Sadržaj

Predgovor 1

Poglavlje 1: C++, SFML, Visual Studio i započinjanje prve igrice 13

Igrice ... 13
Timber!!! ... 14
Zombie Arena ... 15
Thomas was Late ... 16
Upoznajte C++ ... 17
Microsoft Visual Studio ... 17
SFML ... 17
Postavljanje okruženja za razvoj ... 18
Šta ćemo za Mac i Linux? ... 18
Instalirajte Visual Studio Express 2015 na svoj stoni računar ... 19
Postavljanje biblioteke SFML ... 22
Pravljenje višekratno upotrebljivog šablona projekta ... 25
Planiranje igrice Timber!!! ... 29
Pravljenje projekta iz šablona ... 31
Elementi projekta ... 33
Naručivanje elemenata ... 34
Pravljenje vlastitih zvučnih efekata ... 34
Dodavanje elemenata u projekat ... 34
Razgledanje elemenata ... 35
Razlika ekranskih i internih koordinata ... 36
Početak kodiranja igrice ... 38
Objašnjavanje koda pomoću komentara ... 39
Uključivanje osnova Windowsa ... 39
Funkcija main ... 40
Predstavljanje i sintaksa ... 41
Vraćanje vrednosti iz funkcije ... 42
Izvršavanje igrice ... 42

Otvaranje prozora korišćenjem SFML-a . . . 43
Uključivanje SFML odlika . . . 43
OOP, klase, objekti . . . 44
Using namespace sf . . . 46
SFML VideoMode i RenderWindow . . . 46
Izvršavanje igrice . . . 47
Glavna petlja igrice . . . 48
Petlje While . . . 49
Komentari u C-stilu . . . 49
Ulaz, ažuriranje, crtanje, ponavljanje . . . 50
Otkrivanje pritiska na taster . . . 50
Raščišćavanje i crtanje scene . . . 51
Izvršavanje igrice . . . 51
Crtanje pozadine igrice . . . 52
Priprema sprajta pomoću teksture . . . 52
Duplo baferovanje sprajta pozadine . . . 54
Izvršavanje igrice . . . 55
Rešavanje grešaka . . . 55
Greške konfiguracije . . . 56
Greške prevođenja . . . 56
Greške linkovanja . . . 57
Programerske greške . . . 57
Najčešća pitanja . . . 57
Rezime . . . 58

Poglavlje 2: Promenljive, operatori i odluke – animiranje sprajtova 59

C++ promenljive . . . 59
Tipovi promenljivih . . . 60
Deklarisanje i inicijalizovanje promenljivih . . . 61
Menjanje promenljivih . . . 63
C++ operatori za aritmetiku i dodeljivanje . . . 64
Šta sve mogu izrazi . . . 65
Dodavanje oblaka, drveta i pčelice . . . 67
Priprema drveta . . . 67
Priprema pčele . . . 69
Priprema oblaka . . . 70
Crtanje drveta, pčele i oblaka . . . 71
Slučajni brojevi . . . 72
Generisanje slučajnih brojeva u C++-u . . . 73

Donošenje odluka sa if i else ... 74
Logički operatori ... 74
C++ ključne reči if i else ... 76
Ako krenu na most, pucajte na njih! ... 76
Ili umesto toga uradite ovo ... 76
Pitanje za čitaoca ... 78
Merenje vremena ... 79
Problem brzine kadrova ... 79
SFML rešenje za brzinu kadrova ... 80
Pomeranje oblaka i pčele ... 82
Podarimo pčeli život ... 82
Duvanje u oblake ... 85
Najčešća pitanja ... 89
Rezime ... 90

Poglavlje 3: C++ stringovi, SFML Time, ulaz od igrača i HUD 91

Pauziranje i ponovno započinjanje igrice ... 91
C++ stringovi ... 94
Deklarisanje stringova ... 94
Dodeljivanje vrednosti stringovima ... 94
Manipulisanje stringovima ... 95
SFML Text i Font ... 96
Dodavanje rezultata i poruke ... 97
Dodavanje vremenske linije ... 102
Najčešća pitanja ... 109
Rezime ... 109

Poglavlje 4: Petlje, nizovi, switch, nabrajanje i funkcije – primena mehanike igrica 110

Petlje ... 111
Petlje while ... 111
Petlje for ... 114
Nizovi ... 115
Deklarisanje niza ... 115
Inicijalizovanje elemenata niza ... 116
Pa šta ti nizovi zaista znače za naše igrice? ... 117
Donošenje odluka sa switch ... 118
Nabrajanja klasa ... 119

Početak rada sa funkcijama 121
Vraćeni tipovi funkcija 123
Imena funkcija 125
Parametri funkcije 125
Telo funkcije 126
Prototipovi funkcija 127
Organizovanje funkcija 128
Začkoljica sa funkcijama! 128
Konačna reč o funkcijama – za sada 129
Apsolutno konačna reč o funkcijama – za sada 129
Izrastanje grana 129
Priprema grana 131
Ažuriranje sprajtova grana u svakom kadru 132
Crtanje grana 133
Pokretanje grana 134
Najčešća pitanja 137
Rezime 137

Poglavlje 5: Kolizije, zvuk i uslovi za kraj – Igrica može da se igra 138

Priprema igrača (i drugih sprajtova) 138
Crtanje igrača i drugih sprajtova 140
Obrada ulaza od igrača 141
Obrada početka nove igre 143
Otkrivanje da igrač seče 144
Otkrivanje da je taster otpušten 147
Animiranje odsečenih cepanica i sekire 148
Obrada smrti 150
Jednostavan zvučni efekat 153
Kako funkcioniše SFML zvuk? 153
Kada da odsviramo zvukove 153
Dodavanje koda za zvuk 154
Unapređenje igrice i koda 157
Najčešća pitanja 158
Rezime 158

Poglavlje 6: Objektno orijentisano programiranje, klase i SFML prikazi 159

Planiranje i započinjanje igrice Zombie Arena 159
Pravljenje projekta od šablona 161
Elementi projekta 162

Ispitivanje elemenata ... 163
Dodavanje elemenata u projekat ... 164
OOP ... 164
Šta je to OOP? ... 165
Enkapsulacija ... 166
Polimorfizam ... 166
Nasleđivanje ... 167
Zašto se to tako radi? ... 167
Šta je to klasa? ... 168
Deklaracije klasa, promenljivih i funkcija ... 168
Definicije funkcija klase ... 171
Upotreba instance klase ... 172
Konstruktori i funkcije za očitavanje ... 173
Skakanje po kodu ... 176
Pravimo Player – prvu klasu ... 177
Kodiranje fajla zaglavlja za klasu Player ... 178
Kodiranje definicija funkcija klase Player ... 184
Kontrola kamere igrice sa SFML prikazom ... 193
Pokretanje igračke mašine Zombie Arena ... 195
Upravljanje fajlovima koda ... 199
Početak kodiranja glavne petlje igrice ... 201
Najčešća pitanja ... 209
Rezime ... 210

Poglavlje 7: C++ Reference, tabaci sprajtova i Vertex nizovi 211

C++ Reference ... 212
Rezime referenci ... 215
SFML Vertex nizovi i tabaci sprajtova ... 215
Šta je to tabak sprajtova? ... 216
Šta je to Vertex niz? ... 217
Pravljenje pozadine od pločica ... 218
Pravljenje Vertex niza ... 218
Upotreba Vertex niza za crtanje ... 220
Pravljenje slučajno generisane pomerajuće pozadine ... 220
Upotreba pozadine ... 227
Najčešća pitanja ... 229
Rezime ... 230

Poglavlje 8: Pokazivači, biblioteka standardnih šablona i upravljanje teksturama 231

Pokazivači ... 232
Sintaksa pokazivača ... 233
Deklarisanje pokazivača ... 234
Inicijalizovanje pokazivača ... 235
Reinicijalizacija pokazivača ... 236
Dereferenciranje pokazivača ... 236
Pokazivači su svestrani i moćni ... 238
Pokazivači i nizovi ... 241
Rezime pokazivača ... 242
Biblioteka standardnih šablona ... 243
Šta je to Map ... 244
Deklarišemo Map ... 245
Dodavanje podataka u Map ... 245
Pronalaženje podataka u Mapi ... 245
Uklanjanje podataka iz Mape ... 246
Ispitivanje veličine Mape ... 246
Traženje ključeva u Mapi ... 246
Petlje/iteracija po parovima ključ-vrednost u Mapi ... 247
Ključna reč auto ... 247
STL – rezime ... 248
Klasa TextureHolder ... 248
Kodiranje fajla zaglavlja za TextureHolder ... 248
Kodiranje definicija funkcija za TextureHolder ... 250
Šta smo tačno postigli sa klasom TextureHolder? ... 252
Pravljenje horde zombija ... 252
Kodiranje fajla Zombie.h ... 252
Kodiranje fajla Zombie.cpp ... 255
Upotreba klase Zombie da bi se napravila horda ... 260
Oživimo hordu (vratimo je u život) ... 265
Upotreba klase TextureHolder za sve teksture ... 269
Promena načina na koji pozadina dobija teksture ... 270
Promena načina na koji klasa Player dobija teksturu ... 270
Najčešća pitanja ... 271
Rezime ... 272

Poglavlje 9: Otkrivanje kolizije, sakupljanje i meci 273

Kodiranje klase Bullet ... 274
Kodiranje fajla zaglavlja za Bullet ... 274
Kodiranje izvornog fajla klase Bullet ... 277
Da meci polete ... 282
Include za klasu Bullet ... 282
Kontrolne promenljive i niz metaka ... 282
Dopunjavanje šaržera ... 283
Ispaljivanje metka ... 285
Ažuriranje metaka u svakom kadru ... 287
Crtanje metaka u svakom kadru ... 287
Dajemo igraču nišan ... 288
Kodiranje klase za objekte obnavljanja ... 291
Kodiranje fajla zaglavlja Pickup ... 292
Kodiranje definicija funkcija klase Pickup ... 295
Upotreba klase Pickup ... 300
Otkrivanje kolizija ... 304
Da li je neki zombi pogođen? ... 305
Da li je igrača dodirnuo neki zombi? ... 308
Da li je igrač dodirnuo objekat obnavljanja? ... 309
Najčešća pitanja ... 310
Rezime ... 310

Poglavlje 10: Slojevi prikaza i implementiranje HUD-a 311

Dodavanje svih tekstualnih i HUD objekata ... 311
Ažuriranje HUD-a u svakom kadru ... 315
Crtanje HUD-a, početnog ekrana i ekrana za podizanje nivoa ... 317
Najčešća pitanja ... 321
Rezime ... 322

Poglavlje 11: Zvučni efekti, U/I u fajl i završavanje igrice 323

Čuvanje i učitavanje maksimalnog rezultata ... 323
Priprema zvučnih efekata ... 325
Podizanje nivoa ... 327
Ponovno započinjanje igrice ... 330

Sviranje preostalih zvukova 330
Dodavanje zvučnih efekata kada igrač puni pušku 330
Zvuk pucanja 331
Zvuk kada je igrač pogođen 332
Zvuk kada se pokupi objekat obnavljanja 333
Zvuk spljeskavanja kada se pogodi zombi 333
Najčešća pitanja 335
Rezime 335

Poglavlje 12: Apstrahovanje i upravljanje kodom – Bolje koristiti OOP 336

Igrica Thomas Was Late 337
Elementi igrice Thomas Was Late 337
Pravljenje projekta od šablona 341
Elementi projekta 342
Strukturisanje koda za Thomas Was Late 345
Pravljenje mašine igrice 347
Ponovno korišćenje klase TextureHolder 348
Kodiranje fajla Engine.h 350
Kodiranje fajla Engine.cpp 354
Klasa Engine do sada 362
Kodiranje funkcije main 363
Najčešća pitanja 364
Rezime 365

Poglavlje 13: Napredni OOP – nasleđivanje i polimorfizam 366

Nasleđivanje 366
Proširivanje klase 367
Polimorfizam 369
Apstraktne klase – virtuelne i potpuno virtuelne funkcije 371
Pravljenje klase PlayableCharacter 373
Kodiranje fajla PlayableCharacter.h 373
Kodiranje fajla PlayableCharacter.cpp 378
Pravljanje klasa Thomas i Bob 383
Kodiranje fajla Thomas.h 384
Kodiranje fajla Thomas.cpp 384
Kodiranje fajla Bob.h 386
Kodiranje fajla Bob.cpp 387

Ažuriranje mašine igrice upotrebom Thomasa i Boba 389
Ažuriranje Engine.h za dodavanje instance Boba i Thomasa 389
Ažuriranje funkcije input za kontrolu Thomasa i Boba 390
Ažuriranje funkcije update za pravljenje i ažuriranje instanci PlayableCharacter 391
Crtanje Boba i Thomasa 395
Najčešća pitanja 399
Rezime 400

Poglavlje 14: Nivoi koji se mogu odigrati i otkrivanje kolizija 401

Dizajn nivoa 402
Pravljenje klase LevelManager 407
Kodiranje fajla LevelManager.h 407
Kodiranje fajla LevelManager.cpp 409
Kodiranje funkcije loadLevel 416
Ažuriranje mašine 419
Otkrivanje kolizija 422
Kodiranje funkcije detectCollisions 423
Još otkrivanja kolizije 428
Rezime 430

Poglavlje 15: Prostornost zvuka i HUD 431

Šta je to prostornost? 431
Emiter, slabljenje i slušaoci 432
Kako SFML obrađuje prostornost 432
Pravljenje klase SoundManager 435
Kodiranje fajla SoundManager.h 435
Kodiranje fajla SoundManager.cpp 437
Dodavanje SoundManagera u mašinu igrice 441
Popunjavanje emitera zvuka 442
Kodiranje funkcije populateEmitters 442
Sviranje zvukova 444
Klasa HUD 447
Kodiranje fajla HUD.h 447
Kodiranje fajla HUD.cpp 448
Upotreba klase HUD 451
Rezime 454

Poglavlje 16: Proširivanje SFML klasa, sistemi čestica i šejderi 455

SFML klasa Drawable 455
Alternativan način nasleđivanja klase Drawable 458
Zašto je najbolje da se nasledi Drawable? 458
Pravljenje sistema čestica 460
Kodiranje klase Particle 461
Kodiranje klase ParticleSystem 463
Upotreba sistema ParticleSystem 468
OpenGL, šejderi i GLSL 475
Programabilni kanal i šejderi 476
Kodiranje šejdera fragmenta 477
Kodiranje jednog Vertex šejdera 478
Dodavanje šejdera u klasu Engine 479
Učitavanje šejdera 479
Ažuriranje i crtanje šejdera u svakom kadru 480
Rezime 483

Poglavlje 17: Dok niste otišli... 484

Hvala! 485

Indeks 486

Predgovor

Ova knjiga služi da bi se C++ programiranje učilo na zabavan način. Počnete bez ikakvog iskustva i naučite osnove C++-a, kao što su promenljive i petlje, sve do naprednih tema, kao što su nasleđivanje i polimorfizam. Sve što naučite odmah se primenjuje u praksi na izgradnji tri potpuno upotrebljive igrice.

Ovo su naša tri projekta za ovu knjigu.

Timber!!!

Prva igrica je jedna zarazna, brza imitacija veoma uspešne igrice Timberman, koja se može naći na `http://store.steampowered.com/app/398710/`. Naša igrica, Timber!!!, uvodi nas u sve osnove jezika C++ dok istovremeno gradimo pravu upotrebljivu igricu.

Zombie Arena

Zatim ćemo napraviti jednu mahnitu igricu pucanja na zombije, nalik na Steam hit, Over 9000 Zombies, koja se može naći na `http://store.steampowered.com/app/273500/`. Igrač će imati mitraljez i moraće da se bori protiv sve većih talasa zombija. Sve to dešavaće se u nasumično generisanom, pomerajućem svetu. Da bismo to postigli, učićemo o objektno orijentisanom programiranju i o tome kako nam ono omogućava veliku bazu koda (mnogo koda) koji se lako piše i lako održava. Očekujte uzbudljive odlike kao što su na stotine neprijatelja, brzometno oružje, predmete za skupljanje i jednu ličnost koja može da se podigne na viši nivo nakon svakog talasa.

Thomas was Late

Treća će biti jedna otmena i izazovna logička platformska igrica, koju može da igra jedan igrač ili više njih. Zasnovana je na vrlo popularnoj igrici, Thomas was Alone, koja se može naći na `http://store.steampowered.com/app/220780/`. Očekujte učenje o privlačnim temama kao što su efekti čestica, OpenGL šejderi i podeljeni ekrani za saradnju više igrača.

Šta ima u ovoj knjizi

Poglavlje 1: *C++, SFML, Visual Studio i započinjanje prve igrice.* Ovo je prilično obimno prvo poglavlje, ali ćete naučiti apsolutno sve što nam je potrebno, da bi nam prvi deo naše prve igrice funkcionisao. Evo šta ćemo obuhvatiti u ovom poglavlju:

- Saznajemo za igrice koje ćemo praviti
- Učimo malo C++
- Istražujemo SFML i njegov odnos sa C++-om
- Pregledamo softver Visual Studio, koji ćemo koristiti u celoj knjizi
- Postavljamo okruženje za izradu igrica
- Pravimo višekratno upotrebljiv šablon projekta, koji će nam štedeti vreme
- Planiramo i pripremamo prvi projekt igrice, Timber!!!
- Pišemo prvi C++ kôd u knjizi i pravimo upotrebljivu igricu koja crta pozadinu

Poglavlje 2: *Promenljive, operatori i odluke-Animiranje sprajtova.* U ovom poglavlju ćemo još više crtati po ekranu, a da bismo to postigli moraćemo više da učimo osnove jezika C++. Evo šta nas čeka:

- Učimo sve o C++ promenljivima
- Vidimo kako se menjaju vrednosti koje se čuvaju u promenljivima
- Dodajemo statično drvo, koje će igrač da seče
- Crtamo i animiramo pčelu i tri oblaka

Poglavlje 3: *C++ stringovi, SFML Time, ulaz od igrača i HUD.* U ovom poglavlju posvetićemo oko polovinu vremena učenju kako se manipuliše tekstom i kako se on prikazuje na ekranu, a drugu polovinu proučavanju merenja vremena i kako vidljiva vremenska linija može da stvori utisak žurbe u igrici. Obradićemo sledeće teme:

- Pauziranje i ponovno započinjanje igrice
- C++ stringovi
- Klase SFML Text i SFML Font
- Dodavanje HUD u Timber!!!
- Dodavanje vremenske linije u Timber!!!

Poglavlje 4: *Petlje, nizovi, switch, nabrajanja i funkcije – primena mehanika igrica*. Ovo poglavlje verovatno sadrži najviše C++ informacija od svih poglavlja u knjizi. Ono je puno fundamentalnih koncepata koji će enormno unaprediti vaše razumevanje. Ono će takođe početi da razjašnjava neka nejasna područja koja smo pomalo preskakali, kao što su funkcije i petlje u igricama. Pošto istražimo ceo spisak važnih delova jezika C++, upotrebićemo sve što znamo da bi se glavne mehanike igrice, grane na drvetu, micale. Na kraju ovog poglavlja bićemo spremni za završnu fazu i kompletiranje igrice Timber!!!. Obradićemo sledeće teme:

- Petlje
- Nizovi
- Donošenje odluka sa switch
- Nabrajanje
- Upoznavanje funkcija
- Pravljenje i micanje grana na drvetu

Poglavlje 5: *Kolizije, zvuk i uslovi za kraj – Igrica može da se igra.* Ovo je završna faza prvog projekta. Na kraju ovog poglavlja, imaćete svoju prvu kompletnu igricu. Pošto vam Timber!!! proradi, obavezno pročitajte poslednji odeljak ovog poglavlja, jer se tu nalaze predlozi kako da igrica bude bolja. Obradićemo sledeće teme:

- Dodavanje ostalih sprajtova
- Obrada ulaza od igrača
- Animacija leteće cepanice
- Obrada smrti
- Dodavanje zvučnih efekata
- Dodavanje odlika i unapređivanje igrice Timber!!!

Poglavlje 6: *Objektno orijentisano programiranje, klase i SFML prikazi.* Ovo je najduže poglavlje u knjizi. Biće tu prilična količina teorije, ali ta teorija će nam obezbediti znanje da bismo počeli da koristimo moćne efekte Objektno orijentisanog programiranja (OOP). Osim toga, nećemo gubiti vreme. Svu tu teoriju ćemo odmah da upotrebimo. Pre nego što pređemo na C++ OOP, razmotrićemo plan za naš sledeći projekat igrice. Evo šta ćemo raditi u ovom poglavlju:

- Planiranje igrice **Zombie Arena**
- Učenje o OOP-u i klasama
- Kodiranje klase `Player`
- Učenje o SFML klasi `View`
- Izrada mašine za igricu Zombie Arena
- Upotreba klase `Player`

Poglavlje 7: *C++ Reference, tabaci sprajtova i Vertex nizovi.* U ovom poglavlju, istraživaćemo C++ reference koje nam omogućavaju da radimo sa promenljivima i objektima koji su inače van opsega. Osim toga, reference će nam pomoći da izbegnemo obavezu da prosleđujemo velike objekte među funkcijama, što predstavlja jedan spor proces. Taj proces je spor zato što svaki put kada to radimo mora da se napravi kopija te promenljive ili objekta.

Naoružani ovim novim znanjem o referencama, pogledaćemo SFML klasu `VertexArray` koja nam omogućava da napravimo jednu veliku sliku koja može vrlo brzo i efikasno da se nacrta na ekranu pomoću više slika iz jednog jedinog slikovnog fajla. Na kraju ovog poglavlja imaćemo skalabilnu, slučajnu, pozadinu koja se pomera, pomoću referenci i jednog `VertexArray` objekta.

Sada ćemo govoriti o sledećim temama:

- C++ reference
- SFML Vertex nizovi
- Kodiranje slučajne pozadine koja se pomera

Poglavlje 8: *Pokazivači, biblioteka standardnih šablona i upravljanje teksturama*. U ovom poglavlju ćemo mnogo naučiti, a takođe mnogo uraditi za igricu. Najpre ćemo učiti o važnoj C++ temi, pokazivačima (engl. *pointers*). Pokazivači su promenljive koje čuvaju memorijske adrese. Obično, pokazivač čuva memorijsku adresu neke druge promenljive. To ima neke sličnosti sa referencama ali, videćemo po čemu su mnogo moćniji. Takođe ćemo upotrebiti pokazivač da bismo pratili jednu večito rastuću hordu zombija.

Učićemo i o biblioteci standardnih šablona (STL – Standard Template Library) a to je jedna kolekcija klasa koja nam omogućava da brzo i lako implementiramo uobičajene tehnike upravljanja podacima.

Pošto savladamo osnove STL-a, bićemo u stanju da primenimo to novo znanje na upravljanje svim teksturama u igrici, jer ako imamo 1000 zombija, zaista ne želimo da za svakog od tih zombija učitavamo kopiju slike zombija u grafički procesor.

Takođe ćemo još malo da se udubimo u OOP pa ćemo upotrebiti **statičku** funkciju, a to je funkcija neke klase koja može da se pozove bez instanciranja te klase. Istovremeno, videćemo kako možemo da projektujemo klasu tako da za nju nikad ne može da postoji više od jedne instance. To je idealno kada moramo da garantujemo da će različiti delovi našeg koda koristiti iste podatke.

U ovom poglavlju, imaćemo sledeće teme:

- Učenje o pokazivačima
- Učenje o biblioteci STL
- Implementiranje klase `Texture Holder` pomoću statičkih funkcija i jedne singleton klase
- Primena pokazivača na hordu zombija
- Menjanje postojećeg koda tako da koristi klasu `TextureHolder` za igrača i za pozadinu

Poglavlje 9: *Otkrivanje kolizija, nabavke i meci.* Do sada smo implementirali glavne vizuelne aspekte naše igrice. Imamo jedan lik koji može da se kontroliše i koji trči po areni punoj zombija koji ga jure. Problem je u tome da oni ne deluju jedni na druge. Zombi može da prošeta pravo kroz igrača, a da ne ostavi nikakvog traga. Potrebno nam je da otkrivamo kolizije između zombija i igrača.

Ako će zombiji biti u stanju da povrede i eventualno čak ubiju igrača, jedino je fer da igraču damo neke metke za njegovu pušku. Moraćemo da obezbedimo da njegovi meci mogu da pogode i ubiju zombije.

Istovremeno, ako pišemo kôd za otkrivanje kolizija metaka, zombija i igrača, bilo bi krajnje vreme da dodamo i jednu klasu za objekte kojima se obnavljaju municija i zdravlje.

Evo šta ćemo da uradimo i to ovim redom:

- Pucanje mecima
- Dodavanje nišana i sakrivanje pokazivača miša
- Objekti za obnavljanje
- Otkrivanje kolizija

Poglavlje 10: *Slojevi prikaza i implementiranje HUD-a.* U ovom poglavlju, videćemo pravu vrednost SFML prikaza **View**. Dodaćemo jedan veliki niz SFML `Text` objekata i menjaćemo ih kao što smo radili ranije u projektu **Timber!!!**. Novo je to da ćemo da crtamo HUD pomoću druge instance prikaza. Na taj način, HUD će stajati uredno pozicioniran preko glavne akcije u igrici, bez obzira na to šta rade pozadina, igrač, zombiji i ostali objekti u igrici.

Evo šta ćemo da radimo:

- Dodaćemo tekst i jednu pozadinu na početni ekran i na ekran za kraj igre
- Dodaćemo tekst na ekran za podizanje nivoa
- Napravićemo drugi prikaz
- Dodaćemo HUD

Poglavlje 11, *Zvučni efekti, U/I u fajl i završavanje igrice.* Skoro smo tu. Ovo kratko poglavlje će pokazati kako pomoću standardne C++ biblioteke možemo lako da radimo sa fajlovima koji se čuvaju na disku, a dodaćemo i zvučne efekte. Naravno, mi znamo kako da dodamo zvučne efekte, ali ćemo tačno opisati gde u kôd treba da se stave pozivi funkcije `play`. Osim toga, zaokružićemo i nekoliko nerešenih pitanja da bi igrica bila kompletna.

U ovom poglavlju, učićemo o sledećim temama:

- Čuvanje i učitavanje maksimalnog rezultata
- Dodavanje zvučnih efekata
- Omogućavanje igraču da podiže nivo
- Pravljenje mnogo neprekidnih talasa

Poglavlje 12, *Apstrahovanje i upravljanje kodom – Bolje koristiti OOP.* U ovom poglavlju, prvi put ćemo pogledati poslednji projekat u knjizi. U projektu ćemo imati napredne elemente kao što je usmereni zvuk, koji izlazi iz zvučnika zavisno od pozicije igrača. Imaćemo takođe podeljeni ekran za igru više igrača. Osim toga, ovaj projekat uvodi koncept šejdera (engl. *Shaders*), a to su programi napisani u drugom jeziku koji se izvršavaju direktno na grafičkoj kartici.

Kad budete završili poglavlje 16, *Proširivanje SFML klasa, sistemi čestica i šejderi*, imaćete potpuno funkcionalnu igricu na platformi za više igrača izgrađenu u stilu klasičnog hita Thomas Was Alone. Centralna žiža ovog poglavlja će biti da se projekat započne – konkretno, istraživanje kako da se kôd strukturiše da bi se bolje koristilo objektno orijentisano programiranje.

Biće obuhvaćene sledeće teme:

- Uvod u poslednji projekat, Thomas Was Late, uključujući mogućnosti igranja i elemente projekta
- Detaljan opis kako ćemo poboljšati strukturu koda u odnosu na prethodne projekte
- Kodiranje mašine igrice Thomas Was Late
- Implementiranje funkcionalnosti podeljenog ekrana

Poglavlje 13, *Napredni OOP – nasleđivanje i polimorfizam.* U ovom poglavlju, proširićemo naše znanje o OOP-u uvidom u nešto naprednije koncepte, nasleđivanje (engl. *inheritance*) i polimorfizam (engl. *polymorphism*). Onda ćemo biti u stanju to novo znanje da upotrebimo za implementiranje glavnih likova naše igrice, Thomasa i Boba.

Evo šta ćemo malo detaljnije obraditi u ovom poglavlju:

- Kako proširiti i menjati klasu pomoću nasleđivanja?
- Obrađivanje objekta jedne klase kao da je iz više tipova klasa, pomoću polimorfizma
- Apstrahovanje klasa i kako može da bude korisno projektovati klase koje se nikad ne instanciraju
- Izrada apstraktne klase `PlayableCharacter`
- Upotreba nasleđivanja u klasama `Thomas` i `Bob`
- Dodavanje Thomasa i Boba u projekat igrice

Poglavlje 14, *Nivoi koji se mogu odigrati i otkrivanje kolizija.* Ovo poglavlje će verovatno da nam pruži najviše zadovoljstva u ovom projektu. Razlog je to što ćemo kada ga završimo imati igricu koja će moći da se igra. Iako će preostati još svojstava koja će morati da se implementiraju (zvuk, efekti čestica, HUD i šejder efekti), Bob i Thomas će biti u stanju da trče, skaču i istražuju svet. Štaviše, vi ćete moći da pravite vlastite dizajne nivoa skoro bilo koje veličine i složenosti, jednostavnimređanjem platformi i prepreka u tekstualni fajl.

Sve to ćemo postići obradom sledećih tema u ovom poglavlju:

- Istraživanje kako se projektuju nivoi u tekstualnom fajlu
- Izrada klase `LevelManager` koja će učitavati nivoe iz tekstualnog fajla, konvertovati ih u podatke koje naša igrica može da koristi i pratiti detalje nivoa, kao što su pozicije postavljanja, tekući nivo i dozvoljena vremenska granica
- Ažuriranje mašine igrice tako da koristi `LevelManager`
- Kodiranje jedne polimorfne funkcije za otkrivanje kolizija za Boba i za Thomasa

Poglavlje15, *Prostornost zvuka i HUD.* U ovom poglavlju, dodaćemo sve zvučne efekte i HUD. To smo radili u oba prethodna projekta, ali ćemo ovog puta raditi malo drugačije. Istražićemo koncept prostornosti zvuka i kako SFML pomaže da ovaj inače složen koncept bude lep i jednostavan; osim toga, napravićemo klasu HUD da bismo enkapsulirali kôd koji crta informacije na ekran.

Ove zadatke ćemo obaviti sledećim redom:

- Šta je to prostornost?
- Kako SFML obrađuje prostornost
- Pravljenje klase `SoundManager`
- Razmeštanje emitera
- Upotreba klase `SoundManager`
- Pravljenje klase `HUD`
- Upotreba klase `HUD`

Poglavlje 16, *Proširivanje SFML klasa, sistemi čestica i šejderi.* U ovom završnom poglavlju, istražićemo C++ koncept proširivanja tuđih klasa. Konkretnije, razmotrićemo SFML klasu `Drawable` i prednosti njenog korišćenja kao osnovne klase za naše vlastite klase. Takođe ćemo se površno dotaći teme OpenGL šejdera i videti kako pisanje koda u drugom jeziku GLSL (OpenGL Shading Language), koji može da se izvršava direktno na grafičkoj kartici, može da dovede do glatkih grafičkih efekata koji bi inače bili nemogući. Po običaju, primenićemo te svoje nove veštine i znanja na unapređenje tekućeg projekta.

Ovde je spisak tema po redosledu njihove obrade:

- SFML klasa Drawable
- Pravljenje sistema čestica
- OpenGl šejderi i GLSL
- Upotreba šejdera u igrici Thomas Was Late

Poglavlje 17, *Dok niste otišli*, koje opisuje šta da se dalje radi.

Šta vam treba za ovu knjigu

- Windows 7 Service Pack 1, Windows 8 ili Windows 10
- 1.6 GHz ili brži procesor
- 1 GB RAM-a (za x86) ili 2 GB RAM-a (za x64)
- 15 GB raspoloživog prostora na disku
- 5400 RPM disk
- Video kartica sposobna za DirectX 9 koja radi sa rezolucijom prikaza 1024 × 768 ili većom.

Sav softver koji se koristi u ovoj knjizi je besplatan. Pribavljanje i instaliranje softvera opisano je korak po korak u knjizi. U knjizi se svuda koristi Visual Studio za Windows, ali iskusni korisnici Linuxa i Maca verovatno neće imati nikakve teškoće da izvršavaju kôd i prate uputstva u njihovom omiljenom programskom okruženju.

Kome je ova knjiga namenjena

Ova knjiga je savršena za vas ako za vas važi neko od sledećih tvrđenja: nemate uopšte nikakvo znanje C++ programiranja ili vam je potrebno podsećanje na početničkom nivou, ako želite da naučite kako se prave igrice ili igrice koristite samo kao zanimljiv način da naučite C++, ako imate želju da jednog dana objavite neku igricu, možda na Steam-u, ili prosto želite da se zabavljate i da zadivite prijatelje svojim dostignućem.

Konvencije

U ovoj knjizi, videćete niz tekstualnih stilova za različite vrste informacija. Ovde su neki primeri tih stilova i objašnjenja šta oni znače.

Kodne reči u tekstu, imena tabela baza podataka, imena foldera, imena fajlova, ekstenzije fajlova, imena putanji, lažni URL-ovi, ulaz od korisnika, i Twitter ručice se prikazuju ovako: „Druge kontekste možemo da uključimo pomoću direktive `include`."

Blok koda se prikazuje ovako:

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,Voicemail(u100)
exten => s,102,Voicemail(b100)
exten => i,1,Voicemail(s0)
```

Kada hoćemo da privučemo vašu pažnju na određen deo u bloku koda, relevantni redovi ili reči su istaknuti zacrnjenim slovima:

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,Voicemail(u100)
exten => s,102,Voicemail(b100)
exten => i,1,Voicemail(s0)
```

Ulaz ili izlaz na komandnu liniju je napisan na sledeći način:

```
# cp /usr/src/asterisk-addons/configs/cdr_mysql.conf.sample
/etc/asterisk/cdr_mysql.conf
```

Novi izrazi i **važne reči** prikazane su zacrnjeno. Reči koje vidite na ekranu, na primer u menijima ili okvirima za dijalog, javljaju se u tekstu ovako: „Pritiskom na dugme Next prelazite na sledeći ekran."



Upozorenja ili važne napomene se javljaju ovako.



Saveti i ideje su istaknuti ovako.

Povratne informacije od čitalaca

Povratne informacije od naših čitalaca su uvek dobrodošle. Javite nam šta mislite o ovoj knjizi – šta vam se dopalo ili vam se nije dopalo. Povratne informacije od čitalaca su nam važne zato što nam pomažu da pravimo knjige od kojih ćete zaista imati koristi. Za slanje opštih povratnih informacija, ili postoji područje u kojem imate iskustvo, pa vas zanima da napišete knjigu, jednostavno pošaljite e-poruku na `redakcija@cet.rs`, i spomenite naslov knjige u predmetu poruke.

Preuzimanje koda sa primerima

Fajlove koda sa primerima za ovu knjigu i slikama u boji možete da preuzmete sa sjata `www.cet.rs`. Paket koda za ovu knjigu postoji i na GitHub na adresi `https://github.com/PacktPublishing/Beginning-Cpp-Game-Programming`.

Piraterija

Piraterija materijala sa zaštićenim autorskim pravima je na Internetu stalan problem za sve medije. Izdavačke kuće Pact i CET vrlo ozbiljno shvataju zaštitu autorskih prava i licenci. Ako naiđete na neku nezakonitu kopiju ovih knjiga u bilo kojem obliku na Internetu, molim vas da nam odmah dojavite adresu ili ime veb sajta, da bismo mogli da pokrenemo pravni lek.

Molim da nam na `copyright@packtpub.com` ili `redakcija@cet.rs` dojavite link za koji sumljate da sadrži kradeni materijal.

Cenimo vašu pomoć u zaštiti naših autora i naše sposobnosti da vam pružimo vredan sadržaj.

1

C++, SFML, Visual Studio i započinjanje prve igrice

Dobrodošli u C++ programiranja igrica – Osnove. Neću gubiti vreme, već odmah krećemo na put ka pisanju odličnih igrica za PC, koristeći **C++** i **SFML** koji koristi **OpenGL**.

Ovo je prilično obimno prvo poglavlje, ali ćete naučiti apsolutno sve što nam je potrebno, da bi nam prvi deo naše prve igrice funkcionisao. Evo šta ćemo obuhvatiti u ovom poglavlju:

- Saznajemo za igrice koje ćemo praviti
- Učimo malo C++
- Istražujemo SFML i njegov odnos sa C++-om
- Pregledamo softver Visual Studio, koji ćemo koristiti u celoj knjizi
- Postavljamo okruženje za izradu igrica
- Pravimo višekratno upotrebljiv šablon projekta, koji će nam štedeti vreme
- Planiramo i pripremamo prvi projekt igrice, Timber!!!
- Pišemo prvi C++ kôd u knjizi i pravimo upotrebljivu igricu koja crta pozadinu

Igrice

Naučićemo osnove super-brzog jezika C++, korak po korak, a zatim ćemo upotrebiti to novo znanje tako što ćemo prilično lako dodavati privlačne odlike u tri igrice koje pravimo.



Ako vam išta u ovom poglavlju predstavlja problem, potražite pri kraju odeljke Rešavanje grešaka i Najčešća pitanja.

Ovo su naša tri projekta za ovu knjigu:

Timber!!!

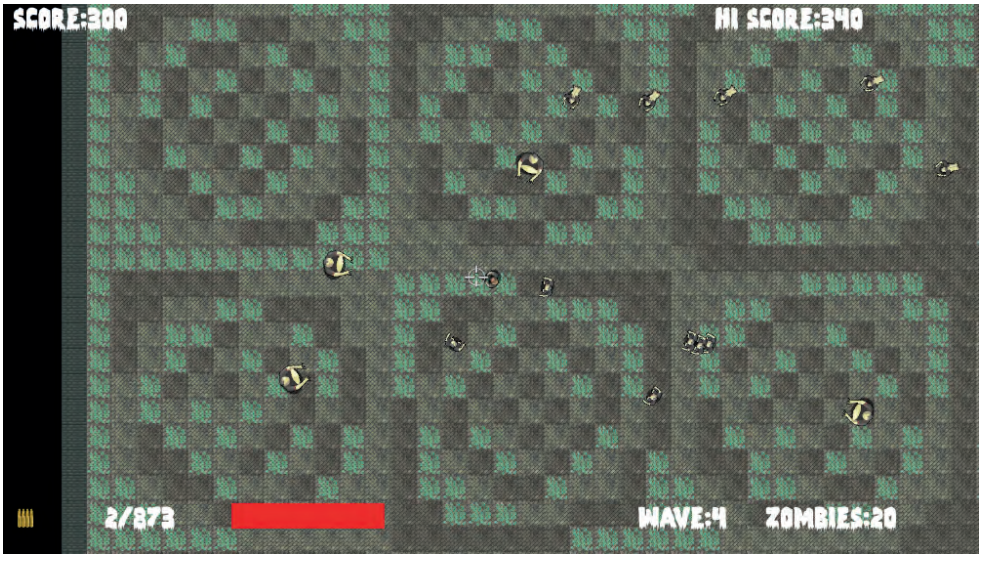
Prva igrica je jedna zarazna, brza imitacija veoma uspešne igrice Timberman, koja se može naći na `http://store.steampowered.com/app/398710/`. Naša igrica, Timber!!!, uvodi nas u sve osnove jezika C++ dok istovremeno gradimo pravu upotrebljivu igricu.

Evo kako će naša verzija igrice izgledati kada je budemo završili i dodali nekoliko poboljšanja u poslednjem trenutku.



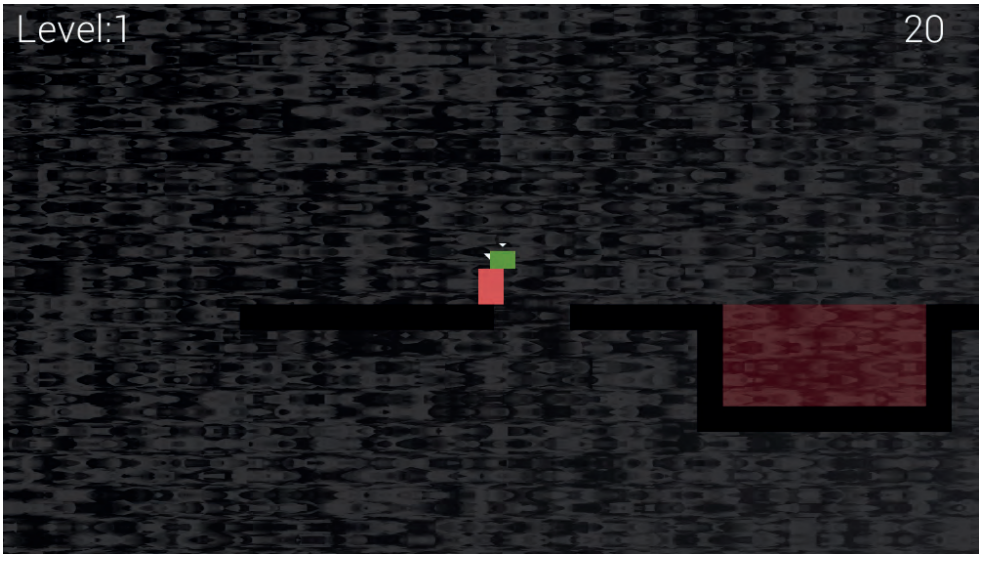
Zombie Arena

Zatim ćemo napraviti jednu mahnitu igricu pucanja na zombije, nalik na Steam hit, Over 9000 Zombies, koja se može naći na `http://store.steampowered.com/app/273500/`. Igrač će imati mitraljez i moraće da se bori protiv sve većih talasa zombija. Sve to dešavaće se u nasumično generisanom, pomerajućem svetu. Da bismo to postigli, učićemo o objektno orijentisanom programiranju i o tome kako nam ono omogućava veliku bazu koda (mnogo koda) koji se lako piše i lako održava. Očekujte uzbudljive odlike kao što su na stotine neprijatelja, brzometno oružje, predmete za skupljanje i jednu ličnost koja može da se podigne na viši nivo nakon svakog talasa.



Thomas was Late

Treća će biti jedna otmena i izazovna logička platformska igrica, koju može da igra jedan igrač ili više njih. Zasnovana je na vrlo popularnoj igrici, Thomas was Alone, koja se može naći na `http://store.steampowered.com/app/220780/`. Očekujte učenje o privlačnim temama kao što su efekti čestica, OpenGL Shaders i podeljeni ekrani za saradnju više igrača.





Ako želite sada da igrate neku od tih igrica, možete to da uradite iz paketa za preuzimanje u folderu **`Runnable Games`**. Prosto dva puta pritisnite odgovarajući **`.exe`** fajl. Videćete u ovom folderu da možete da igrate završene igrice, ili one koje su delimično završene u nekom od poglavlja.

Hajde da počnemo tako što ćemo predstaviti C++, Visual Studio i SFML!

Upoznajte C++

Jedno pitanje koje se možda javlja je, zašto da se uopšte koristi C++? C++ je brz, vrlo brz. Dokaz za to je činjenica da se kôd koji napišemo direktno prevodi u mašinski izvodive instrukcije. Te instrukcije sačinjavaju igricu. Izvršna igrica se nalazi u jednom `.exe` fajlu koji igrač prosto dva puta pritisne da bi igrao.

Proces se sastoji od nekoliko koraka. Najpre **pred-procesor** pregleda da li u naš vlastiti kôd treba uključiti još neki kôd, pa ga dodaje kada je to neophodno. Zatim kompajlerski program prevodi sav kôd u **objektne fajlove**. Na kraju jedan treći program, koji se zove **linker**, spaja sve objektne fajlove u izvršni fajl, a to je naša igrica.

Osim toga, C++ je dobro afirmisan, a istovremeno i izuzetno savremen. C++ je jedan **objektno orijentisan programski** (**OOP**) jezik, što znači da možemo da pišemo i organizujemo svoj kôd na proveren način, tako da naše igrice budu efikasne i podesne za upravljanje. Njegove prednosti, kao i potreba za ovim što pruža pokazaće se kako budemo napredovali kroz knjigu.

Većina preostalog koda koji sam pomenuo je SFML, a o tome ćemo za minut. Malopre pomenuti programi, pred-procesor, kompajler i linker su sastavni delovi razvojnog okruženja Visual Studio **IDE** (**Integrated Development Environment**).

Microsoft Visual Studio

Visual Studio krije složenost pred-procesovanja, prevođenja i linkovanja. On sve to obuhvata kroz jedan pritisak na dugme. Pored toga, pruža nam jedan spretan korisnički interfejs za pisanje našeg koda i za upravljanje onim što će postati jedan veliki komplet fajlova koda, kao i drugih delova projekata.

Mada postoje napredne verzije Visual Studija koje koštaju na stotine dolara, mi ćemo moći da napravimo sve tri igrice u besplatnoj verziji **Express 2015 for Desktop**.

SFML

Biblioteka **SFML** (**Simple Fast Media Library**) nije jedina C++ biblioteka za igrice i multimedije. Mogući su argumenti za korišćenje drugih biblioteka, ali se meni čini da je SFML najbolja, u svakom pogledu. Najpre to što je napisana korišćenjem objektno orijentisanog C++-a. To joj daje brojne prednosti. Većinu tih prednosti ćete osetiti dok budete napredovali kroz ovu knjigu.

SFML je toliko lak za početak da predstavlja dobar izbor ako ste početnik. U isto vreme, on takođe ima potencijal za izradu 2D igrica najvišeg kvaliteta, ako ste profesionalac. Dakle, početnik može na početku da koristi SFML i da ne brine o tome da li će morati ponovo da započinje sa novom kombinacijom jezika/biblioteke kada stekne više iskustva.

Možda je najveća prednost u tome što se u većini savremenog C++ programiranja koristi OOP. Svaka početnica za C++ koju sam ikad pročitao koristi i poučava OOP. OOP je, u stvari, budućnost (i sadašnjost) kodiranja u skoro svim jezicima. Pa zašto biste, ako od početka učite C++, poželeli da programirate na neki drugi način?

SFML sadrži modul (kôd) praktično za sve što biste radili u 2D igrici.

SFML u radu koristi OpenGL, koji može da pravi i 3D igrice. OpenGL je defakto besplatna biblioteka grafike za igrice, ako želite da se izvršavaju na više platformi. Kada koristite SFML, vi automatski koristite i OpenGL.

SFML drastično uprošćava:

- 2D grafiku i animaciju, uključujući igrice sa pomerajućim svetovima.
- Zvučne efekte i reprodukciju muzike, uključujući usmeren zvuk visokog kvaliteta.
- Mogućnost onlajn igranja za više učesnika
- Isti kôd može da se prevodi i linkuje na svim glavnim operativnim sistemima stonih računara, a uskoro i mobilnih!

Opsežnim istraživanjem nije otkriven podesniji način za izradu 2D igrica za PC, čak ni za iskusne programere, a pogotovo ako ste početnik i želite da učite C++ u jednom zabavnom okruženju za igrice.

Postavljanje okruženja za razvoj

Pošto sada malo više znate o tome kako ćemo praviti ove igrice, vreme je da postavimo okruženje za razvoj da bismo počeli da kodiramo.

Šta ćemo za Mac i Linux?

Igrice koje pravimo mogu da se naprave za rad u sistemima Windows, Mac i Linux! Kôd će biti identičan u sva tri slučaja. Međutim, svaka verzija mora da se prevede i linkuje na platformi za koju je namenjena, pa Visual Studio neće moći da nam pomogne za Mac i za Linux.

Ne bi bilo u redu da kažem kako je ova knjiga potpuno prikladna za korisnike sistema Mac i Linux, pogotovo za početnike. Mada će, verovatno, ako ste zagriženi korisnik sistema Mac ili Linux, pa se dobro osećate u tom operativnom sistemu, velika većina dodatnih teškoća na koje možete naići biti u početnom postavljanju radnog okruženja, SFML-a i prvog projekta.

U tom cilju, toplo preporučujem sledeće tekstove za obučavanje koji mogu da zamene sledećih 10 stranica (približno), sve do odeljka *Planiranje igrice Timber!!!*, kada ova knjiga ponovo postaje relevantna za sve operativne sisteme.

Za Linux, pročitajte ovo za opšti pregled:
`http://www.sfml-dev.org/tutorials/2.0/start-linux.php.`

Za Linux, pročitajte ovo kao vodič korak po korak:
`http://en.sfml-dev.org/forums/index.php?topic=9808.0.`

Za Mac, pročitajte ovaj tekst za obuku i povezane članke:
`http://www.edparrish.net/common/sfml-.osx.html.`

Instalirajte Visual Studio Express 2015 na svoj stoni računar

Instaliranje Visual Studija može da bude takoreći tako prosto kao što je preuzimanje fajla i pritiskanje nekoliko dugmadi. Međutim, biće od pomoći da pažljivo opišemo ceo ovaj postupak. Iz tog razloga, vodim vas kroz proces instalacije korak po korak.

Sajt Microsoft Visual Studio kaže da vam je potrebno 5 GB prostora na disku. Međutim, iz iskustva vam predlažem da obezbedite najmanje 10 GB slobodnog prostora. Osim toga, ove cifre su blago dvosmislene. Ako planirate da ga instalirate na sekundarni disk, ipak će biti potrebno i 5 GB na primarnom disku, zato što će ovaj prostor biti potreban bez obzira na to gde instalirate Visual Studio.



Da rekapituliram ovu dvosmislenu situaciju: bitno je imati svih 10 GB prostora na primarnom disku, ako nameravate da instalirate Visual Studio na taj primarni disk. S druge strane, obezbedite 5 GB na primarnom disku i 10 GB na sekundarnom, ako nameravate da instalirate na sekundarni disk. Jeste glupo, znam!

1. Prva stvar koja vam je potrebna je Microsoft nalog i detalji za prijavljivanje. Ako imate Hotmail ili MSN adresu elektronske pošte, onda već imate sve što je potrebno. U suprotnom, možete da se prijavite za besplatan Microsoft nalog na adresi: `https://login.live.com/`.

2. Posetite adresu: `https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx`. Pritisnite **Visual Studio 2015**, zatim **Express 2015 for desktop** a zatim dugme **Downloads**. Na sledećem snimku ekrana vide se tri mesta koja treba pritisnuti:

