo aplikacije Editor za prikaz i analizu bodova.
3. Podržati i postojanje putnika na stanicama i u vozilima. Obučavani ne može da interaguje sa ostalim putnicima.

Moguće su još neke dorade.

Zadatak B

Raskrsnica: Razvoj obrazovnog softvera za obuku snalaženja u saobraćaju

Cilj razvoja je obrazovni program za obuku snalaženja u saobraćaju. Sa jedne strane, program treba da se koristi za obuku dece-pešaka, a sa druge strane da pomaže budućim vozačima motornih vozila da savladaju saobraćajna pravila koja se primenjuju na raskrsnicama koje kontolišu semafori. Od softvera realizovanog kao domaći zadatak se ne očekuje (foto)realističnost prikaza, niti da ispuni postavljeni cilj, već samo da predstavlja osnovu za ciljnu aplikaciju.

Opis izgleda raskrsnice je sledeći:

1. Ulice koje čine raskrsnicu se seku pod pravim uglom, dvosmerne su, sa jednom trakom u svakom pravcu
2. Na ulicama postoji obeležena horizontalna signalizacija (trake za kretanje vozila, pešački prelaz, linija zaustavljanja vozila ispred semafora, itd.), vertikalna signalizacija (saobraćajni znaci kao što su znak STOP, znak za prvenstvo prolaza, znak za ograničenje brzine, semafor, itd.)
3. Deo ulice namenjen kretanju vozila je asfaltiran i odvojen od trotoara posebno obeleženim ivičnjakom. Trotoar može da bude popločan ili presvučen asfaltom koji je svetliji od asfalta na delu ulice namenjenom kretanju vozila. Deo trotoara uz ivičnjak može da bude zelena (travnata) površina
4. Ulice treba prikazati na dužini dovoljnoj da se vidi kolona od 10-ak vozila počevši od raskrsnice, u svim pravcima

Ulicama se kreću motorna vozila saglasno svetlosnim signalima sa semafora. Vozila se kreću pravolinijski (ne skreću u poprečnu ulicu na raskrsnici). Program treba da podržava sledeća vozila:

1. putnički automobil (3 modela koji se razlikuju po izgledu)
2. autobus (1 model)
3. kamion (1 model)
4. kamion sa prikolicom (1 model)

Kod automobila i kamiona varirati boju karoserije. Vozila ne moraju da se kreću istom brzinom i nemaju isto ubrzanje (kamion sa prikolicom ubrzava sporije od putničkog automobila, na primer). Vozila ne smeju da se sudaraju, a onim vozilima koja usporavaju treba da se upale odgovarajuća crvena svetla. Vozila se povremeno pojavljuju u odgovarajućoj traci na kraju svake od ulica i kreću se u pravcu raskrsnice. Po prolasku kroz raskrsnicu, vozila produžavaju do kraja ulice, a zatim nestaju. U svakom trenutku se ulicama mogu kretati vozila proizvoljnih vrsta.

Svi događaji (pojavljivanje i kretanje vozila, promena svetlosnih signala) se dešavaju bez intervencije korisnika. Korisniku treba omogućiti sledeće poglede za razgledanje trenutne situacije na raskrsnici:

2. Pogled odozgo. Kod ovog pogleda, posmatrač se inicijalno nalazi iznad centra raskrsnice, na visini sa koje može da vidi krajeve ulica (maksimalna visina). Posmatrač može da se kreće po vertikalnoj osi (može da menja visinu).
3. Pogled iz sredine raskrsnice. Kod ovog pogleda, posmatrač se nalazi iznad centra raskrsnice, malo iznad visine najvišeg vozila. Posmatrač ne može da se kreće, ali može da menja pravac

posmatranja u horizontalnoj ravni i nagib posmatranja u odnosu na horizontalnu ravan (onemogućiti nagibe kojim bi se pogled usmerio ka nebu) .

4. Pogled koji prati vozilo. Kod ovog pogleda, posmatrač se stalno nalazi iznad i iza izabranog vozila (pozicija posmatrača prati kretanje vozila). Inicijalno, pogled treba da bude usmeren tako da se vozilo nalazi u vidnom polju posmatrača. Korisniku omogućiti promenu nagiba pogleda. Takođe, korisniku treba omogućiti da bira vozilo nad kojim će se primenjivati ovaj pogled.

Studenti sami biraju početnu i maksimalnu brzinu kretanja svakog vozila, ubrzanje pri kretanju, kao i dužinu trajanja svetlosnih signala (zeleni i crveni signali nisu istog trajanja u obe ulice), tako da program bude interesantan za posmatranje.

Voditi računa da brzina kojom se animira kretanje vozila ili promena svetla na semaforima ne sme da zavisi od brzine računara na kojem se izvršava program.

DIPLOMSKI RAD

Za diplomski rad, pored stavki predviđenih domaćim zadatkom, potrebno je realizovati sledeće funkcionalnosti:

1. Stvaranje noćnih uslova uvođenjem uličnog osvetljenja i uključivanjem farova na vozilima
2. Prikazivanje zgrada u okolini raskrsnice i duž ulica
3. Kretanje kroz raskrsnicu kada semafor ne radi
4. Ulice sa više traka
5. Mogućnost skretanja vozila u raskrsnici.

MASTER RAD

Za master rad, pored stavki predviđenih domaćim zadatkom i stavki predviđenih za diplomski rad, potrebno je realizovati sledeće funkcionalnosti:

1. Posebna aplikacija – Editor scene (lekcije). Potrebno je omogućiti raspoređivanje zgrada, raspoređivanje saobraćajnih znakova, definisanje ulica i više raskrsnica sa većim brojem pravaca. Ulice koje se završavaju na ivici scene predstavljaju tačke nastajanja i nestajanja vozila. Zgrade se biraju iz predefinisane biblioteke objekata. Modeli vozila se kreću duž ulica, poštujući signalizaciju i ostale učesnike u saobraćaju i mogu da skreću u raskrsnicama (kada se na vozilu uključuje svetlosna signalizacija promene pravca). Moguće je zadati željenu gustinu saobraćaja za svaku tačku izvora saobraćaja (kao broj vozila u minuti). Svi elementi se pamte u XML fajlovima lekcija.
2. Režim testiranja snalaženja pešaka prilikom prelaska raskrsnice: testirano lice upravlja figurom koju nije potrebno prikazivati i dobija zadatak da od zadate početne pozicije na mapi grada stigne do druge pozicije. Obe pozicije se biraju slučajno. U svakom trenutku, osim tokom prelaska ulice, testirano lice može da pogleda plan grada na kome su označene trenutna i ciljna pozicija. U režimu testiranja, dostupan je samo pogled iz prvog lica. Boduju se radnje vezane za bezbedno kretanje: kretanje po trotoaru, prelazak ulice samo na pešačkom prelazu, onda kada je to dozvoljeno saobraćajnim propisima. Bodovi se upisuju u fajl (formata koji se može jednostavno uvesti u *MS Excel*) i/ili u bazu podataka (eventualno, preko veb-servisa). Bodovanje može biti prikazano ispitaniku, a ne mora.

3. Prikazivanje kretanja pešaka po trotoarima i kroz raskrsnice.
4. Podržati i postojanje prioritetnih vozila (saobraćajna policija, hitna pomoć, vatrogasci).

Moguće su još neke dorade.

Zadatak C

Hanojske kule: Razvoj softvera za interaktivno rešavanje problema Hanojskih kula u 3D

Cilj razvoja je softver za interaktivno rešavanje problema "Hanojske kule": na raspolaganju su tri stuba i zadat broj diskova, različitih poluprečnika. Pravila za rešavanje problema nalažu da u jednom trenutku samo jedan disk može biti skinut sa nekog od stubova kao i da se disk može smestiti na neki stub samo ako na tom stubu nema drugih diskova ili ako su svi diskovi smešteni na dati stub većeg poluprečnika od diska koji treba smestiti. Stubovi, po kojima se ređaju diskovi, fiksirani su za horizontalnu podlogu koju treba prikazati.

Softver korisniku treba da ponudi sledeće mogućnosti:

1. Pozicioniranje posmatrača na površi zamišljene polusfere (iznad ravni podloge) opisane oko stubova za igru. Centar polusfere se nalazi na mestu preseka vertikalne ose srednjeg stuba i ravni podloge. Moguće je menjati poluprečnik polusfere. Pozicioniranje i promena poluprečnika sfere se vrši pomoću miša i tastature.
2. Zadavanje tačke posmatranja: tačka posmatranja može biti centar jednog od stubova (inicijalno–srednji stub).
3. Izbor parametara dinamičkog svetla: pozicija, intenzitet, boja, karakteristike osvetljenja
4. Izbor teksture kojim se crtaju stubovi i diskovi: potrebno je ponuditi nekoliko pripremljenih uzoraka (metal, drvo, plastika itd).
5. Zadavanje težine igre, odnosno broja diskova koje je potrebno premestiti.
6. Pokretanje igre, nakon čega korisnik premešta diskove upotrebom miša na sledeći način: klikom dugmeta miša na disk, korisnik označava disk koji želi da pomera; nakon toga, klikom dugmeta miša na neki od stubova korisnik označava stub na koji želi da prebaci označen disk. Ukoliko je izabrana kombinacija diska i stuba u saglasnosti sa pravilima igre, prikazuje se animirano premeštanje diska na novu poziciju. U suprotnom slučaju, izdaje se adekvatno upozorenje. Od trenutka pokretanja igre, meri se vreme trajanja igre (u sekundama) i broj premeštanja diskova, a obe informacije se prikazuju.
7. Pokretanje igre u pokaznom režimu, kada program samostalno rešava problem i prikazuje prebacivanje diskova u postupku rešavanja. Korisnik može da kontroliše brzinu kretanja diskova u procesu njihovog premeštanja.

Za ovaj zadatak nisu predviđene dorade do završnog/diplomskog rada.

Zadatak D

Arena: Razvoj 3D video-igre za jednog igrača

Cilj razvoja je 3D video-igra za jednog igrača, na temu preživljavanja u areni i skupljanja poena.

Zadatak predstavlja unapređenje zadatka rađenog na 4. laboratorijskoj vežbi. Navedene stavke predstavljaju proširenja u odnosu na zahteve za 4. laboratorijsku vežbu, čija se implementacija podrazumeva. Parametre koji ne mogu da se podešavaju od strane korisnika zadati tako da igra bude interesantna (odrediti ih empirijski).

Potrebno je realizovati sledeće dodatne funkcionalnosti:

- Kreirati sat koji rotira oko vertikalne ose. On funkcioniše slično kao novčić, ali daje produženje vremena. Kad preostane 10 (od početnih 60) sekundi do kraja igre, na sredini jedne nasumice odabrane ploče (*tile*) stvori se sat. Igrač može da ga pokupi i tad mu se dodaje 30 sekundi na preostalo vreme. Ukoliko igrač pokupi taj jedan sat i dobije 30 sekundi, onda će, kad ostane 5 sekundi do kraja igre, da se stvori još jedan sat na nekoj od ploča, koji daje 15 sekundi. Posle toga se satovi više ne stvaraju, dakle u toku trajanja igre mogu se stvoriti samo dva sata.
- Igrač ima dva života. Dodati dva (2D) srca u uglu ekrana. Kad igrač dođe u kontakt sa šiljcima, gubi jedan život. Gubitak života naznačiti kratkotrajnom promenom ambijentalnog osvetljenja u crvenkastu boju i promenom srca u uglu ekrana (recimo, nestajanjem jednog srca ili njegovom menjanju izgleda/boje).
- Kreirati srce koje rotira oko vertikalne ose. Poput novčića i sata, stvara se na sredini nasumice odabrane ploče, ali samo ukoliko igrač nema sve živote. Srce se stvara sa izvesnom verovatnoćom (ne stvara se nužno odmah nakon gubitka života, već postoji šansa da se stvori u nekom trenutku nakon toga). Kad igrač pokupi srce, dobija jedan život. Pri gubitku svih života, igra se završava.
- Obezbediti da na jednoj ploči može da se nalazi samo jedan predmet (novčić, sat ili srce), odnosno da se predmet ne stvara na ploči gde već postoji neki. Takođe, onemogućiti da se predmet stvara na ploči na kojoj se igrač već nalazi, kao i da se stvara na ploči iz koje trenutno iskaču šiljci. Pored toga, šiljci ne smeju da iskaču iz polja na kojima se već nalazi neki predmet. (Šiljci smeju da iskaču iz polja na kom se nalazi igrač, to se ne menja.)
- Dodati mogućnost brzog trčanja (sprinta). Pritiskanjem i držanjem tastera SHIFT povećava se brzina igrača dok se taster ne pusti ili igraču ne ponestane energije (maksimalno nekoliko sekundi). Nivo energije prikazati pravougaonikom u uglu ekrana, koji se prazni kako igrač trči. Dok igrač ne trči, zalihe energije se lagano obnavljaju (do izvesnog maksimuma).

DIPLOMSKI RAD

Za diplomski rad, pored stavki predviđenih domaćim zadatkom, potrebno je realizovati sledeće funkcionalnosti:

1. Kreirati početni meni: pri pokretanju igre na ekranu se prikazuje meni sa opcijama: „Počni”, „Rezultati”, „Opcije” i „Kraj”.
2. Tasterom ESC igra se ne gasi, već se pauzira i prikazuje se meni u pauzi igre sa dugmadima: „Nastavi”, „Ponovo igraj”, „Početni meni” i „Kraj”.
3. Druga prostorija: nakon isteka vremena, igri nije kraj, već šiljci prestaju da iskaču i otvaraju se vrata koja vode iz prvobitne prostorije (arene) u drugu. Nakon što igrač uđe u drugu prostoriju, vrata se zatvaraju (animirati otvaranje i zatvaranje vrata) i počinje novo odbrojanje (60 sekundi). Tlo u drugoj prostoriji je slično kao u prvoj, tj. podeljeno je na ploče na kojima se stvaraju novčići, satovi i srca, ali je druge veličine i broja ploča. Uz to, prepreke su drugačije: ne iskaču šiljci, već se iz levog i desnog zida lansiraju projektili koji idu ka suprotnom zidu i koje igrač mora da izbegava. Odgovarajući red ploča menja boju pre lansiranja projektila, da igrač može da reaguje blagovremeno. Levi i desni zid u ovoj prostoriji su travnatom površinom udaljeni od zidova kako se ne bi desilo da igrač može da bude pogoden u trenutku stvaranja projektila, ako se nalazi na ivičnim pločama. Novčići, satovi i srca funkcionišu kao i inače, poeni se dodaju na one koji su sakupljeni u prvoj prostoriji, a broj života ostaje isti kao što je bio nakon završavanja s prvom prostorijom. Pri dodiru s nekim od tih predmeta, igračem, zidom ili drugim projektilom, projektil nestaje.
4. Treća prostorija: analogno kreiranju druge prostorije, s razlikom u preprekama: s plafona (ograding odozgo prostoriju arene ili deo iznad ploča) vise tanka sečiva (giljotine). Sredine su im iznad centara ploča na koje je tlo podeljeno, a sečivo ne može biti duže od 2 ploče. U nasumičnim vremenskim intervalima sečiva se rotiraju pod nasumičnim uglom oko vertikalne ose i padaju na tlo. Nakon rotacije, a pre pada, na tlu naznačiti da će pasti sečivo, na sličan način kao što je to slučaj sa šiljcima (pocrvene ploče na kojima igrač ne sme da stoji). Sečivo se vrlo kratko zadržava na tlu, nakon čega se diže do plafona. Obezbediti da se modeli sečiva međusobno ne seku, kao i da sečivo pri padu neće doći u kontakt s predmetima (novčić, sat, srce). Kad istekne vreme u ovoj prostoriji, igra je (uspešno) završena.
5. Unos imena igrača pre igre i prikaz 10 najboljih rezultata na kraju igre.
6. Dodati pozadinsku muziku i zvučne efekte pri kontaktu igrača sa predmetima u areni, pri pobedi i porazu. Na ekranu sa opcijama predvideti pojedinačno uključivanje/isključivanje, kao i pojačavanje/utišavanje pozadinske muzike i zvučnih efekata.

MASTER RAD

Za master rad, pored stavki predviđenih domaćim zadatkom i stavki predviđenih za diplomski rad, potrebno je realizovati sledeće funkcionalnosti:

1. Realizovati Editor pitanja za kviz u koji se ulazi iz početnog menija. Pitanja ima nekoliko vrsta: (1) višestruki ponuđeni odgovori – samo jedan tačan (radio-dugmad), (2) višestruki ponuđeni odgovori – više tačnih (polja za potvrdu), (3) odgovor da/ne, (4) numerički odgovor, (5) tekstualni odgovor, (6) odgovor spajanjem pojmova iz dve liste. Pitanja mogu pripadati različitim kategorijama, a kategorije mogu formirati stablo. U kvizu se kategorije pridružuju prostorijama. Kviz se može snimiti pod imenom i kasnije modifikovati.
2. Proširiti igru kvizom znanja. Iz početnog menija je moguće odabrati kviz znanja. Nakon što istekne vreme provedeno u određenoj prostoriji arene, igra se zaustavlja i korisniku se postavlja pitanje preko polutransparentne pozadine, kroz koju se providi scena i meri se predviđeno vreme za odgovor. Ukoliko korisnik odgovori tačno na pitanje, dobija broj poena predviđen za to pitanje. Ako pogreši, gubi život, ali ukoliko mu je preostao samo jedan, onda gubi poene. Svakoj prostoriji arene je pridružena odabrana kategorija pitanja iz koje (uključujući potkategorije) se posle nasumično izvlače pitanja. Ako ih ponestane iz te kategorije i njenih potkategorija, izvlače se pitanja iz prve natkategorije (uključujući sve njene potkategorije). Nakon davanja određenog broja tačnih odgovora, korisniku se omogućava da nastavi igru. Za igrača se formira istorija uspeha na odigranim kvizovima znanja (kviz, broj poena, datum i vreme).
3. Realizovati Analizator kojim se može pratiti istorija (progres) igrača na kvizovima znanja. Omogućiti pretragu po imenu igrača i kvizu i opsegu datuma i vremena. Podrazumeva se poslednji kviz, poslednji igrač i poslednjih 7 dana.