Deo I

Početak

Čas 1.

Učenje osnova pravljenja igara

AKO STE IKADA VIDELI ILI IGRALI PONG, onda svakako cenite napredovanje koje je učinjeno poslednjih 30 godina čime nam je omogućeno da imamo video-igre koje se po svom sudiovizuelnom izgledu mogu uporediti sa filmovima. Ako ste se ikada pitali kako ove igre rade, onda se svakako nalazite na pravom mestu. Ja sam svoje putovanje u programiranje video-igara počeo istim divljenjem i radoznalošću koje su stvorile moj sopstveni virtuelni svet koji je živeo po pravilima koje sam ja odredio. Vođen malim znanjem programiranja i željom da saznam gde će me odvesti mašta ja sam naučio osnove pravljenja video-igara. Na nesreću, morao sam da učim na teži način osmišljavajući sopstvena rešenja i učeći na greškama prilikom rešavanja čak i najjednostavnijih problema. Na vašu sreću ovakav pristup nije više neophodan. U ovoj lekciji ćete naučiti osnove pravljenja video-igara.

U ovoj lekciji ćete naučiti:

- kakve vrste video-igara postoje
- šta sve treba uzeti u obzir prilikom dizajniranja nove video-igre
- zbog čega je objektno orijentisano programiranje (OOP) važno za pravljenje video-igara
- kakve vrste alata koristite kao programer video igara.

Čas 1.

Upoznavanje video-igara

Prvih godina video-igre su smatrane poslom koji ne donosi profit. Kasnije je industrija video-igara izrasla u veoma važan deo globalnog poslovanja. Video-igre se danas finansiraju na način koji se može porediti sa holivudskim filmovima i često se koriste slavni glumci, glumice, pisci scenarija, muzičari i drugi profesionalni zabavljači čiji smo rad nekada mogli da povežemo samo sa filmovima. Zapravo, pravljenje savremene video-igre je umnogome slično sa pravljenjem filma. Prave se reklame, testira se tržište i to samo da bi se finansirala igra. Kada igra pređe u postupak pravljenja, postoje timovi dizajnera, animatora i programera – da ne pominjemo ostale talentovane osobe i kompanije koje učestvuju u procesu pravljenja video-igre. Za pravljenje savremenih igara treba potrošiti milione dolara i one mogu doneti ogromnu dobit ljudima koji su spremni da ulože svoje vreme i novac.

Ovu knjigu verovatno čitate jer nemate nekoliko miliona dolara koje biste uložili u projekat video-igre. Ili možda želite da gomilu novca potrošite negde drugde. Bilo kako bilo, možda ipak u vama tinja nada da jedna osoba može da napravi zanimljivu video-igru bez mnogo novca. Ne samo da je moguće već može biti veoma isplativo. Međutim, video-igre su tradicionalno jedan od najsloženijih i najmisterioznijih oblika programiranja, te su zbog toga obeshrabrile mnoge ljude. Moj cilj je da demistifikujem postupak pravljenja video-igara i da vam pokažem da ne morate biti milioner ili genije kako biste se zabavili pravljenjem igara.

Zašto video-igre?

Ako ste strastven igrač onda možda znate odgovor na ovo pitanje. Ipak vredi postaviti pitanje zašto su video-igre toliko popularne. zašto je toliko mnogo ljudi zainteresovano da nauči kako da napravi svoje igre? Ja mislim da to ima veze sa privlačnom idejom stvaranja sopstvenog sveta za koji su ograničenja samo vaše programersko znanje i mašta. Cilj ove knjige je da obogati znanje i da podstakne maštu.

Da biste bolje razumeli zašto je veliki broj ljudi naklonjen video-igrama, razmislite koliko su filmovi popularni u savremenim kulturama. Samo mali broj nas može da pogleda odličan film a da u nekom trenutku ne poželi da bude deo filmske scene. Video-igre nam omogućavaju da napustimo ulogu publike i postanemo akter događaja. Sve video-igre vam u osnovi omogućavaju da učestvujete u izmišljenom svetu čak iako se on sastoji samo od svemira i nekoliko vanzemaljaca. Dok napredujete kroz video-igru to je kao da svaki ekran predstavlja prozor u novu realnost koja vas moli da postanete aktivni učesnik i saznate šta se sve događa.

Dosta je maštanja – šta video-igra znači nekome ko će postati programer video-igara? Sa stanovišta programera, video-igre su interesantne jer zahtevaju veliki broj različitih talenata. Video-igre pred programera postavljaju sve uobičajene tehničke izazove koji se odnose na pravljenje softvera, a često i više od toga, jer obuhvataju ilustracije, animaciju, zvučne efekte i muziku. Sve ovo se ne odnosi na temu video-igre, često se za

video-igre prave čitavi scenariji. Pravljenjem video-igre od početka do kraja vi, zapravo, postajete moderan renesansni umetnik koji je ekspert u nekoliko disciplina. Na taj način ćete, takođe uspešno, spojiti različita interesovanja u mediju koji će biti privlačan za druge ljude. Zbog toga su mnogi od nas zaintrigirani beskrajnim mogućnostima pravljenja video-igara.

Vrste video-igara

Možda se pitate zašto kada govorim o pravljenju igara te igre nazivam video-igre kada se one igraju na računaru. Mada ne volim da cepidlačim kada je u pitanju tehnologija, vredi istaći da je video-igra interaktivni elektronski medij za zabavu koji koristi video-ekran kao svoj primarni izlaz. Drugim rečima, termin *video-igra* se odnosi na sve igre koje koriste ekran i džojstik ili neki drugi kontroler. Nasuprot tradicionalnom fliperu, na primer, koji možete igrati, ali koji se ne oslanja na korišćenje video-ekrana. Video-igre se mogu podeliti u tri kategorije: arkadne igre, konzolne igre i računarske igre.

Arkadne igre su video-igre koje su ugrađene u velika kućišta koja imaju mehanizam za naplaćivanje u koji morate ubaciti novac kako biste mogli da igrate. Arkadne igre predstavljaju početak video-igara i na njih je 80-tih godina priličan broj Amerikanaca trošio teško zarađen novac. Arkadne igre često zavise od hardverskih komponenata i jedinstvenih kontrolnih uređaja što ih odvaja od ostalih tipova video-igara. Mada arkadne igre i dalje postoje, one ipak predstavljaju manji deo tržišta video-igara i nisu bastion inovacija kakav su nekada bile. Da se razumemo, postoje neke veoma dobre nove arkadne igre ali da biste se upoznali sa zaista aktivnim delom tržišta video-igara morate potražiti igre koje nisu arkadne.

Konzolne igre su se pojavile vrlo brzo nakon arkadnih igara i predstavljaju kućne sisteme, od klasičnog Ponga i Atari 2600 sistema do savremenih sistema Sony Playstation 2 i Microsoft XBox. Konzolni sistemi za igranje su napravljeni kao igračke mašine koje su trebale da istisnu arkadne igre kvaliteta igara. Mi se trenutno nalazimo u situaciji u kojoj digitalne tehnologije za zabavu zauzimaju centralno mesto kućnih zabava. Microsoft već ima razrađene planove za XBox konzolu i namerava da je pretvori u sveopšti uređaj za kućnu zabavu. Narednih nekoliko godina trebalo bi da budu interesantne, jer treba da se vidi kako će se konzolne igre uklopiti u tradicionalnu opremu za zabavu.

Računarske igre su poslednja vrsta video-igara koja se pojavila, jer je personalnim računarima bilo potrebno najviše vremena da dostignu tehničke mogućnosti pomoću kojih bi mogli da urade interesantne stvari sa grafikom i zvukom kako bi igre bile zanimljive. Računarske igre danas predstavljaju ogroman deo industrije video-igara i mogu se nositi sa popularnošću i cenom konzolnih igara. Interesantno je da najpopularnije igre imaju verzije za konzole i za računare, te se možete odlučiti da li ćete koristiti računar ili ćete se posvetiti konzolnim sistemima za igranje. Konzola XBox je jedinstvena među konzolama jer ima zajedničku softversku platformu sa računarskim igrama. Ja ovde govorim o DirectXu, Microsoftovom alatu koji je prvenstveno namenjen presonalnim računarima, a koji se sada preneo i na XBox.

Napomena
Na nesreću, u ovoj knjizi se nećemo baviti programiranjem igara pomoću DirectXa. Razlog je to što je DirectX složena tehnologija čija je kriva učenja veoma strma. Umesto da polovinu knjige posvetim osnovama načina rada DirectXa, ja sam odlučio da celu knjigu posvetim učenju načina na koji igre rade. Ako, nakon što pročitate knjigu, odlučite da naučite DirectX ja vam toplo preporučujem knjigu "Programiranje igara pomoću DirectXa za 21 dan" (Teach Yourself Game Programming with DirectX in 21 Days, Clayton Walnum).

Želim da pojasnim različite tipove video-igara jer postoje razlike u njihovom pravljenju. Na primer, arkadne igre se oslanjaju na specijalizovani hardver i programske alate koji su skupi i koji su teško dostupni programeru početniku. Konzolne igre ne dozvoljavaju programerima početnicima da se lako uključe u njihovo pravljenje. Ne samo da su alati za pravljenje ovih igara teško dostupni i skupi već je često neophodno imati određeni skup programerskih znanja koji se obično nauče tokom rada za neku kompaniju koja pravi video-igre. Računarske igre su jedina vrsta video-igara koju pojedinac može napraviti. Alati za programiranje računarskih igara su lako dostupni. Ti alati su uglavnom besplatni ili je njhova cena niska. Programer može iskoristiti svoje znanje programskog jezika, recimo programskog jezika C++, za pravljenje računarskih igara. Ova knjiga govori o pravljenju video-igara pomoću programskog jezika C++ iako se većina koncepata i tehnika odnosi i na arkadne i konzolne video-igre.

Suština dizajniranja igara

Pošto ste se upoznali sa vrstama video-igara i načinom na koji se prave, vreme je da naučite kako se igre dizajniraju. Imate li nekakve ideje o igrama koje želite da napravite ili ne znate odakle da počnete? Bez obzira na odgovor, ja sam ovde da vam pomognem. Imajte na umu da je dobra ideja za video-igru često lakši deo posla. Prevođenje koncepta u realnost je mesto gde počinju da se pojavljuju problemi i kada mnogi od nas odustaju. Sve dok probleme rešavate jedan po jedan i ne preterujete sa detaljima, vi ste na pravom putu.

Prvi korak pri prebacivanju video-igre sa table za crtanje na tastaturu jeste jasna ideja o tome kako igra treba da izgleda. Ne kažem da treba da napravite listu svih scena, svih likova i svakog minuta interakcije. Međutim, veoma je važno da uspostavite minimalna pravila za celokupnu igru koju ste zamislili. Slede ključni elementi na koje treba da se usredsredite kada ih uklapate u koncept video-igre:

- osnovna ideja
- tema
- grafika
- zvuk
- kontrole
- režimi igranja.

Narednih nekoliko odeljaka detaljnije objašnjavaju ove teme.

Iznalaženje osnovne ideje

Najvažniji korak prilikom pravljenja video-igre je određivanje njene osnovne ideje. Da li je to pucačka igra, lavirint, igra u kojoj se prati glavni lik (engl. role-playing game – RPG), vožnja, igra rešavanja zagonetki ili neka kombinacija? Da li zaista imate originalnu ideju koja se sasvim ne uklapa u neki od ovih žanrova? Da li je cilj osloboditi nekoga, elimisati negativce ili istražiti nepoznatu okolinu? U koji vremenski period je smeštena radnja video-igre, odnosno da li se može smestiti u neki? To su samo neka od pitanja na koje morate dati odgovor kada razvijate osnovnu ideju za novu video-igru. Razmislite i sve zapišite. Zapišite sve što vam padne na pamet jer ideje iskrsavaju i zaboravljaju se, a ništa od tih ideja ne bi trebalo izostaviti. To što ćete se naterati da formalizujete ideje i što ćete ih zapisati prouzrokovaće da više razmišljate i da rasčistite mnoge nedoumice.

Ukoliko imate problema prilikom iznalaženja ideje za video-igru, razmislite o uticajima nekih popularnijih igara. Mnoge igre se baziraju na filmovima, neke na istorijskim događajima, dok neke za temu imaju sportove. Neke igre su toliko popularne da su postale tam filmova (Tomb Raider, Final Fantasy, Resident Evil i tako dalje). Većina računarskih igara su modeli sveta koji nas okružuje, bilo zamišljenog ili stvarnog, te ne morate dalje tražiti ideju. Naročito filmovi mogu da posluže pri stvaranju scenarija za video-igru – pazite samo da ne pozajmite previše.

Upamtite da, bez obzira na inspiraciju, vaša igra mora da bude zabavna. Sveukupni cilj svake igre, bez obzira na lepu grafiku i odličan zvuk, jeste maksimalna zabava. Ko ne bi želeo da ceo dan provede tražeći način na koji će se najviše zabaviti? To je, moj prijatelju, prava draž programiranja video-igara. Ako vaša igra nije zabavna onda je odlična grafika, zvuk i glasovi poznatih ličnosti neće spasiti. Neke od najboljih igara svih vremena su imale lošu grafiku i zvuk mereno današnjim standardima, ali količna zabave koju su donosile se ne može izbrisati.

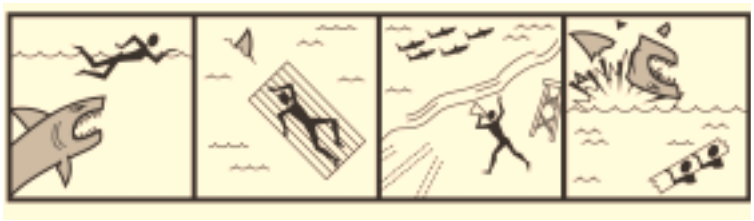
Dobar primer neverovatno jednostavne igre koja je neverovatno zabavna je klasična igra Combat za konzolu Atari 2600. Ova igra se dobijala uz većinu konzola Atari 2600 te je većina igrača mogla da se oproba. Video-igra Combat je bila osakaćena tehnološkim ograničenjima tog vremena što se vidi iz njene grafike i zvuka. Međutim, plan igre Combat je odličan i po mom mišljenju se može nositi sa savremenim igrama. Kada razmislite, jedan aspekt programiranja video-igara koji se nije mnogo promenio tokom godina jeste korisnički unos. Istina, postoje ekskluzivniji džojstici i novi dodaci, kao što je dodatak kojim se može osetiti otpor prilikom pokreta glavnog junaka tokom igranja, ali na kraju dana sve to će postati samo interfejs koji držite u rukama. Ovakvo objašnjenje pomaže da shvatimo zašto su arkadne igre koje su napravljene 1982. godine i dalje popularne, iako su tehnički inferiorne sa savremenim igrama u gotovo svakom pogledu. Dizajneri tih igara nisu imali luksuznu 3D grafiku i CD kvalitet zvuka, te su morali da se posvete načinu na koji njihove igre rade.

Ono što želim da istaknem jeste da prilikom pavljenja video-igre morate dati prioritet zabavi. Pošto iznađete osnovnu ideju za video-igru i pošto odlučite da ćete je učiniti zabavnom po svaku cenu, možete se posvetiti njenom scenariju.

Smišljanje scenarija

Nisu li scenariji samo za filmove i komplikovane sineastičke igre? Apsolutno ne. Čak i kada je vaša igra jednostavna akciona igra sa nekoliko likova, pravljenje scenarija vam pomaže da uspostavite raspoloženje i okolinu kao i da smislite kreature koje će nastanjivati svet video-igre. Smeštanje igre u kontekst priče takođe doprinosi približavanju igrača njenom svetu. Sećate li se klasične igre Pong koju smo pominjali na početku lekcije? Iako je ta igra postigla uspeh a da nije imala dobar scenario, ona bi svakako bila mnogo interesantnija da je napravljena tako da iza nje stoji nekakva priča. Možda bi lopta mogla da bude atom koji švrlja unaokolo u oštećenom nuklearnom reaktoru i ako ode sa ekrana onda će se sve istopiti. Vama i vašem prijatelju je dat zadatak da pomerate "atomske deflektore" i da ne dozvolite da atom pobegne. Kada razmislim, možda je za Pong bolje da bude samo par linija i bela tačka, ali mislim da ste shvatili šta sam imao na umu.

Ako se ozbiljno pozabavite scenarijom igre možda biste mogli da napravite skice. Skice se često koriste pri pravljenju filmova i predstavljaju grubi izgled scene. Skice su važan vizuelni alat jer vam pomažu da zamislite igru na osnovu scenarija. Skice vam pomažu da iz vida ne izgubite nit priče kada budete preokupirani programiranjem. Na slici 1.1 vidite niz skica popularnog filma koji možda prepoznajete. Možete li pogoditi koji je to film?



Slika 1.1
Čak i niz skica može preneti priču koja se lako prepoznaje.

Predajete se? Ovo je moja prezentacija filma "Ralje" (*Jaws*), verovatnog mog najomiljenijeg filma svih vremena. Kao što vidite na slici, čak i moje ograničene umetničke sposobnosti mogu preneti osnovnu temu. Naravno, skica za pravu igru ili film ne bi bilo samo četiri.

Vizualizacija grafike

Moja umetnost skiciranja je odličan uvod u sledeći važan deo pravljenja igre: grafiku. Iako video-igre mogu biti izuzetno zabavane čak i kada nemaju kitnjastu grafiku, svaku igru možete značajno unaprediti grafikom koju koristite. Važno je odlučiti se za nivo grafike koji će odgovarati vašim ciljevima i resursima. Nekada su morale da se u obzir uzmu grafičke mogućnosti računara ali su danas čak i slabiji računari dovoljni za video-igre srednje i umerene složenosti. Ipak, morate se držati 256 boja kako biste igre učinili bržim ako se igraju na starijim računarima. Zapamtite, što više boja koristite u video-igri to će računaru biti potrebno više vremena da manipuliše i iscrta slike video-igre.

Napomena
Postoje odlični grafički alati koji omogućavaju da napravite neverovatno dobre grafičke efekte a da ne morate obavezno biti umetnik. S druge strane, ako imate prijatelja koji je bolji crtač od vas, vredi ga angažovati da vam pomogne oko pravljenja video-igre.

Ekranska rezolucija je veoma važan faktor savremenih video-igrara. Ako primetite da slika treperi, a da vaš monitor proizvodi čudne zvuke kada prvi put pokrenete komercijalnu videoigru, onda znate šta su to različite rezolucije. Najčešća rezolucija koja se danas koristi na Windows PC računarima je 800x600, što znači da se prikazuje 800 piksela (obojenih grafičkih kvadrata) po širini ekrana u 600 redova. Ako posedujete monitor koji je veći od 15 inča vaša rezolucija može biti mnogo veća; moja rezolucija je trenutno 1280x1024. Rezolucija je značajna jer direktno utiče na veličinu grafike u video-igrama. Mnoge komercijalne igre su napravljene tako da mogu da rade u više različitih rezolucija ali nemojte žuriti da to ugradite u svoje video-igre. Zapravo, u ovoj knjizi sam odlučio da se držim stalne veličine ekrana za svaki primer, što je pristup koji se pokazao kao sasvim dobar i koji pojednostavljuje grafiku igre.

Sledeće o čemu vredi razmisliti prilikom dizajniranja video-igre jeste perspektiva igrača. Da li će prostor za igru biti 2D, 3D ili njihova kombinacija? Da li će igrač moći da na ekranu vidi svog junaka ili će sve biti u prvom licu kao kada bi svet posmatrao očima junaka koji kontroliše? Sve ovo ima ogroman uticaj na složenost programiranja video-igre te se treba odlučiti još u ranoj fazi njenog projektovanja.

Pravljenje zvuka

Zvuk je element video-igre koji možda ne izgleda važan koliko grafika, sve dok ne pokušate da igrate bez njega. Ozbiljno, isključite zvučnike dok igrate svoju omiljenu igru i obratite pažnju kako to utiče na sveukupni utisak. S druge strane, ako su plan igre i grafika dovoljni da vašu igru učine primamljivom, onda će zvuk biti "šlag na tortu". Veoma je važno koristiti zvuk na svakom mestu gde to možete. Zvučni efekti su minimum, a ako zaista želite to uradite kako treba, razmislite o korišćenju muzike.

Prvo što morate odlučiti u vezi zvuka jeste njegov kvalitet, što može drastično uticati na količinu potrebne memorije i sveukupne performanse video-igre. Zvučne efekte ćete praviti pomoću zvukova koji su zapisani u WAV datoteke. Morate odabrati učestalost semplovanja, da li će zvuk biti mono ili stereo i da li ćete koristiti 8 ili 16 bita za zapisivanje svakog sempla. Daleko više o zvuku ćete naučiti u Lekciji 13, "Upoznavanje sa digitalnim zvukom i muzikom", ali za sada treba da zapamtite da oni direktno utiču na kvalitet i performanse zvuka. Na primer, CD kvalitet zvuka se sempluje na 44kHz u stereo tehnici sa 16 bitova. CD kvalitet zvuka obezbeđuje najbolji kvalitet, ali zato zauzima ogroman prostor. Uobičajeni muzički CD ima oko 70 minuta te je potrebno 600MB za zapisivanje te muzike. Više je nego verovatno da ćete se zadovoljiti manjim kvalitetom zvuka kako biste smanjili potrebe za prostorom na disku i kako biste poboljšali performanse.

Napomena
Semplovanje se odnosi na postupak pretvaranja audio signala u digitalni format koji se može zapisati i reprodukovati na računaru. Više o semplovanju ćete naučiti u Lekciji 13.

Muzika se u video-igrama često obrađuje na drugačiji način od zvučnih efekata. Iako je muziku moguće snimiti kao semplovan zvuk, ipak zauzima mnogo prostora. Zbog toga je *MIDI (Musical Instrument Digital Interface)* muzika odlična alternativa i predstavlja standard za reprodukovanje muzičkih kompozicija na računarima. Za razliku od semplovane muzike, MIDI muzika zahteva navođenje nota i njihovo pridruživanje virtuelnim muzičkim instrumentima. Pravljenje MIDI muzike može biti veoma teško ako nemate muzičko znanje ali možete prepraviti postojeću MIDI muziku tako da odgovara vašim potrebama. Na sreću, sve što je potrebno da znate u ovoj etapi pravljenja video-igre jeste da u obzir treba da uzmete pravljenje koda koji može da reprodukuje MIDI muziku.

Kontrole za igranje video-igre

Kontrole koje se koriste za interakciju sa igračem su veoma važan deo dizajniranja video-igre. Veoma je važno podržati što je moguće više ulaznih uređaja (miš, tastatura, džojstik i tako dalje) tako da igrači koji daju prednost nekim uređajima mogu da ih koriste. Igrači koriste širok spektar kontrola, a vaš je posao da odlučite koje od njih vredi podržati i kako će se koristiti za igranje. Nije na odmet podržati više takvih uređaja.

Najmanje što možete uraditi jeste podrška za miša i tastaturu. Takođe možete planirati da podržite džojstik ako to ima smisla za video-igru. Na vama je da odlučite da li ćete podržati još neke uređaje pored ova tri. Neki ulazni uređaji se neće moći koristiti u vašoj video-igri pa je donošenje odluke, u tom slučaju, veoma lako. Pored volana, šlemova i rukavica postoje mnogi drugi sofisticirani ulazni uređaji koje možete podržati ako to želite. Ko zna kakvi će se sve novi uređaji pojaviti.

Napomena
Game Pad radi slično džojstiku te zbog toga ne utiče mnogo na programiranje video-igre. U igri u kojoj se može koristiti džojstik, verovatno se može koristiti i Game Pad.

Režimi igranja

Poslednje o čemu treba razmisliti, a tiče se sveukupnog dizajna video-igre, su režimi igranja. Hoće li to biti igra za jednog igrača, dva igrača ili će to biti mrežna igra koju će igrati veliki broj igrača? Ako više igrača može da igra video-igru, da li se oni međusobno takmiče, da li u igri pomažu jedan drugome ili i jedno i drugo? Ove važne

odluke imaju veliki uticaj na zabavnost igre i tehničke probleme prilikom pravljenja igre. Istina, mrežna igra za više igrača donosi ozbiljne tehničke probleme na nivou programiranja. Ne želim da vas odvratim od pravljenja mrežnih igara; želim samo da naglasim da je to prilično napredno polje programiranja.

Da biste bolje razumeli šta sve morate uzeti u obzir prilikom određivanja režima igara, dozvolite mi da se poslužim primerom. Napravio sam jednostavnu video-igru Combat Tanks koja liči na staru Atari 2600 video-igru Combat. Prilikom pravljenja igre predvideo sam borbu između dva igrača. Tada kada sam pravio igru, mene nije zanimalo pravljenje podrške za mrežu, jer je to tada bilo veoma teško programirati, tako da su dva igrača na istom računaru koristila tastaturu kako bi upravljala tenkovima. Nakon testiranja se ispostavilo da bi i igra za jednog igrača takođe bila zabavna. Bio sam u zabludi da će na računaru biti lako dizajnirati tenk koji može da razmišlja i reaguje kao igrač. Tada sam bio veoma naivan.

Veoma brzo sam naučio da je ulivanje inteligencije protivnicima koje kontroliše računar veoma nezgodan posao. Sve strateške odluke koje mi ljudi pravimo dok igramo veoma teško se usađuju u protivnike koje kontroliše računar. Zbog toga sam ja smislio trik. Umesto da postoji samo jedan inteligentan protivnik, ja sam se odlučio za glupu armiju protivnika. Rezultat je bila igra sa režimom za dva igrača i režimom za jednog igrača koji treba da uništi sve protivnike. Igra za jednog igrača je i dalje bila dovoljno zabavna, a ja nisam morao da se nosim sa "učenjem" računara da bude prepreden koliko i ljudsko biće.

Napomena

Iako je moja igra sada stara gotovo deset godina, još uvek je zabavna. Ako želite da je probate onda je možete preuzeti sa mog Web sajta na adresi http://www.michaelmorrison.com/compgames.html. Zapravo, ovu igru sam napravio sa prijateljem Rendijem Vimsom (Randy Weems), koji mi je ulio većinu znanja o programiranju igara.

Molim vas da shvatite da se ja ne zalažem za laka rešenja kada govorim o određivanju režima igranja kao što sam to nekada radio. Ja samo želim da istaknem da postoji mnogo načina da se dobije željeni rezultat. Za mene je najjednostavnije rešenje koje dovodi do dobrih rezultata bilo dovoljno, jer mi je omogućilo da u video-igru unesem interesantne neprijatelje kao što su vojnici, bombarderi i helikopteri. Kreativnost vas često može dovesti do boljih rezultata nego rešavanje tehničkih problema.

Objektno orjentisano programiranje i video-igre

Pošto imate nekakvo iskustvo sa programskim jezikom C++ onda ste, bez sumnje, čuli za objektno orjentisano programiranje. *Objektno orjentisano programiranje*, odnosno OOP, je programerska tehnika koja delove programa tretira kao objekte, nasuprot delovima programskog koda. OOP je od velike koristi u programiranju igara jer ih je

lakše razumeti kada ih podelite u objekte. Zbog toga su programski jezici kakav je C++ idealni za programiranje video-igara. U naredna dva odeljka ću vam osvežiti znanje OOP-a, a zatim ću ga smestiti u kontekst programiranja video-igara.

Razumevanje objektno orjentisanog programiranja

Svrha objektno orjentisanog programiranja je učiniti programiranje bližim ljudskom načinu razmišljanja. Ljudi razmišljaju tako što koriste stvari sa određenim značenjem (objekte). Na primer, ako opisujete hokej onda govorite o klizalištu, igračima, mrežama i paku. Iz ugla računara igra hokeja se ne razlikuje mnogo od bilo kog drugog programa – to je samo gomila nula i jedinica. Programiski jezici omogućavaju programerima da se oslobode razmišljanja pomoću nula i jedinica. Objektno orijentisani programski jezici su unapređenje proceduralnih programskih jezika jer omogućavaju korišćenje objekata.

Klase objekata su važan deo objektno orjentisanog programiranja. *Klasa* (engl. class) je kategorija objekata. Drugi način na koji se može objasniti klasa jeste da se kaže da je objekat *instanca* klase. Ako pravite računarsku igru hokeja onda možete napraviti klasu igrača hokeja. Igrači koji se pojavljuju u igri su instance te klase. Na slici 1.2 vidite kako su objekti igrača hokeja povezani sa klasom igrača.

Slika otkriva kako se nekoliko objekata pravi iz jedne klase koja služi kao šablon za objekte. Ako umesto hokeja kao analogiju upotrebimo hranu, onda klasu možete zamisliti kao modlu za kolačiće a objekte kao kolačiće. Druga važna karakteristika klasa je to što omogućavaju pravljenje *podklasa.* Podklase imaju dodatna svojstva i karakteristike kako bi bile specijalizovanije. *Podklasa* nasleđuje (engl. inherit) svojstva i karakteristike od *roditeljske* klase (engl. parent class) i potom dodaje svoje. Na ovaj način je podklasa slična detetu jer osobine nasleđuje od roditelja, ali ima i nove osobine.

Slika 1.2
Nekoliko objekata igrača hokeja se može napraviti iz jedne klase igrača.

Korišćenje objektno orjentisanog programiranja u video-igrama

Kako biste razumeli prednosti koje objektno orjentisano programiranje donosi u programiranje video-igara onda video-igru treba da zamislite kao apstraktnu simulaciju. Ako razmislite o većini igara koje ste videli ili igrali onda teško da se možete setiti neke igre koja nije simulacija nečega. Sve avanture i sportske igre su očigledno simulacije.

Igre koje se odvijaju u dalekom svemiru su objekti modela koji postoje u nekoj vrsti virtuelnog sveta. Znajući da su igre modeli svetova onda možete uspostaviti vezu po kojoj većina stvari u video-igri (pejzaži, kreature i tako dalje) odgovara stvarima u tim svetovima. Kada jednom igru prevedete u kolekciju "stvari" onda možete koristiti tehnike objektno orjentisanog programiranja. Sve ovo je moguće jer se stvari lako mogu pretočiti u objekte u okruženju objektno orjentisanog programiranja.

Kako biste bolje razumeli o čemu govorim, razmotrite objektno orjentisani dizajn jednostavne avanture. Objekat koji predstavlja svet ove igre može sadržati informacije kao što su mapa tog sveta i slike koje predstavljaju vizuelizaciju te mape, kao i vreme. Svi drugi objekti u igri bi sadržali informacije o poziciji kojima se opisuje gde se svet nalazi. Ove informacije bi mogle da budu koordinatne vrednosti kojima se određuje pozicija objekata na mapi sveta.

Objekat za glavnog junaka u igri obuhvataju informacije kao što su životna energija i predmeti koji se skupljaju tokom igre (oružija, lampe, ključevi i tako dalje). Sama igra bi sa glavnim junakom komunicirala tako što bi mu govorila da se kreće u različitim pravcima, a sve to na osnovu unosa korisnika. Predmeti koje bi glavni junak nosio sa sobom se takođe mogu napraviti kao objekti. Na primer, objekat lampe bi verovatno beležio koliko je goriva ostalo u lampi i da li je uključena ili ne. interakcija sa lampom bi obuhvatala uključivanje i isključivanje. Objekat lampe bi bio dovoljno pametan da smanji potrošnju goriva kada je lampa isključena.

Kreature u video-igri bi bile napravljene na osnovu jedne klase koja opisuje opšte karakteristike svih kreatura. Te karakteristike bi mogle da budu životna energija, agresivnost i količina štete koju nanose kada se upustite u borbu sa njima. Zatim bi se napravili objekti specifičnih kreatura koje bi bile puštene u virtuelni svet video-igre. Objekti kreatura i glavnog junaka se svi mogu izvesti iz iste klase organizma. Za razliku od objekta glavnog junaka, kojim upravlja korisnik, objekte kreatura bi kontrolisala nekakva vrsta inteligencije koja bi bila programirana u video-igru. Na primer, agresivnije kreature bi uvek mogle da jure glavnog junaka kada se zajedno sa njim nalaze na ekranu, dok bi pasivne kreature imale tendenciju da se sakrivaju i izbegavaju kontakt. Mogli biste da napravite podklase kreatura kako biste poboljšali glavnu kreaturu dajući joj jedinstvene sposobnosti, kao što su recimo, mogućnost da leti, pliva ili da bljuje vatru. Na slici 1.3 vidite kako bi se mogle organizovati klase u ovakvoj hipotetičnoj avanturi.

Slika 1.3
Klase u hipotetičkoj avanturi su igrađene jedna iznad druge kako bi uspostavile virtuelni svet objekata.

Nemojte zaboraviti da su klase šabloni objekata, što znači da su klase koje vidite na slici napravljene tako da predstavljaju šemu za pravljenje objekata video-igre. Međutim, objekti ne mogu da postoje u virtuelnom svetu sve dok ih ne napravite i ne omogućite njihovu međusobnu interakciju. Ponoviću, klasa je modla za kolačiće, a sami objekti su kolačići. Dakle, pravljenje video-igre pomoću objektno orjentisanog programiranja prvo uključuje pravljenje klasa za različite "stvari" u video-igri a zatim pravljenje objekata pomoću tih klasa.

Lepota objektno orjentisanog pristupa pravljenju video-igara jeste da igra više ili manje dobro radi kada se sve postavi na mesto. Drugim rečima, svaki objekat je odgovoran za obavljanje svog dela posla tako da sama igra ne mora da se mnogo brine šta neki od objekata radi. Video-igra samo obezbeđuje okvir u kojem postoje objekti. Ovaj okvir se u osnovi sastoji od prozora u kojem se crtaju objekti i petlje koja obaveštava objekte da je neko vreme proteklo. Ta petlja bi iziskivala određivanje pravca u kojem bi se kreature kretale i da li bi trebalo da napadaju ili beže, a za glavnog junaka bi trebalo obraditi unos korisnika. Ono što želim da istaknem jeste da su objekti nezavisni entiteti koji znaju da se o sebi staraju.

Praktičnija korist objektno orjentisanog programiranja u video-igrama se odnosi na ponovno korišćenje koda. Pretpostavimo da ste napravili avanturu o kojoj smo govorili i da želite da napravite sličnu igru koja će se odvijati u kosmosu. Umesto da pravljenje igre počnete od nule, mogli biste da objekte koje ste napravili za prvu igru iskoristite za novu igru. Istina je da bi neke izmene trebalo napraviti, ali bi većina tih izmena mogle da se naprave pravljenjem novih podklasa umesto da se ponovo prave nove klase. Ponovno korišćenje koda je svakako najveća dobit korišćenja objektno orjentisanog programiranja pri pravljenju video-igara.

Upoznavanje alatima za pravljenje video-igara

Pre nego što se pozabavimo bilo kojim projektom pravljenja video-igara, veoma je važno napraviti skup alata koje ćete koristiti prilikom pravljenja video-igre. Iako postoji veliki broj alata koji bi se lako uklopili u pravljenje složene video-igre, najbitniji alati su:

- kompajler
- grafički alati
- alati za obradu zvuka i muzike.

Kompajler

Bez obzira koji programski jezik koristite za programiranje video-igara, verovatno će vam biti potreban kompajler kako biste programski kod pretvorili u izvršnu datoteku. Pošto se u ovoj knjizi za pravljenje video-igara za Windows koristi programski jezik C++, potreban vam je C++ kompajler koji može da napravi Windows aplikacije. Postoje razni C++ kompajleri, od moćnih komercijalnih razvojnih okruženja, kakva su Microsoft Visual C++ i Borland C++ Builder, do besplatnih kompajlera kakav je DJGPP. U Dodatku B, "Odabir alata za programiranje video-igara" pronaći ćete više detalja o izboru kompajlera za pravljenje video-igara. Primeri igara koji se nalaze u knjizi se mogu kompajlirati pomoću bilo kog od tih kompajlera.

Grafički alati

Slično kompajlerima, postoje razni grafički alati od komercijalnih alata do alata koji se mogu dobiti besplatno. Takođe možete koristiti grafičke alate koji su ugrađeni u Windows (Paint). Ako odlučite da sami iscrtate grafiku onda ćete morati da skenirate crteže koje ćete "očistiti" specijalnim alatom. Naravno, potreban vam je skener, ali skeneri, na sreću, imaju pristupačnu cenu. Čak i ako ne nameravate da sami nacrtate grafiku video-igre, biće vam potreban grafički program ili program za crtanje. Program Paint je standardni program za rad sa slikama u Windowsu koji je iznenađujuće koristan za pravljenje i menjanje osnovnih slika video-igre. Na drugom kraju spektra grafičkih alata nalaze se alati kakav je Adobe Photoshop, koji koriste profesionalci za ostvarivanje izuzetnih specijalnih efekata.

Alati za obradu zvuka i muzike

Na početku ove lekcije ste naučili da su zvučni efekti koji se koriste u video-igrama obično semplovani, što znači da ih snimate pomoću mikrofona ili možda sa audio CD-a. Zvučna kartica u vašem računaru verovatno ima ulaz za zvuk i ulaz za mikrofon koji se mogu koristiti za semplovanje zvuka. Možete koristiti program Sound Recorder koji je ugrađen u Windows za semplovanje zvuka ili možete uložiti novac u komercijalne alate kakav je Cool Edit Pro. Postupak semplovanja je veoma jednostavan: povežite izvor zvuka (mikrofon, CD plejer i tako dalje) i započnite snimanje zvuka pomoću odgovarajućeg softvera. Nakon snimanja je potrebno da obradite zvuk tako što ćete izbaciti, recimo, šumove koji se javljaju pre i nakon zvuka.

Pored semplovanih zvučnih efekata možete eksperimentisati sa pravljenjem MIDI muzike. Da biste to uradili morate koristiti specijalan alat koji je poznat kao alat za pravljenje MIDI muzike. Ova vrsta alata se razlikuje od editora zvuka jer se koristi tako što se unosi muzička kompozicija na način koji je sličan onome koji vidite u notnom sistemu. Zatim različite note pridružuju instrumentima, a rezultat je muzika koju svira

neki muzički sastav. Intuitivniji pristup korišćenja MIDI alata jeste priključivanje muzičke tastature na kojoj se odsvira melodija. Ako vaša zvučna kartica ima MIDI port, a najveći broj kartica ga imaju, ono što odsvirate biće zabeleženo pomoću MIDI softvera. Nakon toga možete menjati muziku kako biste je prilagodili potrebama.

Rezime

Iako niste uposlili ruke oko pisanja koda, ova lekcija je poslužila kao polazna tačka za učenje osnova programiranja video-igara. Cilj ove lekcije je da vam pomogne da usmerite pažnju na osnovne ideje pri pravljenju video-igre na konceptualnom nivou. Upoznali ste se sa osnovnim konceptima dizajniranja video-igre koji su neophodni kako biste krenuli putem pravljenja igara. Naučili ste ponešto o objektno orjentisanom programiranju (OOP) i zbog čega je važno za programiranje video-igara. Lekciju sam završio uvodom u razne alate koje obično koriste programeri video-igara.

Lekcija 2, "Bukvar programiranja igara u Windowsu", uvodi vas u programiranje u Windowsu sa blagim naglaskom na programiranje video-igara. Učićete osnove Windows API-ja i šta može da ponudi programeru igara. Još je važnije to što ćete napraviti konstrukciju Windows aplikacije koja će predstavljati osnovu igara koje ćete napraviti u ostatku knjige.

Pitanja i odgovori

P Zbog čega je ekranska rezolucija važna za video-igre?

O Ekranska rezolucija je fizička veličina ekrana, ne u inčima, već u pikselima. Većina monitora je danas podešena na rezoluciju 800x600 ili veću. rezoluciju koju koristite u Windowsu možete proveriti tako što ćete na radnu površinu kliknuti desnim tasterom miša i što ćete iz kontekst-menija odabrati Properties. U okviru za dijalog Display kliknite karticu Settings pa ćete u donjem delu okvira za dijalog videti rezoluciju. Različite rezolucije možete podešavati tako što ćete pomerati klizač. Video-igre koje ćete napraviti u knjizi ne menjaju rezoluciju već koriste određenu veličinu ekrana.

P Da li treba da ulažem u alate za programiranje?

O Odgovor na ovo pitanje umnogome zavisi od toga koliko ozbiljno ćete se baviti programiranjem video-igara i od toga koliko novca želite da utrošite. Sasvim je sigurno da dobar alat za programiranje može određene delove postupka pravljenja video-igre učiniti lakšim. Međutim, ja vam preporučujem da neko vreme radite sa besplatnim ili jeftinim alatima tako da shvatite vrednost sofisticiranijih alata. Pošto sam sve ovo rekao, vi verovatno nećete investirati novac u kvalitetna okruženja sa C++ kompajlerom kakvo je Microsoft Visual C++. Ako postoji alat koji mogu da

preporučim za kupovinu, onda je to kompajler, a razlog je što veoma lako možete naći veoma kvalitetne besplatne programe za rad sa grafikom ili zvukom. Molim vas da pročitate Dodatak B u kome se nalaze informacije o različitim kompajlerima i razvojnim okruženjima koje možete koristiti kao programer video-igara u Windowsu.

Radionica

Radionica je osmišljena tako da vam pomogne da predvidite moguće probleme, proverite šta ste naučili i da u praksi počnete da primenjujete ono što ste naučili. Odgovori na pitanja se mogu naći u Dodatku A, "Odgovori na pitanja".

Pitanja

1. Kakve vrste video-igara postoje?
2. Šta je najvažnije uzeti u obzir prilikom pravljenja video-igre?
3. Čemu služi niz skica?
4. Kakav je najveći kvalitet zvuka koji možete koristiti u video-igrama?

Vežbe

1. Igrajte neke igre i dobro razmislite kako se mogu podeliti u grupe objekata. Takođe obratite pažnju na to kako su zvuci i muzika uklopljeni u plan video-igre i kako su usklađeni sa interakcijama objekata.
2. Napravite hijerarhiju klasa za vašu video-igru, sličnu onoj koja je napravljena za hipotetičku avanturu koju smo razmatrali u lekciji.