

V4T PROJEKTAS

ELEKTRONINIO MOKYMOŠI KURSAS

1 Skyrius – Įvadas į vaizdo žaidimų programavimą

Prieinamos technologijos – 1 dalis

Žaidimų kūrimas apima kelias skirtingas darbo ir žinių sritis, pradedant menine kūryba ir baigiant žemo lygio programavimu. Kiekvienoje srityje yra pagalbos arba tobulinimo įrankiai bei specializuotos technologijos. Daugelį šių įrankių galima naudoti savarankiškai, suteikiant žaidimų kūrėjams galimybę dirbti didžiąją projekto dalį atskirai nuo komandos. Paskutiniuose kūrimo etapuose visas skirtingų komandų ir kūrėjų darbas yra integruojamas į galutinį produktą. Norėdamos tai padaryti, žaidimų kūrimo komandos naudoja programinę įrangą, kuri perima bei surenka visus žaidimo elementus ir sukuria galutinį produktą. Ši programinė įranga gali tvarkyti atvaizdus, garso bei vaizdo įrašus kartu su žaidimo programa ar taisyklių rinkiniu ir taip sugeneruoti tikrąjį žaidimą.

Iš esmės, žaidimo ar bet kurios programinės įrangos kūrimui, reikalingas programavimas. Tačiau tobulėjant programinei įrangai ir augant specializacijai įvairiose srityse, atsiranda skirtumų tarp žaidimų ir programinės įrangos vystymo. Priemonės, kurias naudoja programinės įrangos kūrėjai, yra specializuojamos bei tobulinamos. Dažnai programinės įrangos kompanijos įtraukia naujus įrankius, tinkamus bet kurios programos, tinklalapio vystymui ar jau įtaisytai programinei įrangai. Šių įrankių naudojimas turėtų supaprastinti bei padidinti kūrimo ir vystymo efektyvumą. Tą patį galima pasakyti ir apie žaidimų kūrimo įrankius.

Tačiau, skirtingai nuo kitų programinės įrangos sektorių, žaidimų kūrimo technologijose naudojami tiek specifiniai, tiek paprasti, įvairiose srityse naudojami įrankiai. Yra keletas įrankių, kurie gali būti naudojami kuriant sudėtingus, bet kurio žanro žaidimus, tuo tarpu su kitais įrankiais galima kurti tik specifinio žanro žaidimą. Kartais pačiame žaidime yra įrankių, kurie leidžia išplėsti ar papildyti jo turinį naujais veikėjais ir lygiais, taip sukuriant mišrų žaidimų vystymo įrankį.

Be to, žaidimų kūrime naudojamos ir kai kurios pagrindinės technologijos be kurių neapsieina mokslo ir informacinių technologijų sritys. Pavyzdžiui, trimatės kompiuterinės grafikos (angl. 3D) vaizdavimas yra didelė žaidimų revoliucijos dalis. Ši technologija taip pat yra naudojama kompiuteriniame projektavime ir gamyboje (angl. CAD / CAM), chirurgijoje bei "Google" žemėlapiuose (angl. „Google Earth“).



1.2 Prieinamos technologijos – 2 dalis

Skaitydami toliau rasite informatikos mokslo sričių, tiesiogiai susijusių su žaidimų kūrimu, sąrašą:

- Programų inžinerija (angl. Software engineering). Kaip ir bet kurią kompiuterinę programinę įrangą, žaidimus kuria kūrėjai, kurie privalo būti kvalifikuoti bei išmokyti naudoti naujausias programinės įrangos inžinerijos metodikas.
- Programavimo kalbos (angl. Programming languages). Šios kalbos yra būtinos kuriant bet kokią sudėtingą programinę įrangą.
- Trimatės ir dvimatės grafikos vaizdai (angl. 3D and 2D imaging). Žaidimai tampa sudėtingiausia programinė įranga, kuomet reikia parodyti tam tikrus atvaizdus, animaciją, judesius ir pan. Tai paskatino aparatinės įrangos gamintojus vystyti ir sukurti galingas vaizdo plokštes, galinčias nuolat ir sklandžiai pavaizduoti sudėtingas scenas. Dėl to atsirado tam tikrų standartų, leidžiančių programinės įrangos kūrėjams naudoti šią galingą aparatinę įrangą. Šiuo metu universaliausia yra „OpenGL“ platforma. Be to, modeliavimo programinė įranga, padedanti žmonėms kurti objektus trimatėje erdvėje pakankamai išsivystė ir dabar ją paprasčiau naudotis.
- Mašininis mokymasis ir dirbtinis intelektas (angl. Machine learning and artificial intelligence). Žaidimas prieš kompiuterį reikalauja, kad programuotojas kurtų žaidimą taip, jog žaidėjas susidurtų su įvairiais iššūkiais. Paprastai žaidimas turi kelis „sunkumų lygius“, kad kompiuterio keliamas pasipriešinimas atitiktų žaidėjo įgūdžius. Tačiau nuolat stengiamasi, kad žaidimai, automatiškai prisitaikytų prie žaidėjų, ir išmokytų, kaip bėgant laikui juos nugalėti.

Be to, mes negalime pamiršti technologijų, žinių ir patobulinimų, kuriais grindžiama ne tik kuriamo žaidimo programavimo dalis.

- Kinas, komiksai ir grafinis dizainas. Žaidimai paprastai vaizduoja veikėjus, peizažus ir aplinką įvairiausiais stulbinamais būdais, taip stengiantis priversti žaidėją patirti neužmirštamus nuotykius. Yra gerai žinomi filmų kūrimo būdai, kurie pakartotinai naudojami ir žaidimų kūrime. Šie būdai yra savaime kaip technologija, kurią turi įgyti ir įvaldyti žaidimų kūrėjai. Žaidimo istorijos scenų filmavimas.
- Psichologija. Tam tikru požiūriu, žaisti žaidimus yra tarsi bandyti išvengti tikrovės. Žaidimų kūrėjams pasiseka, kai jų žaidimas įtraukia žaidėją ir atskiria nuo pasaulio, kuriame jis iš tikrųjų gyvena. Kai kurie žaidimai turi gilesnes reikšmes ir sukelia žaidėjams įvairius jausmus: liūdesį, pyktį, baimę ir pan. Žaidimų kūrėjai lankosi psichologų paskaitose, tam, kad geriau suprastų kaip paveikti žaidėjus.
- Socialiniai mokslai (angl. Social science). Kelių žaidėjų žaidimai, ypač didelį žaidėjų skaičių generuojantys žaidimai, išnaudoja žaidėjų socialinio gyvenimo aspektus ir pasinaudoja papildoma motyvacija, kurią žaidėjas jaučia žinodamas, kad žaidžia prieš kitą asmenį.

Žaidimas tampa priemone, leidžiančia žaidžiant vieniems žaidėjams sutikti kitus.

Taip, kaip žaidėjai bendrauja, yra mokslu paremtas aspektas. Žmonės linkę susidraugauti, tapti priešais, planuoti mūšius ir pan. Socialiniuose moksluose atliekami tyrimai yra susiję su masiniais kelių žaidėjų internetiniais žaidimais. Šių tyrimų pabaigoje gautos vertingos žinios yra pritaikomos naujų, kelių žaidėjų žaidimų kūrimui.

1.2 Kūrimo aplinka ir programavimo kalbos – 1 dalis

Kūrimo aplinką arba „integruotą kūrimo aplinką“ (angl. Integrated Development Environment) mes vadiname specialia programine įranga, kuri padeda kūrėjams kurti programinę įrangą. Papratoje integruotoje kūrimo aplinkoje yra redaktoriaus, kompiliatoriaus ir derintuvės funkcijos. Šiandieninėje, šiek tiek patobulintoje aplinkoje yra dar keletas kitų pagalbinių įrankių. Programinės įrangos tipams vis labiau skiriantis, keičiasi ir integruoto kūrimo aplinkos. Todėl yra daugybė integruoto kūrimo aplinkų, skirtų programinės įrangos kūrimui įvairiose srityse. Šioje aplinkoje asmenys, kuriantys žaidimus, yra vieni iš labiausiai specializuotų bei turinčių daug savitų, ypatingų savybių. Pagrindinis bet kurios programinės įrangos kūrimo elementas yra naudojama programavimo kalba. Kai kurios integruoto kūrimo aplinkos yra sukurtos viena kalba, o kitos gali būti sukonfigūruotos naudojant kelis kompiliatorius ar kalbas.

Žaidimų kūrimas šioje aplinkoje turi keletą specialių funkcijų, tačiau, ypač dvi iš jų yra labai svarbios. Pirmoji yra kelių įrankių, skirtų kurti skirtingas žaidimo dalis (grafiką, programavimą, garso takelius, tinklus ir t.t.), integravimas, o antroji funkcija žaidimų kūrime yra specializacija, kuomet dauguma žaidimo elementų yra iš anksto sukurti: sunkumo lygiai, žemėlapiai, veikėjai, įvairios animacijos ir t.t.).

Integruota kūrimo aplinka, skirta kurti žaidimus, turi leisti kūrėjui programuoti skirtingas žaidimo dalis, kurios turi skirtingas paskirtis. Kad atitiktų skirtingus reikalavimus, aplinka gali leisti kūrėjui naudoti skirtingas programavimo kalbas. Pavyzdžiui, internetinės pirmojo asmens šaudyklės kūrimui reikia naudoti trimatės grafikos programavimą, tačiau būtina atkreipti dėmesį ir į susijungimą per tinklą su nuotoliniu serveriu. Šiems dviem reikalavimams įvykdyti mums gali prireikti visiškai skirtingų programavimo kalbų. Be to, žaidime gali būti taisyklių, kas leidžiama ar draudžiama žaidėjui, arba kontrolinių taškų, kuriuos vartotojas turi aplankyti. Paprastai šiems reikalavimams išpildyti reikalinga aukšto lygio programavimo kalba, su kuria būtų galima lengvai surašyti taisykles. Taip pat atsižvelkite į tai, kad žaidime gali būti robotai, t.y. dirbtinio intelekto veikėjai, kurie žaidžia žaidimą savarankiškai. Kaip mes užprogramuotume šių robotų intelektą ir elgesį?

Daugelyje integruoto kūrimo aplinkų yra atskiros programavimo kalbos, taip bandoma vykdyti visus reikalavimus, bei tuo pačiu stengiamasi, kad tokia aplinka būtų lengva naudotis. Kitose aplinkose



leidžiamas bendrųjų programavimo kalbų naudojimas, kaip pavyzdžiui „C ++“. Naujausiose kūrimo aplinkose nėra ribojamas kitų kalbų naudojimas, nes yra įrankių ir būdų, kaip įterpti norimą šabloną ar parašytą kodą nesinaudojant „C++“ programavimo kalba, į „C++“ programą. Bet net ir bendrosios kalbos, tokios kaip „C++“, „Rust“, „Python“ ar „Lua“, kurios naudojamos žaidimų kūrime, yra išplėtos klasių ir bibliotekos funkcijomis, kurios yra įtrauktos į žaidimo variklį.

1.3 Kūrimo aplinka ir programavimo kalbos – 2 dalis

Įdomi bendros žaidimų kūrimo integruotoje kūrimo aplinkoje alternatyva yra žaidimo „redaktorių“ naudojimas. Kai kurie žaidimai platinami su juose įrašytą redagavimo funkcija. Tai yra priemonės, kurias kūrėjai naudojo kurdami žaidimo lygius, žemėlapius ir žaidimo etapus. Visas šias priemones ir sukūrė patys žaidimo kūrėjai.

Žaidėjas gali laisvai naudoti redagavimo funkciją, kad pats sukurtų žaidimo lygį su norimais atvaizdais ir garsais. Kai kurios redagavimo funkcijos leidžia paprasčiausiai atlikti tam tikrus numatytus žemėlapių pakeitimus, o dauguma funkcijų leidžia smarkiai pakeisti ar iš naujo sukurti žemėlapius bei lygius.

Realaus laiko strategija (angl. Real time strategy) ir pirmojo asmens šaudyklės (angl. First Person Shooters) – tai žaidimų žanrai, kuriuose paprastai yra redagavimo funkcijos. Redagavimo ar modifikavimo funkciją turinčius žaidimus sudaro šie elementai:

- Išteklių ar pranašumų, tokių kaip atvaizdai, garsai, animacijos ir kt., rinkinys, kuris sukuria pagrindinius žaidimo objektus.
- Žemėlapių, sunkumo lygių ar žaidimo etapų rinkinys, atitinkantis daugelį iššūkių, kuriuos žaidėjas turi įveikti.
- Programinė įranga, kuri iš aukščiau išvardytų elementų sukuria patį žaidimą. Šis elementas vadinamas žaidimų varikliu (angl. Game Engine)

Norint sukurti žaidimą, gali tekti pasirinkti vieną, jau esamą žaidimą su įrašyta redagavimo funkcija ir tada iš naujo suprojektuoti jo lygius bei siūlomus išteklius. Daugelyje, jei ne visose redagavimo funkcijose yra skriptų programavimo kalba, leidžianti tiksliai ir detaliam įgalinti žaidimo elementus (priešus, šaunamuosius ginklus, momentinį veikėjo sustiprinimą ir kt.)

Šis metodas buvo tęsiamas ir naudojamas toliau. Norėdami sukurti tikrai sudėtingus žaidimus, programuotojai naudoja du aspektus su skirtingomis pabaigomis. Iš pradžių jie kuria žaidimų variklį, kuris gali paleisti žaidimą, tačiau pats žaidimo turinys yra kuriamas atskirai. Žaidimo variklis yra sujungiamas su specializuota redagavimo priemone, naudojama generuoti tik šio žaidimo variklio lygius, grafiką, veikėjus ir pan. Tada, antrajame etape, kūrėjai užpildo visą žaidimo turinį, kurį sudaro žaidimo istorija su įvairiais lygiais. Kai žaidimas baigtas, visuomenei bus prieinami du produktai:

tikrasis žaidimas su pilnu turiniu ir kartu integruotu žaidimo varikliu, bei redagavimo funkcija, kurią naudos entuziastai, mėgstantys kurti naują turinį.

Šie žaidimai gali būti parduodami atskirai, todėl kai kurie kūrėjai naudoja žaidimo variklį ir redagavimo funkciją, kad sukurtų visai kitą žaidimą, o ne tik pradinio žaidimo modifikaciją ar pratęsimą. Kartais žaidimo variklis pristatomas ir parduodamas trečiosioms šalims ar individualiems asmenims dar prieš baigiant bet kurį žaidimo, su tuo varikliu, kūrimo procesą. Tačiau tai vis tiek negali būti laikomi „bendrosios paskirties“ žaidimų kūrimo įrankiais, nes jie nuo pat pradžių yra nukreipti į vieną tikslų žaidimo tipą.

Tai skiriasi nuo žaidimo su tiesiog įrašyta redagavimo funkcija.

Čia yra naudojimui tinkamas bei populiarėjantis žaidimų variklių [rinkinys](#)

1.4 Kūrimo aplinka ir programavimo kalbos – 3 dalis

Galiausiai, sekdami internetinių programinės įrangos parduotuvių, tokių kaip „Valve's Steam“, „Google Play“, „Sony PlayStation Store“, steigimo tendenciją, kai kurios integruotos kūrimo aplinkos palengvino programinės įrangos pristatymą ir platinimą. Integruota kūrimo aplinka gali naudotis kūrėjo paskyra bet kurioje iš šių skaitmeninių parduotuvių tam, kad pristatytų žaidimą.

Apibendrinant šį skyrių, turime apsvarstyti ar patogiu naudoti kokį nors bendrąjį žaidimų kūrimo įrankį, kuris yra orientuotas į paprastumą ir pritaikytas pradedantiesiems, neturintiems ankstesnių žinių apie programavimą. Iš didelio turimos programinės įrangos pasirinkimo, „GameMaker“ programinė įranga yra vienas iš tinkamiausių įrankių šiam projektui. Tai patikrintas bei naudotas sprendimas su didele vartotojų bendruomene ir keliais sėkmingais produktais.

2 skyrius – "Game Maker" žaidimo variklio platforma

2.1 „Game Maker“ platformos ypatybės. Žaidimų tipai.

[„GameMaker Studio“](#) (šiam vadove „GameMaker“) yra integruota žaidimų kūrimo aplinka. Pradžioje šios platformos paskirtis buvo leisti pradedantiesiems kurti paprastus žaidimus neturint daug žinių apie programavimą. Siekdami suteikti šią galimybę, platformos kūrėjai pasiūlė keletą technologijų, leidžiančių sudaryti vizualų kompiuterio programos atitikmenį. Prireikus, kūrėjai galėtų rašyti kodą ir naudoti programavimo kalbą, panašią į „C“, tačiau su kai kuriomis savybėmis paimtomis iš „JavaScript“ programavimo kalbos. Dėl šios priežasties programavimo kalba yra pakankamai lengva, kad ja galėtų naudotis pradedantieji.

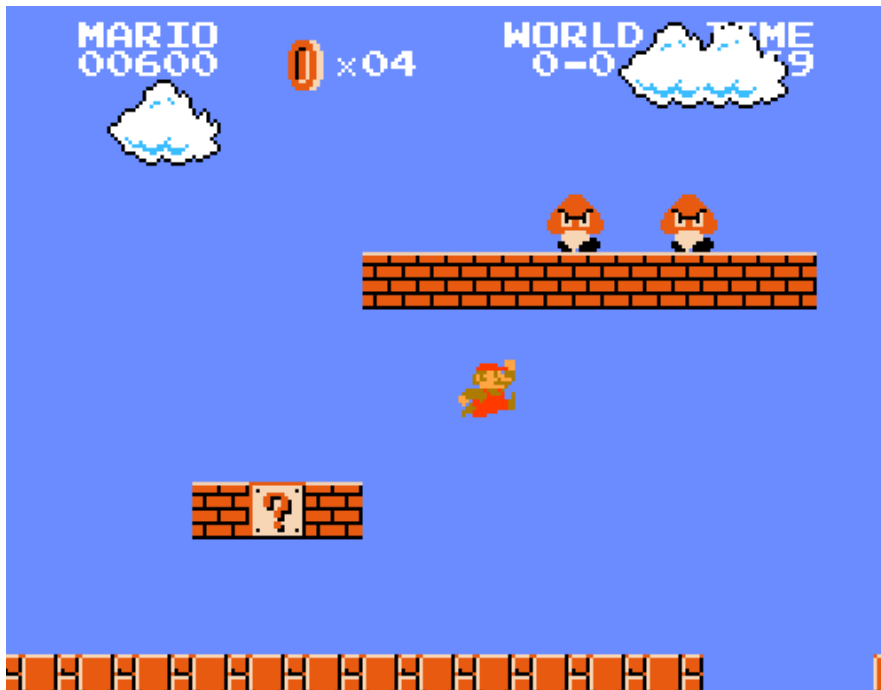
Pagrindinės savybės:

- Integravimas. Visus arba daugumą žaidimo kūrimo etapų galima atlikti naudojant įtaisytus įrankius. Dažniausiai tam nereikia išorinės programinės įrangos pagalbos.
- Platforma turi savo kalbą („Game Maker Language“ arba „GML“), kuri nėra įprasta programuotojams, įpratusiems naudoti „C“, „Javascript“ ir panašias kalbas. Ši programavimo kalba yra integruota į žaidimo variklį ir kūrimo aplinką.
- Paprastiems žaidimams programavimo nereikia. „Game Maker“ leidžia kūrėjui naudoti nutempimo (angl. Drag and Drop) programavimo būdą kaip alternatyvą „GML“ programavimo kalbos naudojimui.
- Žaidimai gali būti sukurti daugumai operacinių sistemų ir žaidimų platformų. Taip pat skaitmeninio platinimo platformų integracija yra svarbi.
- Išskirtinai tinka tik kelių rūšių žaidimams, tačiau gali būti pritaikyta daugeliui žanrų, atlikus papildomus modifikavimo darbus.

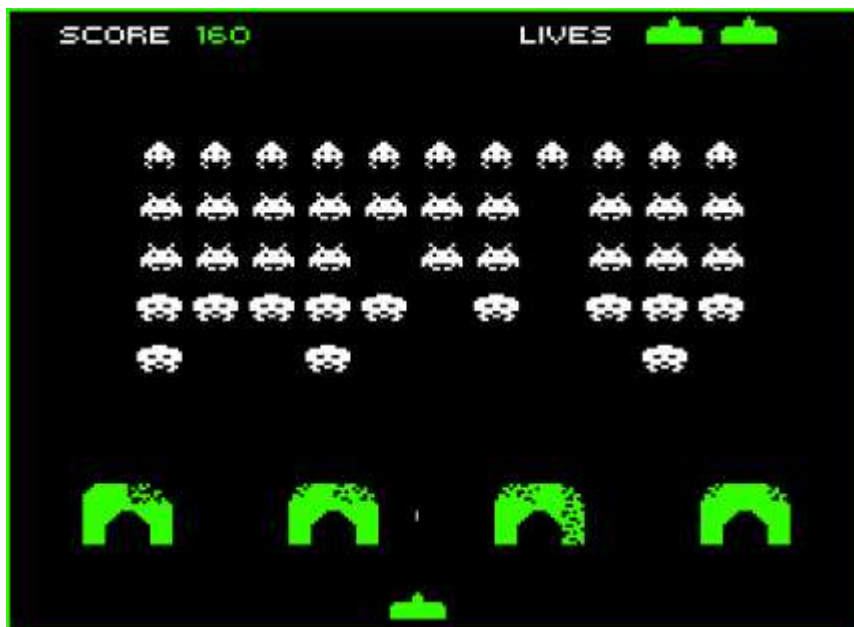
Žaidimų kūrimas „Game Maker“ platformoje pasižymi paprastumu ir naudingumu, ypač, kuriant šaudykles (angl. Shoot 'em up) bei platforminius žaidimus. Taip pat galima kurti ir kitus žaidimų tipus.

- Platforminiuose žaidimuose veikėjai juda dviejų dimensijų pasaulyje rinkdami objektus, vengdami kliūčių ir naikindami priešus. Žinomiausias tokio žaidimo pavyzdys yra „Mario Bros“





- Šaudyklės (angl. Shoot 'em up). Šiame žaidime žaidėjas nuolat keliauja per žaidimo žemėlapij, tai jis daro arba skridamas, arba eidamas. Taip pat jis gali pasiekti skirtingus žaidimo vietas už žemėlapijo ribų. Priešai atsiranda žaidėjui judant į priekį. Tuomet jis turi išvengti priešų arba juos nužudyti. Tokių žaidimų pavyzdys yra „Space invaders“



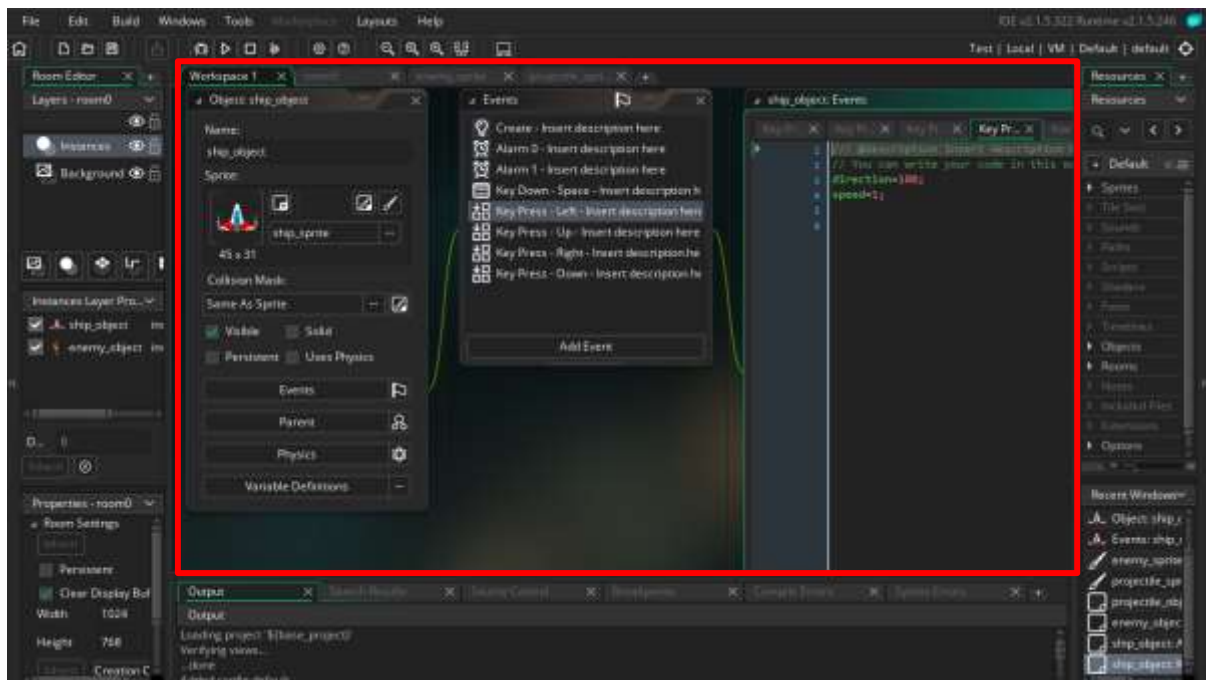
2.2 Kūrimo aplinka

„Game Maker“ yra universali kūrimo aplinka. Į šią aplinką įeina visos reikalingos priemonės žaidimams kurti bei juos išbandyti, nereikalaujant išorinių įrankių, prietaisų ar specialių nustatymų. Vartotojo sąsaja yra tiesiogiai susieta su centriniu „rėmu“, kuriame galima paskirstyti kelias darbo vietas sritys. Darbo vietos yra skydeliai arba plokštės, kuriose žaidimo elementus ir išteklius galima patalpinti ir tvarkyti kūrimo metu. Tokiu būdu galima labiau suasmeninti ir padalinti kūrimą. Darbo vietos pačios rodydys skirtingo turinio vidinius langus, pradedant nuo bendrųjų objektų su ypatybėmis, įvykių sąrašais ir „GML“ programavimo kalbos kodu. Kiekvienas, vienas su kitu susietas elementų langas, yra sujungtas tam tikra linija.

Be to, aplink centrinę darbo vietos sritį yra antriniai skydeliai. Jei reikia, daugumą iš jų galima sutraukti, tam, kad centrinė zona galėtų išsiplėsti. Šiuose skydeliuose vaizduojama:

- Medžio formos schema su visais žaidimo duomenų komponentais. Ši schema yra nuolatinė ir keičiasi tik sukuriant ar ištrinant tam tikrus žaidimo duomenų komponentus.
- Valdiklių, skirtų bet kokio pasirinkto komponento ypatybei pakeisti, sąrašas. Šie skydeliai keičiasi, kai kūrėjas sutelkia dėmesį į kitą elementą.
- Išvesties langas žaidimo kompiliavimo ir vykdymo rezultatams pamatyti.

Kitas langas gali būti rodomas atskirai, virš kitų, arba kaip skirtukas centrinėje srityje, įgalinus tam tikrą, specifinį įrankį. Pavyzdžiui, vaizdo redagavimo funkcija naudojama retai, tačiau, redaguojant vaizdo medžiagą, toks skirtukas atsidaro centrinėje srityje.



„Game Maker“ platforma siūlo du žaidimo kūrimo būdus:

- Nutempimas (angl. Drag and Drop). Šis režimas yra skirtas naujokams, nemėgstantiems ar nenorintiems koduoti bei tiems, kurie tik apsimeta jog kuria žaidimus. „Game Maker“ platforma leidžia nustatyti žaidimo logiką naudojant vaizdinio programavimo įrankį. Šiame režime kūrėjas nutempia bei atskiria susietus ir sugrupuotus loginius elementus, kad galėtų detalizuoti bet kurio elemento elgseną.
- „Game Maker Language“ programavimo kalbos projektai. Šis režimas yra numatytasis ir standartinis, o žaidimo logika ir bet kurio elemento elgsena apibrėžiama programavimo kodu "Game Maker Language" programavimo kalba.

Šiame vadove mes naudosime „Game Maker Language“ variantą, nes pavyzdžių sudėtingumas nebus didelis ir mes neatliksime sudėtingo programavimo, tačiau tai darydami turėsime daugiau galimybių ateityje, kuomet darbas jau bus atliktas.

2.3 Sprukliai, kambariai, daiktai ir susidūrimai.

Žaidimai kompiuterio ekrane pateikia dinamiškai besikeičiančius elementus. Tai, ką žaidėjas mato, yra tiesiog „daiktai“, judantys teritorijos ar „pasaulio“ viduje, ir dažnai šie „daiktai“ dūžta, susiduria ar susikerta su kitais dalykais.

„Game Maker“, taip pat kūrėjai ir žaidėjai nurodo konkrečius skirtingų vaizdinių elementų pavadinimus.

Teritorija arba „pasaulis“, kuriame vyksta žaidimas, „GameMaker“ platformoje yra vadinamas kambariu (angl. Room).

Animuoti vaizdai yra vadinami sprukliais (angl. Sprites). Statiniai ar neanimuojami vaizdai yra tik dekoratyviniai ir gali būti vadinami fonu (angl. Background).

Žaidimo elementams, kurie turi tam tikrą elgesį ar bet kokią įtaką žaidimui, priskiriamas objekto (angl. Objects) pavadinimas.

Susidūrimas (angl. Collision) yra labiausiai paplitęs veiksmas tarp dviejų objektų.

Šie elementai yra pagrindinės „GameMaker“ platformos funkcijos.

[Sprukliai, kambariai, daiktai ir susidūrimai](#) – išsamus vadovas, skirtas nagrinėti pagrindinius žaidimo elementus.



2.4 Įvykiai ir veiksmai

Žaidimas, kai jis yra paleistas ar kažkas jį žaidžia, yra dinamiška sistema. Tai kažkas, kas dažnai keičiasi žvelgiant iš žaidėjo perspektyvos. Šie nuolatiniai žaidimo pokyčiai iš tikrųjų yra mažų, labai dažnų pasikeitimų, suvokiamų kaip didesnio, sudėtingesnio įvykių posūkio, rezultatas. Svarbiausias veiksnys nustatant žaidimo raidą yra pagrindinių pokyčių supratimas ir jų komponentų nustatymas. Šie komponentai vadinami įvykiais ir veiksmais.

Norėdami geriau suprasti įvykius ir veiksmus, pirmiausia turime atsižvelgti į tai, kad kompiuterinis žaidimas yra programa, kuri veikia nuolat. Kartais ši programa vadinama žaidimo varikliu (angl. Game Engine). Ši programa stebi klaviatūrą, pelę ir kitus valdiklius. Programa fiksuoja laiką, žino bet kurio žaidimo objekto poziciją ir kryptį. Todėl žaidimo variklis pirmasis sužino, jog kažkas nutiko: ar buvo paspaustas klaviatūros klavišas, ar įvyko delsa, ar ekrane susikirto du objektai. „GameMaker“ platformoje įvykiais vadinami šie dalykai, kurie atsiranda netikėtai ir teoriškai gali turėti įtakos žaidimui. Daugybė tokių įvykių atsiranda dėl žaidėjo sąveikos su kompiuteriu. Kai kurie kiti įvykiai atsiranda dėl žaidimo evoliucijos.

Žaidimo variklis kontroliuoja žaidimą ir bet kurią akimirką gali pakeisti jo būseną. Jis gali sujudinti ar perkelti objektą, padaryti tam tikros žalos, sukurti naują objektą ir pan. Šie pakeitimai vyksta taip, kaip numatyta programuotojo atliktame programavime, kuriame žaidimo variklis yra tik programos vykdytojas. „GameMaker“ platformoje veiksmais vadinami pakeitimai, kuriuos nurodo kūrėjas. Svarbiausia, kad kiekvienas veiksmas įvyktų prasmingai. Net ir atsitiktinai nustatyti veiksmai yra susieti su įvykiais, kuriuose yra atsitiktinių komponentų. Kūrėjo tikslas yra numatyti ir nustatyti, kurie veiksmai ir dėl kokių įvykių jie įvyksta. Pirmiausia išsamiai išanalizuosime įvykius, o tada pamatysime, kaip su jais susieti veiksmai

„Game Maker“ platformoje galima rasti ilgą, iš anksto nustatytų įvykių sąrašą. Vis tik, žaidimo kūrėjas gali laisvai imtis bet kokių veiksmų, reaguodamas į pasirinktą įvykį (atkreipkite dėmesį į internetinius šaltinius).

[Įvykiai ir veiksmai](#) – išsamus vadovas, skirtas pagrindiniams „GameMaker“ platformos rodomiems įvykiams.

2.5 Sąlyginiai veiksmai

Kartais, kuomet įvykiai nėra susieti, veiksmus reikia atlikti sudėtingomis sąlygomis. Tokiais atvejais kūrėjas turi parašyti specialią sąlygą, pagal kurią veiksmas bus atliktas. Žaidimo variklis taip pat turi

būti suprogramuotas taip, kad tikrintų sąlygą kiekvienu įmanomu momentu.

Kai įvykdoma sąlyga, veiksmas atliekamas. Kūrėjui tai suteikia didelį lankstumą.

Tačiau sąlyginiai veiksmai yra tie veiksmai, kuriuos reikia susieti su įvykiu (taip veikia „GameMaker“ platforma).

Jei sąlygą reikia tikrinti kiekvieną kartą, tada įvykis yra natūralus.

Veiksmas tiesiog yra programos dalis, kuri papildomai tikrina sąlygą.

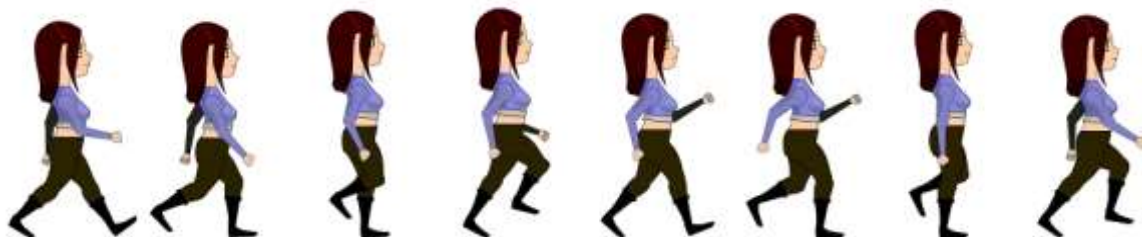
[Sąlyginiai veiksmai](#) – dokumentas, kuriame aiškiai nurodyta kaip veikia sąlyginiai veiksmai

2.6 Animacija ir sudėtingi susidūrimai

Animacija žaidimuose yra nepaprastai svarbi dalis, leidžianti žaidėjui patirti visus žaidimo teikiamus malonumus. Nuo vaizdo žaidimų kūrimo pradžios, programuotojai ir menininkai daug dirbo, kad sukurtų kuo tikroviškesnius ir patrauklesnius žaidimus. Norint tai pasiekti, reikia atkartoti daugelio objektų judesius realiame gyvenime. Pavyzdžiui, vairuodami mašiną žvelgdami iš šono ar iš viršaus turėtumėte ne tik stebėti, kaip automobilis juda keliu, bet ir pamatyti kaip sukasi ratai, kaip veržiasi dūmai iš duslintuvo ar kaip vibruoja padangos nelygiame kelyje. Yra keletas būdų, kaip sukurti animacijos efektus žaidimuose ir kai kurie iš jų yra gerai integruoti į „Game Maker“ platformą. Atvaizdai yra skirti fizinių objektų, kurie dažnai juda ir susiduria su kitais objektais, vaizdavimui ar pateikimui.

Labiausiai paplitusi technika animacijoms kurti yra panašių vaizdų, kurie rodomi cikliškai, sugrupavimas į vieną rinkinį. Jei vaizdų keitimosi sparta yra tinkama, žaidėjas suvoks vaizdus kaip animaciją. Lieka sunkiausia užduotis – sudaryti atvaizdų rinkinį, kuris sukurs ir sugeneruos geriausios kokybės animaciją.

Mes žinome, kad sprukliai (angl. Sprite) turi keletą atvaizdų pogrupių. Paimkite, pavyzdžiui, eilę po atvaizdų ruožu. Sprukliai pavaizduotų, kad viena mergina vienu metu greitai pereina į kitą eilę, sukurdami efektą, pagal kurį atrodo, kad mergaitė iš tikrųjų vaikšto judindama kojas (vaizdas keistųsi kartu su padėtimi).



<https://opengameart.org/content/girl-walking-side>

„Game maker“ platforma yra labai galinga, kuomet reikia kurti ir naudoti spruklius. Ji gali nuskaityti failą, kaip aprašyta aukščiau, ir gana lengvai padalyti jį į skirtingus pavienius vaizdus, tokiu būdu sukurdamą spruklius beveik automatiškai.

Ji gali tvarkyti vaizdus, kurių atvaizdų pogrupiai yra sulygiuoti kituose maketuose (pavyzdžiui, tam tikrų vienodo tipo elementų lentelėje – matricoje)

Spruklių naudojimas „Game maker“ platformoje yra gana paprastas ir naudingas. Tačiau, jų kūrimas ir „piešimas“, sudėtingas. Kai kuriose svetainėse yra iš anksto pateiktų spruklių, kuriais naudojantis galite pradėti kurti žaidimus. Viena iš šių svetainių yra <http://opengameart.org>.

Sprukliai taip pat nustato objekto formą. Priklausomai nuo objektų dydžio erdvėje, susidūrimai tarp jų gali neįvykti nepaisant tinkamų koordinatų ir išdėstymo. Pavyzdžiui, pagalvokite apie mažus objektus, judančius link kur kas didesnių. Dažniausiai susidūrimų aptikimo sistema turi užkirsti kelią objektų užklojimui ar prasiskverbimui į kitus. Todėl susidūrimo sistemoje atkreipiamas dėmesys ne tik į objektų padėtį, bet ir į formą ar matmenis.

Kuriant spruklius, taip pat naudinga nustatyti spruklių „susidūrimo kaukę“ (angl. collision mask), kuri bus taikoma objektams, rodantiems spruklius. „Susidūrimo kaukė“ (angl. collision mask) yra galimai savavališkos formos sritis. Paprastai dėl efektyvumo priežasčių „susidūrimo kaukė“ yra stačiakampis, kurį prireikus galima pasukti, kad geriau atitiktų spruklių formą. Apskritimus ir elipses taip pat galima nustatyti kaip „susidūrimo kaukę“, tačiau jie apdorojami lėčiau. Galiausiai, labai ypatingais atvejais, „susidūrimo kaukę“ galima nustatyti taip, kad ji atitiktų visus spruklio atvaizdų pogrupius. Visgi, tokiais atvejais veikimo greitis yra labai mažas. Atminkite, kad susidūrimai turi būti tikrinami 60 kartų per sekundę kiekvienam objektui priartėjant prie kito.

„Game maker“ platformoje paprastus susidūrimus kūrėjai apdoroja dviem būdais:

- Naudojant susidūrimo įvykį, kuriam kūrėjas gali parašyti veiksmų kodą, kuris reguliuos tai, ką reikia padaryti. Tai yra tinkamiausias būdas, kai susidūrimai su tam tikru objektu nėra įprasti (pavyzdžiui, kosminiu laivu, skraidančiu kosmose)
- Tikrinant veiksmus, susijusius proceso įvykiu. Tai daryti rekomenduojama, jei tam tikras objektas dažnai susiduria su kitu. Jei nenaudosite susidūrimo įvykio kūrimo, sutaupysite šiek tiek laiko, nes tą patį galima pasiekti ir su žingsnio įvykiu, tiksliai patikrinant kodą.

„Game Maker“ platforma siūlo žaidimų variklį, kuris suteikia visiškai naujus būdus valdyti judesius, susidūrimus ir fizinę objektų sąveiką. Kai naudojama ši funkcija, susidūrimus visiškai valdo žaidimo variklis, gana stipriai supaprastindamas kūrimą. Vienintelis trūkumas yra tas, kad kūrėjas turi nurodyti kiekvieno objekto ir erdvės fizines savybes ir patikslinti kai kuriuos parametrus, norėdamas gauti norimą rezultatą.

[Sprukliai](#) – svetainė, teikianti prieigą prie spruklių saugyklos.

2.7 Vaizdo žaidimų eksportavimas ir leidyba

Žaidimo eksportavimas reiškia „Game Maker“ platformos teikiamą galimybę sukurti programą, galinčia veikti su tam tikra aparatine ir programine įranga. Leidyba yra žaidimo, kurį gali įsigyti visuomenė, prieinamumas. Tradiciškai abi veiklas vykdo visiškai skirtingos įmonės. Tačiau, „Game maker“ platformos kūrėjai bando susidoroti su abejomis problemomis, palengvindami žaidimo pristatymo ir populiarinimo procesą.

Pagrindinė „Game Maker“ funkcija yra teikiama galimybė eksportuoti žaidimą į daugybę skirtingų platformų nekeičiant pačio žaidimo. Pagrindiniai adresatai (platformos ar sistemos, kurioms žaidimų kūrėjas gali sukurti žaidimą) yra „Linux“ („Ubuntu“), „Mac OS X“, „Android“, „iOS“, „fireTV“, „Android TV“, „Windows“, „Microsoft UWP“, HTML5, „PlayStation 4“ ir „Xbox One“. Kiekvienam iš šių adresatų būtinos specialios licenzijos. „Game Maker“ turi galimybę generuoti paketus, kuriuos galima įdiegti į bet kurią iš aukščiau išvardytų sistemų. Bet norint, kad žaidimas pasiektų žaidėjus, dar reikia pasistengti. Žaidimą kaip ir knygas reikia visur skelbti, reklamuoti ir pristatinti.

Šiame skaitmeniniame amžiuje, kur viskas yra sujungta į tinklą, leidyba išgyvena tikrą revoliuciją. Žaidimų formatas keičiasi į skaitmeninį turinį. Pagrindiniai leidėjai dabar centralizuoja šių skaitmeninių žaidimų leidybą. Kūrėjams yra trumpinamas žaidimo paskelbimo proceso kelias. Žaidimai gali būti prieinami žaidėjams iškart po jų sukūrimo. Žaidimų kūrėjams reikia tik leidybos platformų prenumeratos.

„Game Maker“ gali skelbti žaidimus „Steam“ platformoje. Naudodamiesi "Steam" platformoje esančiais žaidimais, tuo pačiu galite naudotis ir kai kuriomis „Steam“ platformos teikiamomis paslaugomis, tokiomis kaip pasiekimų lentelė, mokamas atsisiunčiamas turinys, debesų saugykla ir kt.

„Game Maker“ siūlo skirtingas licencijas, pagrįstas tikslinėmis platformomis, taip pat prenumeratomis ar integracija su skirtingomis leidybos platformomis.

3 skyrius – Praktiniai pavyzdžiai

3.1 – Įvadas į praktinius pavyzdžius

Šiame vadove mes parodysime įžanginę dalį "Game maker" platformoje, dirbdami su dviem praktiniais pavyzdžiais.

Abu pavyzdžiai yra labai paprasti ir nereikalauja jokių ankstesnių žinių apie programavimą.

Abu pavyzdžiai papildo vienas kitą. Taip pat, antrasis pavyzdys yra šiek tiek kitoks, tam, kad būtų galima išnagrinėti trūkstamus pirmojo pavyzdžio elementus.

3.2 - Sukurkite šaudyklę (angl. Shoot'em up)

Pirmojo praktinio pavyzdžio tikslas yra sukurti šaudyklę, kurioje žaidėjas kontroliuos erdvėlaivį.

Šis laivas gali šaudyti sviediniais, kad sunaikintų priešus, kurie savo ruožtu yra mažesni erdvėlaiviai.

Priešų laivai pasirodys ekrano viršuje ir judės žemyn bei šaudys ten, kur paprastai yra žaidėjas.



Galite remtis šiuo [gudu](#) tam, kad suprastumėte procesą, būtiną kuriant paprastą šaudyklę.

3.3 – Platforminio žaidimo kūrimas

Antrame pavyzdyje siekiama sukurti platforminį žaidimą. Jame dauguma tų pačių punktų, kurie buvo nagrinėjami pirmame pavyzdyje, tačiau dabar šie punktai bus išnagrinėti kur kas detaliau.

Kai kurios panašios savybės, kuriomis dalijasi abu žaidimai, išspręstos skirtingai, suteikiant skaitytojui platesnį žaidimo programavimo būdų pasirinkimą.

Platforminiai žaidimai daugelį metų buvo labai populiarūs. Daugeliu atvejų tai yra tiesiniai (angl. linear) žaidimai, kuriems tik reikalingi įgūdžiai judinti veikėją 2-dimensijų pasaulyje, šaudyti, šokinėti ir pan.

Tokiems žaidimams iškart kuriamas minimalus, lengviausias žaidžiamo lygis be papildomų grafinių elementų, kuriuos pridėsime vėliau.

Žaidėjas galės judinti ir priversti pašokti vieną veikėją tik su trimis klavišais: rodyklėmis į kairę ir į dešinę bei tarpu (angl. Space).

Galite remtis šiuo [gиду](#) tam, kad suprastumėte procesą, būtiną kuriant paprastą platforminį žaidimą.

