

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS, FIZINIŲ IR BIOMEDICINOS MOKSLŲ FAKULTETAS
INFORMATIKOS IR MATEMATIKOS KATEDRA

Donatas Skrodenis

Informatikos specialybės IV kurso nuolatinių studijų studentas

Šaudyklės žaidimo kūrimas
Shooter Game Development

BAKALAURO DARBAS

Darbo vadovas: A. Margienė
Recenzentas: dr. K. Žilinskas

Šiauliai
2017 m.

*„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros
sąrašą pateiktus informacinius šaltinius bei savo tyrimų duomenis“*

Darbo autoriaus _____

(vardas, pavardė, parašas)

Darbo tikslai ir uždaviniai

Darbo tikslas:

- Sukurti šaudyklės žanro žaidimą

Uždaviniai:

- Išnagrinėti galimus šaudyklių tipus ir pasirinkti vieną.
- Sugalvoti šaudyklės žaidimo idėją;
- Apžvelgti populiariausius žaidimų variklius ir išsirinkti tinkamiausią šaudyklės žaidimo kūrimui.
- Išanalizuoti dirbtinio intelekto elgesio medžių taikymo galimybės šaudyklės žanro kompiuteriniame žaidime ir pritaikyti juos kuriamame žaidime įvairioms situacijoms modeliuoti.
- Sukurti žaidimą.
- Atlikti žaidimo testavimą

Turinys

1.	Analitinė dalis	7
1.1.	Temos analizė.....	7
1.1.2.	Šaudyklės žanro žaidimų tipai	7
1.1.2.1.	„Shoot em up“ šaudyklės.	7
1.1.2.1.1.	“Fixed” šaudyklės.	7
1.1.2.1.2.	„Top-Down“	7
1.1.2.3.	Pirmo asmens šaudyklės.....	8
1.1.2.4.	Trečio asmens šaudyklės.....	8
1.1.2.5.	Taktinės šaudyklės.	8
1.1.2.6.	„Herojų“ šaudyklės.	8
1.2.	Dirbtinio intelekto panaudojimas	9
1.2.1.	Dirbtinis intelektas šaudyklės žanro žaidimuose	9
1.2.3.	Dirbtinio intelekto elgesio medžių naudojimas	10
1.3.	Žaidimo idėja.....	12
1.4.	„Top-Down“ šaudyklės žaidimo apimtis.	12
2.	Projektinė dalis.....	13
2.1.	Darbo įrankių pasirinkimas	13
2.2.	Pradinis projektas	14
2.2.1.	Programinė pradinio projekto dalis.....	14
2.2.2.	Pagrindinis veikėjas ir ginklai.....	16
2.2.3.	Dirbtinis intelektas	16
2.2.4.	Grafinio dizaino pradinė projekto dalis	18
2.3.	Numatytas darbų planas	18
3.	Realizacinė dalis	19
3.1.	Darbo eigos grafas.....	19
3.2.	Realizuotas projektas.....	20
3.2.1.	Klasių diagrama	20
3.2.2.	Žaidimo būsenos diagrama	20
3.2.3.	Vartotojo veiksmų diagrama.....	21
3.2.4.	Dirbtinio intelekto elgesio medis	22
3.3.	Žaidimo realizacijos būsenos aprašymas	23
4.	Problemos ir jų sprendimai	26
5.	Išvados.....	27

Anotacija	28
Summary	28
Literatūros sąrašas.....	29
Priedai	30
CD turinys	30
Testavimo protokolas	30

Įvadas

Pastaraisiais dešimtmečiais kompiuteriniai žaidimai tapo mūsų kultūros dalimi. Kompiuteriniai žaidimai yra daugumos žmonių laisvo laiko praleidimo būdas. Nuo pat jų atsiradimo, žaidimų populiarumas nuolat auga. Didėjant žanrų skaičiui ir žaidimų kiekiui, žaidėjų auditorija nuolat plečiasi. Žaidimai jau nėra vien laisvalaikio praleidimo būdas, nes specifiniai žaidimai naudojami kaip efektyvios mokomosios priemonės. Tobulėjantys žaidimai labai stipriai prisidėjo prie techninės įrangos tobulėjimo ir jos inovacijų kūrimo.[12]

Vienas iš seniausių kompiuterinių žaidimų žanrų yra šaudyklės. Viena iš pirmųjų šaudyklių buvo „Spacewar!“ kurią sukūrė Steven Russell. Žaidimo tikslas buvo sunaikinti kito žaidėjo erdvėlaivį manevruojant žvaigždės gravitacijos lauke ir stengtis išvengti susidūrimo su ja. [5, 13].

Šiame darbe nagrinėjami šaudyklių tipai ir jų bruožai, analizuojami būdai, kurie naudojami kurti dirbtinį intelektą šaudyklės žanro žaidimuose, aprašytas šaudyklės žaidimo kūrimas (kas buvo sukurta ir panaudota žaidime).

Darbo tikslas:

- Sukurti šaudyklės žanro žaidimą

Uždaviniai:

- Išnagrinėti galimus šaudyklių tipus ir pasirinkti vieną.
- Sugalvoti šaudyklės žaidimo idėją;
- Apžvelgti populiariausius žaidimų variklius ir išsirinkti tinkamiausią šaudyklės žaidimo kūrimui.
- Išanalizuoti dirbtinio intelekto elgesio medžių taikymo galimybės šaudyklės žanro kompiuteriniame žaidime ir pritaikyti juos kuriamame žaidime įvairioms situacijoms modeliuoti.
- Sukurti žaidimą.
- Atlikti žaidimo testavimą

1. Analitinė dalis

1.1. Temos analizė

Šiame darbe bus analizuojamas šaudyklės žaidimo kūrimas personaliniam kompiuteriui. Bus išnagrinėti, kokie yra šaudyklių žanrai, sugalvota idėja šaudyklės žaidimui, bus išanalizuotas dirbtinio intelekto naudojimas šaudyklės žanro žaidimuose, bus apžvelgti, kuriant šaudyklę, panaudoti įrankiai.

1.1.2. Šaudyklės žanro žaidimų tipai

Šaudyklės žanro kompiuteriniai žaidimai yra veiksmo žaidimų požanris, kuriame žaidėjas turi pasikliauti savo greita reakcija bei mąstymu. Šis žanras turi daug požanrių, bet pagrindinis dėmesys yra nukreipiamas į veiksmus, atliekamus žaidėjo valdomiems veikėjams, kurie naudojami įvairiais ginklais. Dažniausiai yra naudojami šaunamieji ginklai. Pagrindinis šaudyklės tikslas yra šaudyti į dirbtinio intelekto valdomus priešus ir pereiti misiją ar lygį.

Šaudyklės žanro požanriai:

1.1.2.1. „Shoot 'em up“ šaudyklės. Šaudyklės požanris, kuriame žaidėjas gali vaikščioti į keturias puses: aukštyn, žemyn, kairėn ir dešinėn, ir šaudyti tik tiesiai. Dažniausiai šiame žanre žaidėjas valdo erdvėlaivį ar kitą transporto priemonę ir kovoja su dideliais priešų kiekiais, renka resursus (amunicija, gyvybės, taškus) ir tuo pat metu stengiasi išvengti priešų atakų. Šio žanro žaidimai reikalauja žaidėjo greitos reakcijos. [5, 13]

Shoot 'em up žanro šaudyklės yra skirstomos į kategorijas pagal dizaino elementus, ypač kameros poziciją ekrane ir veikėjo judėjimą.

1.1.2.1.1. „Fixed“ šaudyklės. Veikėjas gali judėti tik viena ašimi (daugiausiai horizontaliai), priešai puola viena kryptimi (daugiausiai nuo ekrano viršaus) ir scena telpa į ekraną. „Space invaders“ vienas iš geriausiai žinomų šio žanro žaidimų.

1.1.2.1.2. „Top-Down“ šaudyklės. Pagrindinis šio žanro bruožas, kamera kurios pozicija yra virš žaidėjo veikėjo. Dažniausiai šio žanro žaidimai pačia žaidimo mechanika nesiskiria nuo tradicinių Shoot 'em up žaidimų. Žaidėjas yra vienas ir kovoja su dideliu priešų skaičiumi ir renka taškus. Pastaraisiais metais

top-down šaudyklėse pradėtos naudoti žaidimų mechanikos iš kitų šaudyklės žanrų. Žaidžiant lygis progresuoja, nes pagrindinis tikslas nėra rinkti kuo daugiau taškų, o stengtis nueiti iki kito lygio galo ar įvykdyti tam tikrą užduotį. Oponentų kiekis yra sumažintas, bet jų dirbtinis intelektas yra labiau rafinuotas ir atlieka daugiau veiksmų: patruliavimas, slėpimasis už priedangos, skirtingų ginklų naudojimas. Žaidimai, kaip „Alien Swarm“, leidžia žaisti keturiems žaidėjams vienoje komandoje, taip suteikiant šiam žanrui kooperacinį žaidimo būdą.

1.1.2.3. Pirmo asmens šaudyklės. Vienas iš populiariausių šaudyklės žanro požanrių. Šis šaudyklės žanras skiriasi nuo kitų šaudyklės žanro žaidimų tuo, kad žaidėjas nevaldo jokios transporto priemonės, o visas veiksmas matomas iš pirmo asmens perspektyvos. Tokiu būdu žaidėjui suteikiamas jausmas, kad jis viską mato pro valdomo veikėjo akis ir leidžia žaidėjui pasinerti į žaidimą ir įsivaizduoti, kad jis yra žaidimo herojus. Šis žanras tapo labai populiarus dvidešimto amžiaus 10-ajame dešimtmetyje, kai „ID Software“ 1993 metais išleido DooM. DooM ir kitos ankstyvosios pirmo asmens šaudyklės turi panašumų į „Shoot ‘em up“ žanro žaidimus, nes jos reikalauja žaidėjo greitos reakcijos ir pagrindinis tikslas yra kovoti prieš didelį priešų skaičių.[6, 7, 8]

1.1.2.4. Trečio asmens šaudyklės. Šios šaudyklės daug nesiskiria nuo pirmo asmens šaudyklių. Vienas iš pagrindinių skirtumų, kad kameros pozicija yra už veikėjo nugaros šiek tiek aukščiau peties.[7]

1.1.2.5. Taktinės šaudyklės. Šios šaudyklės žanro žaidimai gali būti tiek pirmo asmens, tiek trečio asmens. Žaidimuose yra simuliuojama realistiška kovojimo mechanika, todėl taktikos naudojimas ir atsargumas yra svarbesnis nei greita reakcija, kuri būdinga kituose šaudyklės žanruose. Taktinėse šaudyklėse pagrindinis dėmesys yra skiriamas realistiškumui, todėl veikėjas ar kompiuterio valdomas priešas gali būti užmušamas vienu šūviu, tad žaidėjas turi būti atsargus kovos eigoje. Dažnai šio žanro žaidimuose žaidėjas savo komandoje turi būrį kompiuterio valdomų veikėjų, kuriems jis gali duoti nurodymus, kaip užimti tam tikrą poziciją, pulti specifinius priešus ir t.t. Tokios taktinės šaudyklės, kaip „Operation Flashpoint“, ir „Arma“ serijos žaidimai simuliuoja didelio masto karinį konfliktą, vykstantį didelėje zonoje, kurioje veikėjas vykdo misijas ir gali naudotis įvairia karine technika.[10]

1.1.2.6. „Herojų“ šaudyklės. Šios šaudyklės daugiausiai yra skirtos žaisti internete su kitais žaidėjais, šie žaidimai gali būti tiek pirmo asmens, tiek trečio, taip pat ir top-down

žanro. Žaidėjai yra paleidžiami į areną ir padalinami į dvi ar daugiau komandų. Žaidėjai prieš mačo pradžią gali pasirinkti herojus, kurie turi skirtingus unikalius atributus, gebėjimus ir ginklus. Šios šaudyklės skatina žaidėjus dirbti komandoje, todėl žaidėjai renkasi savo veikėjus, kurie efektyviausiai padėtų savo komandai. Pastaraisiais metais „herojų“ šaudyklės, kaip „Team Fortress 2“, „OverWatch“, turi didelį šansą būti žaidžiamos Esport turnyruose.[6,8]

1.2. Dirbtinio intelekto panaudojimas

1.2.1. Dirbtinis intelektas šaudyklės žanro žaidimuose

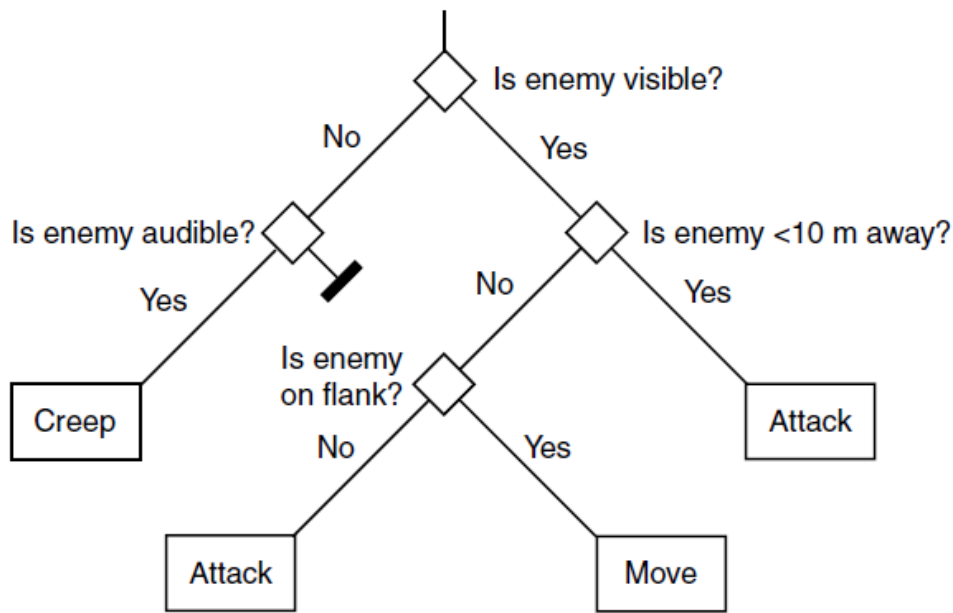
Žaidimo dirbtinis intelektas – tai programos kodas, kuris priverčia kompiuterio valdomus veikėjus priimti protingus sprendimus, esant tam tikroms aplinkybėms žaidimo aplinkoje. Kitaip tariant, iš galimų sprendimų aibės pasirenkamas labiausiai tinkantis esančiai situacijai sprendimas.

Dauguma senesnių šaudyklių turi gana paprastą dirbtinio intelekto logiką, kuri atlieka tik keletą funkcijų (pamačius priešą eiti į jo pusę, šauti į priešą), nes šiuose žaidimuose svarbiau yra priešų kiekis nei priešų logika. Modernios šaudyklės naudoja pažangesnius dirbtinio intelekto logikos algoritmus, kurie suteikia daugiau funkcionalumo. Šie algoritmai yra naudojami siekiant, kad dirbtinio intelekto valdomas veikėjas turėtų galimybę pasirinkti, kaip elgtis skirtingose situacijose. Taip suteikiamas veikėjui tikroviškumo jausmas, o žaidėjui didesnis iššūkis, nes dirbtinis intelektas reaguos vis kitaip. Tai atliekama panaudojant sprendimų medžius arba elgesio medžius.[6]

1.2.2. Dirbtinio intelekto sprendimų medžių naudojimas

Sprendimų medis yra greitas, lengvai įgyvendinamas ir lengvai suprantamas. Jis yra vienas iš paprasčiausių dirbtinio intelekto logikos algoritmo kūrimo būdų. Sprendimų medžiai naudojami valdyti kompiuterio valdomiems veikėjams, taip pat ir animacijų valdymui.

Sprendimo medis susideda iš sujungtų „sprendimo“ mazgų. Pradinis sprendimo mazgas vadinasi „ROOT“ mazgu. Kiekvienam sprendimui, pradedant nuo „ROOT“ mazgo, yra pasirenkamas vienas iš kelių galimų sprendimų.

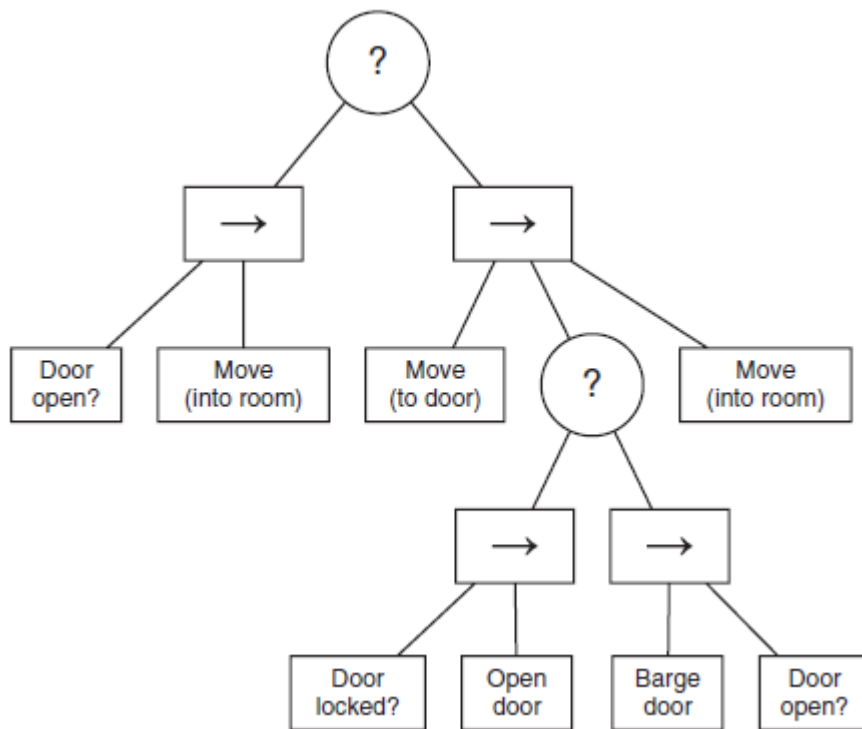


Dažniausiai sprendimo medis naudojamas kaip greitas ir paprastas sprendimo priėmimo mechanizmas, todėl veikėjas pasirenka sprendimą pagal esamą žaidimo būseną, vietoj to, kad būtų užkoduojamas tam tikras elgesys, kas būdinga elgesio medžiuose. Sprendimo medžio algoritmas atlieka sprendimo pasirinkimą prie sprendimo mazgo ir veikia tol, kol sprendimų pasirinkimo procesas nebeturi daugiau sprendimų. Dauguma sprendimo medžių mazgų turi gana paprastus sprendimus, dažniausiai jie susidaro tik iš dviejų galimų atšakų.[4]

1.2.3. Dirbtinio intelekto elgesio medžių naudojimas

Elgesio medis veikia naudodamasis mazgais. Mazguose yra užrašomas elgesys, kurį dirbtinis intelektas turės atlikti. Elgesio medis susideda iš šakninio mazgo, į kurį yra jungiami mazgai, kurie bus atsakingi už elgesio pasirinkimą. Į pasirinkimo mazgus yra jungiami elgesio mazgai, kuriuose yra aprašoma, kaip dirbtinis intelektas elgsis. Šie mazgai yra skaitomi pagal svarbumą iš kairės į dešinę. Jei mazgo nurodymai atitinka jam duotą sąlygą, tuomet mazgas nurodomas kaip „veikiantis“. Jei mazgo nurodymai neatitinka sąlygos, jis nurodomas kaip „neįvykdytas“. Tuomet sąlyga yra grąžinama atgal į pasirinkimo mazgą ir pasiunčiama kitam, pagal eilę esančiam, elgesio mazgui.[3,4]

2 pav. Elgesio medis



„F.E.A.R“ laikomas vienu geriausiu, dirbtinį intelektą turinčiu, šaudyklės žanro žaidimu. Šioje šaudyklėje dirbtinis intelektas gali naudoti taktikas, kurios suteikia jam tikroviškumo jausmą, turi daug pasirinkimų, kuriuos gali naudoti įvairiose situacijose: jis gali pradėti šaudyti į žaidėjo poziciją, taip nukreipdamas žaidėjo dėmesį ir leisdamas kitiems priešams prieiti arčiau žaidėjo. Dažnai priešai slėpsis vietose, kur yra blogas apšvietimas. Jei žaidėjas pradės jį pulti, atsitrauks ir mėgins įspėti kitus priešus, naudosis aplinkoje esančiais objektais kaip priedanga.

Dauguma Top-Down šaudyklių turi gana paprastą elgesio medį, kuriame yra tik keletas elgesio pasirinkimų. Šaudyklėje „Alien Shooter“ dirbtinis intelektas turi tris elgesio būsenas:

- stovėti ir laukti, kol žaidėjas įeis į matomumo zoną;
- gaudyti priešą po sceną, kol jis yra matomumo zonoje;
- pulti žaidėją, kuomet jis yra puolimo atstume.

Tokio žanro šaudyklėse žaidėjas kovoja prieš didelį priešų skaičių, todėl nereikalingas elgesio medis, turintis daug būsenų.

Taktinės ir dauguma trečio asmens šaudyklių naudoja priedangos sistemą, ir dažniausiai dirbtinio intelekto elgesys yra kuriamas aplink šią sistemą. Šio tipo šaudyklėse dirbtinio intelekto elgesys, kuomet jis naudojasi priedanga, gali priklausyti nuo naudojamų ginklų. Ginklų efektyvumas gali priklausyti nuo atstumo, todėl dirbtinio intelekto elgesys priklausys nuo atstumo iki žaidėjo.

1.3. Žaidimo idėja.

Žaidimo kūrimui buvo pasirinkta „Top-Down“ žanro šaudyklė. Dauguma šio žanro šaudyklių turi labai primityvią dirbtinio intelekto logiką. Priešas, pamatęs žaidėją, visada eina link jo pozicijos ir puola. Dažniausiai priešų išvaizda skiriasi viena nuo kitos, bet jie visados atlieka tas pačias funkcijas. Beveik visados Top-Down šaudyklės yra 2D, nes kameros pozicija yra 90 laipsnio kampų.

Kuriamos „Top-Down“ šaudyklės kameros pozicija bus tarp 80 ir 70 laipsnio kampų, taip suteikiant žaidimui 3D vaizdą. Žaidėjas kovodamas su priešais galės naudotis scenoje esančiais objektais, kaip priedanga ir dviejų tipų ginklais. Bus naudojamos fizikos (kai žaidėjas nušaus priešą, priešo modelis vietoj to, kad iškarto pranyktų, jis suklups ant žemės). Tipiniuose „top-down“ šaudyklės žaidimuose visas veiksmas vyksta arenoje, todėl kuriamo žaidimo lygių dizainas bus panašus į tokių žaidimų kaip „DooM“ lygius, kai visas lygis susidaro iš kelių kambarių, kurie yra sujungti durimis. Žaidėjo pagrindinis tikslas yra ne rinkti taškus, o stengtis pereiti lygį nenumirus. Kartais žaidėjas turės tyrinėti aplinką, kad rastų mygtuką, kuris atidarys duris į išėjimą iš lygio. Tai yra būdinga klasikinėse pirmo asmens šaudyklėse, kaip „DooM“ ir „Quake“. Šaudyklėje bus sukuriami skirtingi priešų tipai, kurie turės skirtingus ginklus ir skirtingą elgesį. Keli priešų tipai naudosis priedanga puolant žaidėją.

1.4. „Top-Down“ šaudyklės žaidimo apimtis.

Šiame darbe pasiryžta suprojektuoti ir realizuoti pilnavertę Top-down žanro šaudyklę. Galutinis darbo projektas turi turėti visą žaidime numatytą funkcionalumą, veikiančią vartotojo sąsają ir leisti žaidėjui pereiti lygį be jokių trukdžių.

2. Projektinė dalis

2.1. Darbo įrankių pasirinkimas

Žaidimo variklis – Žaidimui sukurti buvo renkamas iš dviejų populiariausių žaidimo variklių: Unreal Engine 4 ir Unity.

Unreal Engine 4 – daugiaplatforminis žaidimų kūrimo variklis, sukurtas kompanijos „Epic Games“, išleistas 1998 metais su žaidimu „Unreal“. Variklis daugiausiai naudojamas šaudyklės žanro žaidimams kurti, tačiau jis sėkmingai buvo pritaikytas ir kituose žanruose: rolių, lengtynių ir kituose žaidimuose. Variklis palaiko daug platformų, kurioms galima kurti žaidimus: konsolėms, įvairioms kompiuterių operacinėms sistemoms, mobiliems įrenginiams.[2]

Unity - daugiaplatforminis žaidimų variklis, sukurtas kompanijos „Unity Technologies“ ir naudojamas kurti žaidimams personaliniams kompiuteriams, konsolėms, mobiliems įrenginiams ir internetinėms svetainėms. Pirmą kartą paskelbtas 2005 metais per Apple konferencijoje „Worldwide Developers Conference“ tik OS X operacinei sistemai. Šiuo metu Unity palaiko 27 platformas. [1]

Paanalizavus abu variklius buvo nuspręsta naudoti Unreal Engine 4. Šis variklis yra labiau pritaikytas kurti šaudyklės žanro žaidimus, nei Unity. Unreal Engine 4 turi „pradėjimo paketą“, kuriame yra veikėjo modelis, ginklo modelis, 64-ios veikėjo animacijos, taip pat medžiagos scenos modeliavimui, garso ir vaizdo efektai bei scenarijų pavyzdžiai. Kuriant naują projektą, Unreal Engine 4 leidžia vartotojui pasirinkti žaidimo šabloną. Jis gali būti pirmo asmens šaudyklės, Top-Down, puzlės, lenktyninio, kuriuose automatiškai sukuriama valdymo scenarijai. Unreal Engine 4 kurti scenarijus galima naudojantis C++ programavimo kalba ir integruotu vizualinio programavimo įrankiu „Blueprint“.

Microsoft Visual Studio 2015 – Microsoft Visual Studio yra “Microsoft” sukurta IDE. Ji naudojama kurti programas Microsoft Windows operacinei sistemai, internetiniams puslapiams, internetinėms aplikacijoms ir paslaugoms. Visual Studio palaiko beveik visas programavimo kalbas, išskyrus Java(F#) . Ji buvo palaikoma senesnėse versijose.

Visual Studio 2015 yra naujausia VS. Ją nemokamai gali naudoti individualūs kūrėjai, atviro kodo programų kūrėjai ir mažos komandos, kurios kuria programas.[11]

BluePrint – Unreal Engine 4 esantis vizualinis scenarijų kūrimo įrankis. Naudojantis šiuo įrankiu nereikia ranka rašyti kodo, nes yra naudojamas *node-based* metodas. Šis įrankis

puikiai tinka greitam įvairių žaidimo mechanikų sukūrimui, nuo tokių paprastų scenarijų, kaip veikėjo judėjimas, iki dirbtinio intelekto mąstymo, tačiau šis įrankis turi limitus.[1]

Žaidimo scenarijų kūrimui bus naudojamas BluePrint. Su Blueprint greičiau yra kuriami scenarijai, galima įjungti scenarijaus simuliaciją ir vizualiai matyti, kaip kas veikia scenarijuje. Taip pat yra išleista daug knygų, kuriose rašoma, kaip dirbti su Blueprint, kaip kurti tam tikrus scenarijus ir ką atlieka kiekvienas mazgas (*angl. node*).

Objektų kūrimui buvo naudojama 3D kūrimo aplinka „Autodesk 3ds MAX“. Pirmoji aplinkos versija buvo išleista 1990 metais kompanijos “Autodesk Media And Entertainment”. Aplinka skirta kurti 3D modelius ir animacijas. Ji turi lanksčią įskiepių architektūrą. Naujausios 3Ds MAX versijos palaiko šešėliavimą (*angl. shaders*), dalelių sistemą, *normal map* kūrimą ir vaizdavimą, apšvietimą, pritaikomą vartotojo sąsają ir savo scenarijų kūrimo kalbą.[10]

2.2. Pradinis projektas

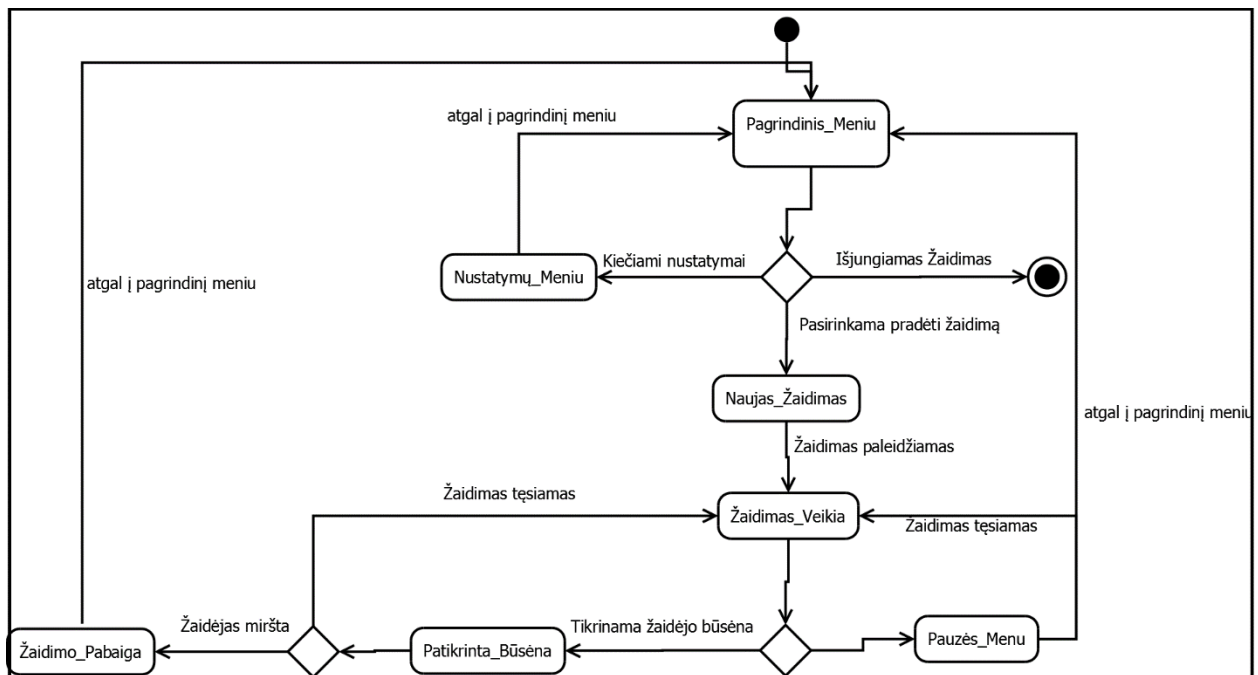
2.2.1. Programinė pradinio projekto dalis

Darbe numatyta suprojektuoti ir realizuoti „Top-Down“ žanro šaudyklę.

- Pagrindinės žaidimo būsenos:
 - pagrindinio meniu būsena;
 - nustatymų būsena;
 - žaidimo veikimo būsena:
 - žaidėjo animacijų būsena;
 - dirbtinio intelekto animacijų būsena.
 - žaidimo pauzės meniu būsena;
 - žaidimo pabaigos („Game Over“) būsena.

Kiekviena žaidimo būsena tiesiogiai lemia, kokias funkcijas atliks aktyvinta būsena, kokie žaidimo elementai bus naudojami ekrane ir koks valdymas bus naudojamas.

3 pav. Pradinio projekto žaidimo būsenų diagrama



- Sukuriami ir atvaizduojami
 - Pagrindiniai valdymo elementai
 - Mygtukai
 - Tekstų dėžės
 - Slankiosios
 - Tekstų spalvos ir stiliai
 - Paveikslėliai
 - Pelė
 - Žaidimo eigos elementai
 - Tekstų dėžės
 - Progreso juostos
 - Pranešimų dėžė
 - Pelės taikiklis

Pagrindiniai valdymo elementai bus naudojami pagrindinio meniu, nustatymų, žaidimo pauzės ir žaidimo pabaigos būsenose. Jie leis žaidėjui valdyti langų keitimą ir keisti vaizdo bei valdymo nustatymus.

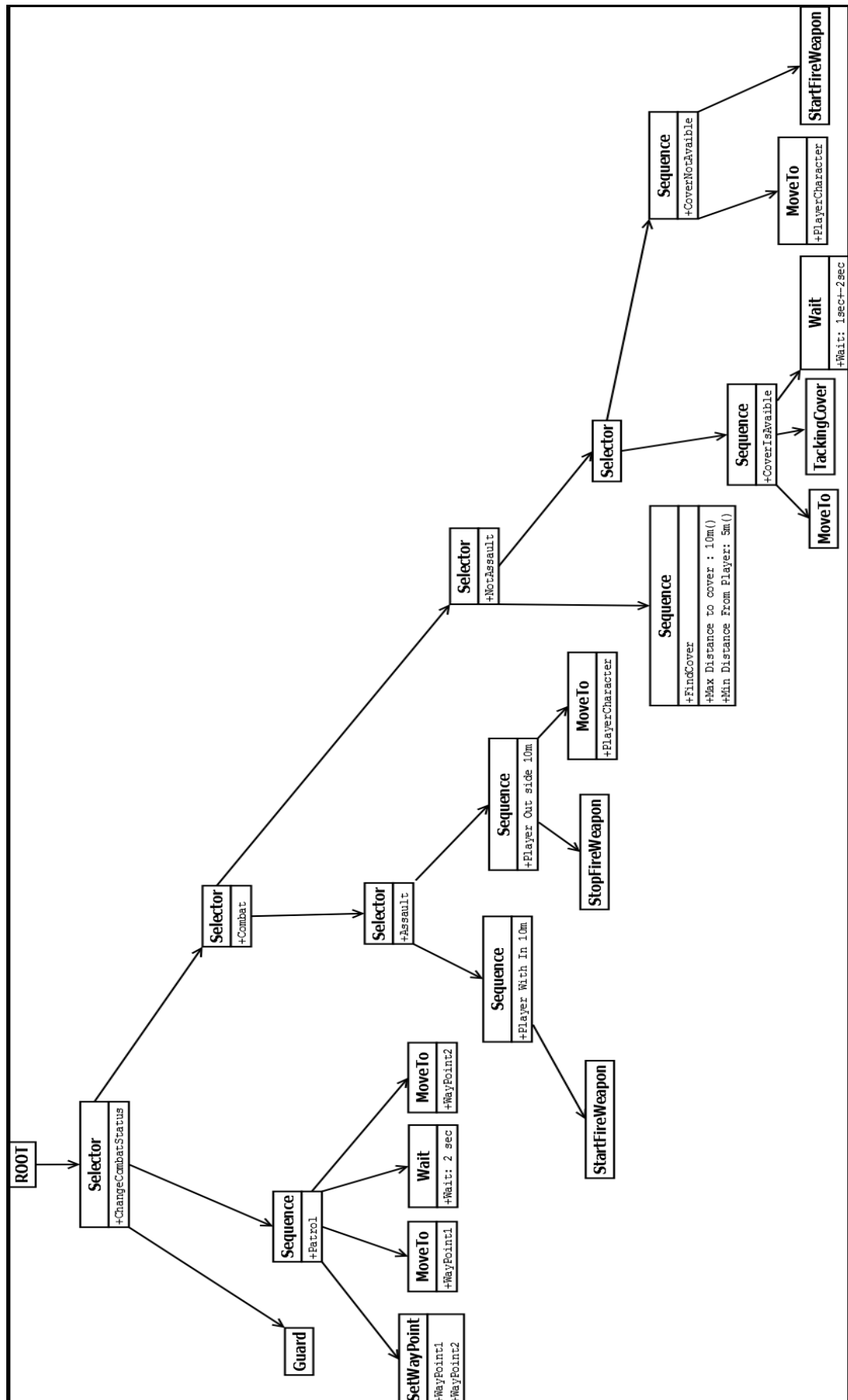
Žaidimo eigos elementai veiks tik žaidimo veikimo būsenoje. Jais siekiama suteikti žaidėjui informacijos apie tai, kas vyksta žaidime ir kaip jam sekasi.

2.2.2. Pagrindinis veikėjas ir ginklai

Žaidime planuojama sukurti veikėją, kuris turės kelias judėjimo būsenas (ėjimas, bėgimas ir gebėjimas atsitūpti). Žaidėjas galės keisti judėjimo būseną bet kuriuo metu, bus kuriami kelių tipų ginklai, kurie turės skirtingus amunicijos tipus, skirtingus garsus ir efektus.

2.2.3. Dirbtinis intelektas

Žaidime bus realizuojamas dirbtinio intelekto elgesio medis nes, jis labiau tiks kuriamam žaidimui, taip pat Unreal Engine 4 turi integruotą dirbtinio intelekto elgesio medžio kūrimo įrankį. Bus sukurti keli priešų tipai, kurie turės skirtingus ginklus ir skirtingą elgesį.



2.2.4. Grafinio dizaino pradinė projekto dalis

„Top-Down“ šaudyklei vizualizuoti yra sukuriami žaidimo langai pagal atitinkančias žaidimo būsenas.

- Pagrindinio meniu langas
- Nustatymų langas
 - Grafikos nustatymų langas
 - Pelės valdymo langas
- Žaidimo veikimo langas
- Žaidimo pauzės langas
- Žaidimo pabaigos „Game Over“ langas.

2.3. Numatytas darbų planas

1 lentelė. Numatytų darbų ir jų atlikimo planas

	Vasaris				Kovas				Balandis				Gegužė				
	Savaitių skaičius																
Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	

Darbų sąrašas pagal numerius	
1	Išnagrinėti galimus šaudyklių tipus ir pasirinkti vieną
2	Sugalvoti šaudyklės žaidimo idėją
3	Įrankių pasirinkimas ir jų analizė
4	Dirbtinio intelekto elgesio medžių analizė
5	Kuriamas pradinis žaidimo projektas
6	Kuriami žaidime naudojami objektai
7	Realizuojamas žaidimas
8	Darbo aprašymo rengimas

3. Realizacinė dalis

3.1. Darbo eigos grafas

2 lentelė. Atliktų darbų planas

	Vasaris Savaitių skaičius				Kovas				Balandis				Gegužė				
Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	

Darbų sąrašas pagal numerius	
1	Išnagrinėti galimus šaudyklių tipus ir pasirinkti vieną
2	Sugalvoti šaudyklės žaidimo idėją
3	Įrankių pasirinkimas ir jų analizė
4	Dirbtinio intelekto elgesio medžių analizė
5	Kuriamas pradinis žaidimo projektas
6	Kuriami žaidime naudojami objektai
7	Realizuojamas žaidimas
8	Darbo aprašymo rengimas

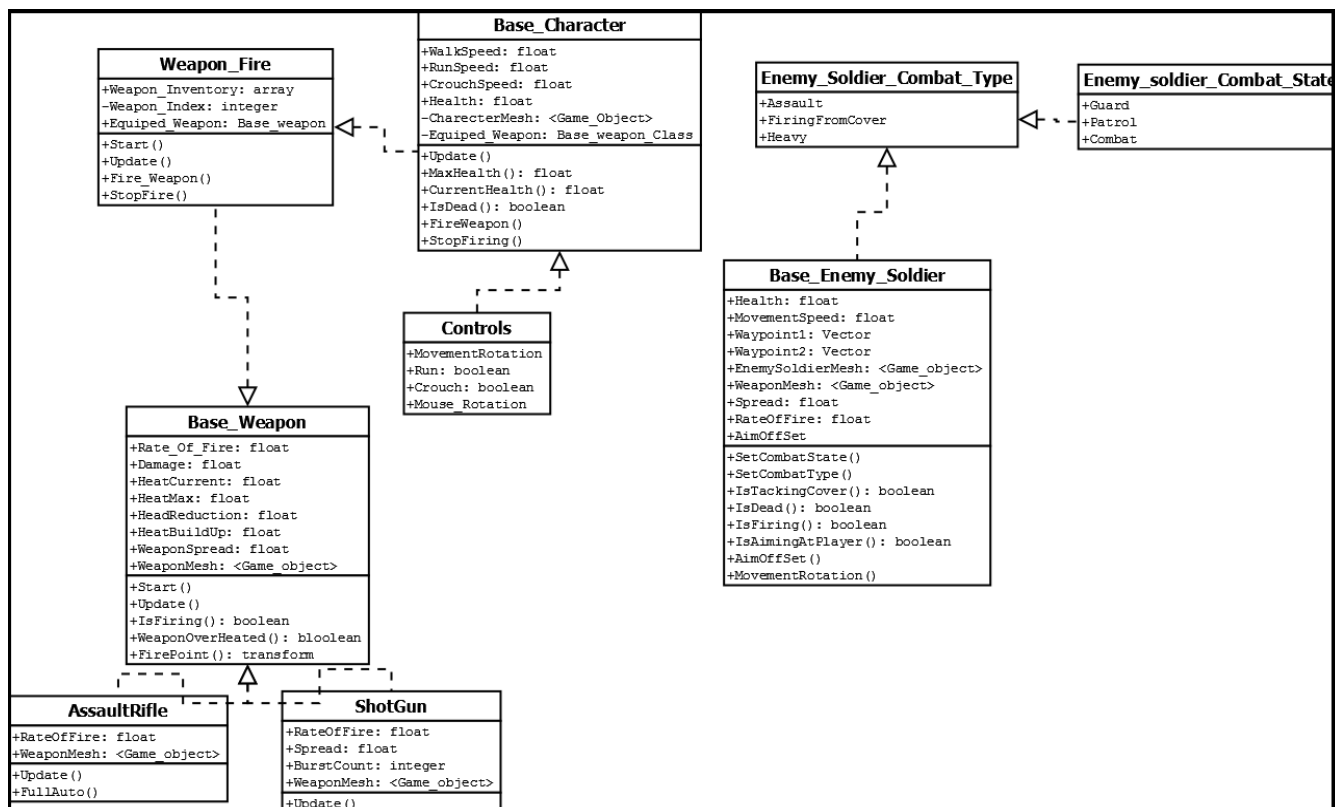
Darbų grafikas keitėsi, daugiau laiko buvo skirta pačio žaidimo kūrimui, nes buvo iškilusios problemos su dirbtinio intelekto animacijų veikimu, žaidėjo veikėjo turimų ginklų veikimu, taip pat su nustatymų meniu veikimu.

3.2. Realizuotas projektas.

3.2.1. Klasių diagrama

Sukurta galutinio projekto veikėjų ir ginklų klasių diagrama. Šioje diagramoje nurodytos svarbiausios dalys, sudarančios veikėjus ir ginklus.

5 pav. Veikėjų ir ginklų klasių diagrama



3.2.2. Žaidimo būsenos diagrama

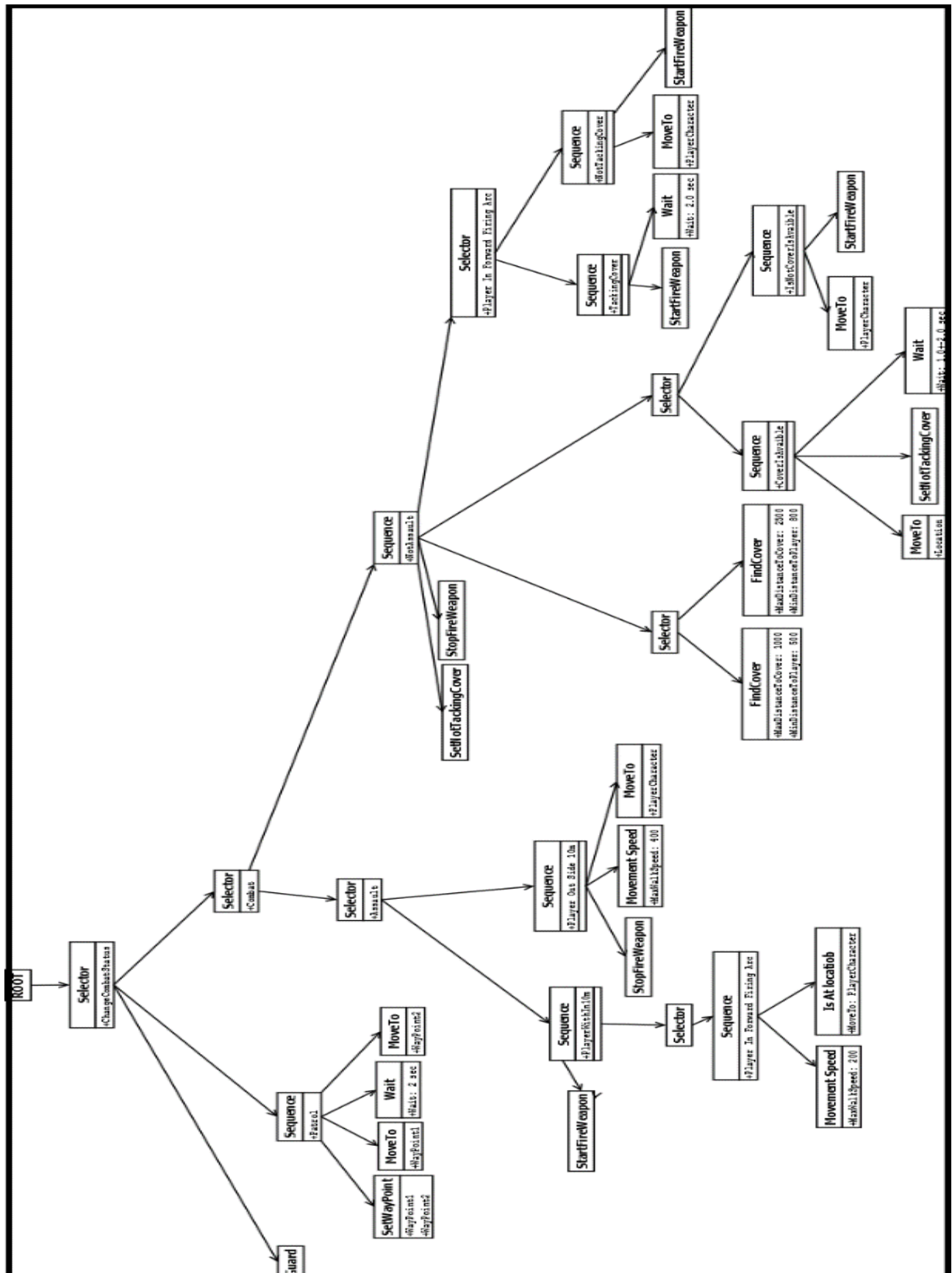
Galutinio projekto būsenos diagrama. Šioje diagramoje nurodyta, koku eiliškumu keičiasi būsenos, keičiantis būsenoms, keičiasi ir žaidime esantys langai.

6 pav. Galutinio žaidimo būsenų diagrama

3.2.4. Dirbtinio intelekto elgesio medis

Čia pavaizduotas dirbtinio intelekto elgesio medis, kuris yra naudojamas žaidime.

8 pav. Galutinio projekto dirbtinio intelekto elgesio medis



3.3. Žaidimo realizacijos būsenos aprašymas

Darbas atliktas kompiuteriu, kurio techniniai parametrai:

- Operacinė sistema: Windows 7 Ultimate 64bit
- Procesorius: AMD FX-8300
- Operatyvioji atmintis: 10GB
- Vaizdo plokštė: Geforce GTX 950

Sėkmingai buvo sukurtas „Top-Down“ šaudyklės žanro žaidimas. Buvo įvykdytas minėtas reikalavimas, kad žaidime turi veikti visos realizuotos funkcijos ir, kad žaidėjas galėtų pereiti demonstracinį lygį be jokių trukdžių.

Sukurti veikėjai ir jų animacijos. Veikėjų kūrimui buvo naudojamas Unreal Engine 4 esantis nemokamas veikėjo modelis, pagrindiniam veikėjui buvo sukurtas scenarijus, kuriame buvo užrašoma gyvybių skaičiavimo funkcija, kurios pagalba būtų skaičiuojama, kiek žaidėjas praranda gyvybių, jei priešas į jį pataiko ir kas atsitinka, kuomet žaidėjo gyvybės pasiekia 0. Taip pat šiame scenarijuje buvo sukurta funkcija, kuri veikėjo modeliui priskiria ginklo modelį, nes kitaip žaidimo metu veikėjo modelis judėtų be ginklo modelio rankose. Pagrindiniam veikėjui iš Unreal Engine 4 animacijų paketo buvo priskirtos penkios animacijos: stovėjimo, ėjimo, bėgimo, atsitūpimo ir mirties. Priešams buvo naudojama daugiau animacijų, nes reikėjo atskirai nurodyti, kada priešas naudojasi priedanga ir kada jis pasitraukia nuo priedangos.

Sukurtas ginklų scenarijus. Šiame scenarijuje yra aprašomas ginklo veikimas žaidimo metu. Buvo sukurta šaudymo funkcija. Šioje funkcijoje yra aprašyta pagrindinė šovimo informacija: iš kurios ginklo vietos bus paleistas šūvis, kuomet žaidėjas keičia veikėjo žiūrėjimo kryptį su pele, šūvio paleidimo vieta taip pat turi būti keičiama pagal veikėjo žiūrėjimo kryptį. Realizuojant projektą vietoj ginklų, kurie naudoja amuniciją, buvo pasirinkta naudoti ginklų keitimo sistemą. Buvo sukurta funkcija, kurioje nurodoma kiek ginklas galės iššauti šūvių prieš pasiekdamas kritinę karščio ribą, taip pat sukurta funkcija, kurioje nurodoma, koku greičiu karštis bus mažinamas.

Sukurta ginklų keitimo sistema. Šios sistemos dėka žaidėjas gali pasikeisti rankose turimą ginklą iš ginklų inventoriaus. Buvo sukurtas masyvas, kuriame yra saugomi ginklo

indeksai. Kiekvieną kartą žaidėjui pasinaudojus ginklo keitimo mygtuku, sistema tikrina, kuris ginklo indeksas yra naudojamas ir pakeičia kitu, masyve esančiu ginklo indeksu.

Sukurtas dirbtinio intelekto elgesio medis - Realizuojant dirbtinio intelekto sprendimų medį buvo padaryta pokyčių. Pradiniame projekte buvo planuota kurti dviejų tipų priešus - „Assault“ tipo ir „Cover“ tipo, realizuojant buvo pridėtas trečias tipas „Heavy“. „Assault“. Šio tipo priešai turi paprastą elgesį, pamatę žaidėją eina link jo, kai atstumas tarp jo ir žaidėjo bus 4 metrai, priešas pradės pulti. Jei žaidėjas atsitrauks nuo priešo daugiau nei 10 metrų, priešas nustos pulti ir pradės eiti link žaidėjo, kol atstumas tarp jų vėl bus 4 metrai. „Cover“ tipo priešai turi sudėtingesnę elgesį - pamatęs žaidėją priešas pradės 10 metrų spinduliu ieškoti priedangos, kuri būtų nutolusi nuo žaidėjo ne mažiau kaip per 5 metrus. Jei dirbtinis intelektas randa priedangą naudodamasis atsitūpimo animacija, jis pasislepia už priedangos, o elgesio medis tikrina, ar dirbtinis intelektas naudoja priedangą. Tuo metu atsitiktinai kas vieną arba dvi sekundes jis atsistos nuo priedangos ir pradės šaudyti į žaidėją. Jei dirbtinis intelektas neranda priedangos, jis paprasčiausiai puola žaidėją 5 metrų atstumu. „Heavy“ tipo priešai daug nesiskiria nuo „Cover“ tipo priešų. Pagrindiniai skirtumai: „Heavy“ tipo priešų ginklai daro daugiau žalos taškų, lėčiau šauda, „Heavy“ tipo priešai ieško priedangos 15 metrų spinduliu ir minimalus atstumas iki žaidėjo turi būti 8 metrai.

Sukurti valdymo elementai – šiais elementais yra valdomi visi meniu langai, mygtukams yra pritaikytos specifinės animacijos (užvedus ir nuvedus pelės žymeklį, paspaudus ir atspaudus mygtuką su pelės žymekliu) ir funkcijos. Mygtukų pavadinimams ir kitai tekstiniai informacijai pateikti buvo naudojami iš interneto parsisiųsti šrifto stiliai.

Pagrindinis meniu langas – šiame lange vartotojui pateikiami 3 mygtukai:

- Žaidimo paleidimo mygtukas („New Game“)
- Nustatymų mygtukas („Settings“)
- Žaidimo išjungimo mygtukas („Quit“)

Nustatymų langas – Šiame lange vartotojas gali nusistatyti žaidimo vaizdo nustatymus ir žaidimo valdymą.

- Vaizdo nustatymuose galima pasikeisti rezoliucijos dydį, „Anti-Aliasing“, „Post Processing“, šešėlių, tekstūrų ir efektų kokybę. Norint išsaugoti pasirinktus nustatymus, reikia paspausti mygtuką „Apply“.

- Valdymo nustatymuose galima pasikeisti, koks pelės jautrumas bus žaidimo metu. Šie nustatymai išsisaugo savaime.

Žaidimo sesijos langas – šiame lange yra naudojami žaidimo eigos elementai. Specifiniai žaidimo eigos elementai pateikia svarbią informaciją. Žaidėjo gyvybės yra pateikiamos naudojantis progreso juosta, kuri yra užpildyta raudona spalva, ir prie jos panaudojus teksto dėžę yra užrašytas žodis „Health“, kad vartotojas žinotų, ką ši juosta reiškia. Kuomet žaidėjas patiria žalą, progreso juosta pradeda mažėti į kairę pusę. Taip pat progreso juosta naudojama matuoti ginklo karščiui. Kai žaidėjas naudojasi ginklu, kyla karščio juosta. Kai ši juosta pasiekia maksimalią ribą, virš žaidėjo atsiranda pranešimas, kad ginklas perkaito.

Pauzės meniu langas – šis langas atidaromas, kai žaidėjas paspaudžia pauzės mygtuką („P“). Šiame lange yra trys mygtukai:

- tęsti žaidimą („Resume“), šis mygtukas sugrąžina atgal į žaidimo sesijos langą;
- nustatymai („Settings“), šis mygtukas atidaro nustatymų langą;
- išjungti („Quit“), šis mygtukas sugrąžina vartotoją atgal į pagrindinį meniu.

4. Problemos ir jų sprendimai

Realizuojant projektą viskas vyko sklandžiai, tačiau buvo iškilusios kelios problemos.

Viena iš problemų buvo dirbtinio intelekto nusitaikymas į žaidėją naudojantis priedanga. Jei žaidėjas stovi tiesiai prieš priešą, priešas visada pataikys į žaidėją, bet jei žaidėjas pasitraukia į šoną 90 laipsnių kampų, priešų judėjimo animacijos neleisdavo jam pasisukti ir nusitaikyti daugiau nei 60 laipsnių kampų. Ši problema buvo išspręsta sukuriant „AimOffset“. Naudojant „AimOffset“ yra sukuriamos atskiros nusitaikymo animacijos, kurios nekonfliktuoja su kitomis animacijomis.

Kita iškilusi problema buvo, kai du „Cover“ tipo priešai mėgino naudotis vienu priedangos objektu ir kartais vienas iš jų negalėdavo pasinaudoti priedanga. Dažniausiai tas priešas, kuris jau naudojosi priedanga, užtvėrdavo kelią kitam priešui, kuris mėgino taip pat pasinaudoti ta pačia priedanga, tuomet priešas, kuris mėgino pasinaudoti priedanga, paprasčiausi stovėjo vietoje ir šaudė į žaidėją. Šią problemą galėjo sukelti tiek užrašytas priedangos naudojimosi scenarijus elgesio medyje, tiek ir naudojamos animacijos. Ši problema buvo neišspręsta iki galo, tačiau buvo rastas būdas, kaip sumažinti šios problemos atsiradimą. Priešai buvo išdėlioti didesniu atstumu vienas nuo kito, taip pat priedangos objektų kiekis lygyje buvo padidintas, kad priešai galėtų pasirinkti atskirus priedangos objektus.

Dar viena iškilusi problema buvo, kad žaidėjas nežinodavo, ar pataikė į priešą. Ši problema buvo lengvai pašalinta panaudojus progreso juostą, kuri parodo, kiek priešas turi gyvybių, taip suteikiant žaidėjui informaciją, ar jis pataiko į priešą.

5. Išvados

1. Remiantis darbo realizavimo metu įgyta patirtimi, galima teigti, kad kuriant žaidimo langus ir jų būsenas, naudoti Unreal Engine 4 nėra sudėtinga, nes variklis turi labai patogius įrankius žaidimo langų kūrimui.
2. Realizavus projektą buvo pastebėta, kad „Top-Down“ žanro šaudyklę, kuriant su Unreal Engine 4, galima nesunkiai perdaryti į kitą šaudyklės žanrą.
3. Kuriant žaidimą nustatyta, kad standartiniai Unreal Engine 4 animacijos įrankiai turi trūkumų, todėl programuotojui tenka juos patobulinti.
4. Panaudojus žaidime dirbtinio intelekto elgesio medžio algoritmą, galima tvirtinti, kad Unreal Engine 4 variklis turi puikius standartinius įrankius, veikėju elgesio valdymui kurti.

Anotacija

Autorius: Donatas Skrodenis

Tema: Šaudyklės žaidimo kūrimas

Šiaulių universitetas 2017

Šio darbo pagrindinis tikslas - realizuoti šaudyklės žanro žaidimą.

Darbe išanalizuota, kokius požanrius turi šaudyklės žanro žaidimai, analizuojama, kokie yra dirbtinio intelekto elgesio kūrimo būdai, jų veikimo principai ir kaip jie pritaikomi šaudyklėse. Buvo išanalizuoti pasirinkti įrankiai žaidimo kūrimui. Sukurtas pradinis projektas, kuriame parodomos numatomo žaidimo funkcionalumo schemos ir numatomos žaidimo būsenos. Taip pat sukurtas galutinis projektas, kuriame pavaizduotos ir aprašytos realizuotame žaidime esančios funkcionalumo schemos ir būsenos. Aprašyti problemų sprendimai ir galutinio darbo rezultatai.

Summary

Author: Donatas Skrodenis

Subject: Shooter Game Development

Šiauliai University 2017

The main purpose of this work is to develop a shooter game.

This work analyses what subgenres the shooter genre has, what methods are used to create artificial intelligence behavior, how these methods work and used in shooter games. The tools used to create the game are analyzed. Initial project is created, that consists of functionality schemes and game states, expected to be created and the final project, that consists of functionality schemes and games states which are used in the final game project. Problems that occurred in the game are analyzed and the final result of the game is described.

Literatūros sąrašas

1. Unity3d.com. *Unity5.5*. [žiūrėta 2017-02-04]
<https://unity3d.com/>
2. unrealengine.com. *what is unreal engine 4*. [žiūrėta 2017-02-04]
<https://www.unrealengine.com>
3. docs.unrealengine.com. *Behavior tree*. [žiūrėta 2017-02-10]
<https://docs.unrealengine.com/latest/INT/Engine/AI/BehaviorTrees/index.html>
4. Artificial Intelligence For Games(Second Edition). Ian Millington. John Funge (2009)
5. Mark J. P. Wolf (2008). *The video game explosion: a history from PONG to PlayStation and beyond*
6. techopedia.com. *First person shooter fps*. [žiūrėta 2017-02-20]
<https://www.techopedia.com/definition/241/first-person-shooter-fps>
7. gamasutra.com. *differences between firstperson and thirperson*. [žiūrėta 2017-02-25]
http://www.gamasutra.com/blogs/MichelSabbagh/20150827/252341/The_important_differences_between_firstperson_and_thirdperson_games.php
8. mobygames.com. *tactical shooter*. [žiūrėta 2017-03-02]
<http://www.mobygames.com/game-group/genre-tactical-shooter>
9. gamasutra.com. *hero shooters*. [žiūrėta 2017-03-05]
http://www.gamasutra.com/view/news/271933/Hero_Shooters_Charting_the_rebirth_of_a_genre.php
10. autodesk.com. *Autodesk 3ds Max*. [žiūrėta 2017-03-06]
<http://www.autodesk.com/products/3ds-max/overview>
11. microsoft.com. *Visual Studios 2015*. [žiūrėta 2017-03-10]
<https://msdn.microsoft.com/en-us/library/dd831853.aspx>
12. cleverism.com. *Gaming industry introduction*. [žiūrėta 2017-02-18]
<https://www.cleverism.com/gaming-industry-introduction/>
13. thocp.net. *Video game genres*. [žiūrėta 2017-02-22]
<https://www.thocp.net/software/games/reference/genres.htm>

Priedai

CD turinys

1. Žaidimo projektiniai failai (norint paleisti reikia turėti Unreal Engine 4 4.13.0 versiją).
2. Realizuotas žaidimas (paleisti žaidimui reikia kompiuterio su 64 bitų Windows operacine sistema, taip pat kompiuteris turi turėti *microsoft visual studio c++ 2015 redistributable*).
3. Bakalaurinio rašto darbo trys skirtingi dokumento formatai.
4. Darbe naudotos diagramos.

Testavimo protokolas

1.1. Tikslai

- Patikrinti, ar žaidime veikia visos sukurtos funkcijos;
- Surasti kodo klaidas ir jas ištaisyti;
- Patikrinti, ar žaidimo metu dirbtinis intelektas atlieka visas funkcijas;
- Nustatyti, ar žaidimas veikia su skirtingomis Windows operacinės sistemos versijomis;

1.2. Testavimo resursai

Techninės įrangos resursai, su kuriais atliktas testavimas:

- Procesorius – AMD FX-8300 8 Core 3.6GHz
- Operatyvioji atmintis – Kingston DDR 3 10Gb 1600 dažnis
- HDD – Samsung HD503HI 1Tb 7200rpm
- Vaizdo plokštė – Nvidia Geforce GTX 950 2Gb
- Monitorius – 1280x1024 rezolucija

Programinė įranga:

- Operacinė sistema – Windows 7 Ultimate SP1 x64
- Variklis – Unreal Engine 4 version 4.13.0

1.3. Programinės įrangos/sistemos funkcijų testavimas

Testo ID	Reikalavimai	Įvykis / įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos
1.	Žaidėjo veikėjo judėjimas lygyje	Pradedama judėti	Veikėjas pradeda judėti	Žaidėjo veikėjas juda	T	
2.	Ginklų šaudymas	Pradedama šaudyti	Kulkos iššovimas	Kulka iššaunama iš ginklo	T	
3.	Ginklo keitimas	Pasukus pelės ratuką pasikeičia ginklas	Ginklo pakeitimas	Ginklas yra pakeičiamas	T	Jei pelės ratukas yra pasukamas per greit, ginklas gali nepasikeisti.
4.	“Assault” tipo priešai	Priešas, pamatęs žaidėją, jį puola, kol jis yra 5 m atstumu nuo jo	Priešas puola žaidėją	Priešas puola žaidėją	T	
5.	“Cover” tipo priešai	Priešas, pamatęs žaidėją, ieško priedangos	Priešas naudojami priedanga	Priešas naudojami priedanga tol, kol žaidėjas jo nenušauna ar neprieina per arti	T	
6.	“Heavy” tipo priešai	Priešas, pamatęs žaidėją, ieško priedangos labiausiai nutolusios nuo žaidėjo	Priešas naudojami priedanga	Priešas naudojami priedanga tol, kol žaidėjas jo nenušauna ar neprieina per arti	T	

1.4. Suderinamumo testavimas

Testo tikslas:							
Testo ID	Reikalavimai/tikslai	Įvykis/įvestis	Laukiamas rezultatas	Gautas rezultatas	T/N	Pastabos	
1	Žaidimas veikia su skirtingomis OS	Windows 7 X64, Windows 8 X64	Žaidimas veikia	Žaidimas veikia	T		
2	Žaidimas veikia su skirtingomis OS	Windows 7 X86, Windows 8 X86	Žaidimo Neveikia	Žaidimas neveikia	N	Norint paleisti žaidimą su 32 bitų sistema Unreal Engine 4, reikia pasirinkti, kad žaidimas būtų sukurtas 32 bitų OS	

1.5. Testavimo išvados

- Atlikus testavimą, buvo rasta klaidų, bet jos nėra kritinės ir neturi įtakos žaidimo veikimui.
- Atlikus dirbtinio intelekto elgesio medžio testavimą, nebuvo aptikta kritinių klaidų, kurios turėtų didelės įtakos žaidimui.
- Suderinamumo testavimas parodė, kad žaidimas veikia tik su 64 bitų Windows operacinėmis sistemomis.