

Документацията на проекта

1. Тема: Стрелкова игра с изкуствен интелект

2. Автор:

- a. Михаил Ивайлов Киров
- b. ЕГН: 0151086647
- c. адрес: София, Младост 2, бл 205-Б, вх 1 ет 3 ап 18
- d. телефон: 088991516
- e. email: mihail.kirov2@gmail.com
- f. училище: Технологично училище “Електронни системи” към ТУ-София
- g. клас: 12

3. РЪКОВОДИТЕЛ:

- a. Виктор Кетипов
- b. телефон: 0896843015
- c. email: viktor@kipiinteractive.com
- d. длъжност: учител в Технологично училище “Електронни системи” към ТУ-София

4. Резюме:

1. Цели (предназначение, кратък анализ на потребностите и на съществуващите решения)

Триизмерна стрелкова игра с различни видове оръжия, поддържаща мултиплейър. Режима на играта е кооперативен. Позволява на 4 играчи да играят заедно срещу противник, имплементиран чрез изкуствен интелект. Цели да създаде по-добър изкуствен интелект за традиционните стрелкови игри.

Съществуват малък на брой игри, използващи обучение с утвърждение за създаване на реалистични противникови герой. Reese A. Boyd сравнява обикновения

изкуствен интелект чрез Behavior Tree и изкуствен интелект трениран чрез обучение с утвърждение. Той контролира противниковия герой чрез табличен Q-learning. За своя проект обаче той ползва визуално Blueprint програмиране, което макар и лесно за създаване се оказва изключително трудно за разбиране от други програмисти и за дебъгване. Тази дипломна работа е вдъхновена от неговия проект, но разширява функционалността чрез добавяне на мултиплейър функционалност и писане на целия алгоритъм в програмен код. Това ще улесни разбираемостта на алгоритъма и неговото бързодействие и мащабируемост.

2. Основни етапи в реализирането на проекта (основни дейности, роли на авторите)

Разработката на проекта следва Agile методологията - циклично планиране, анализиране, дизайн, имплементация, тестване за кратък период от време.

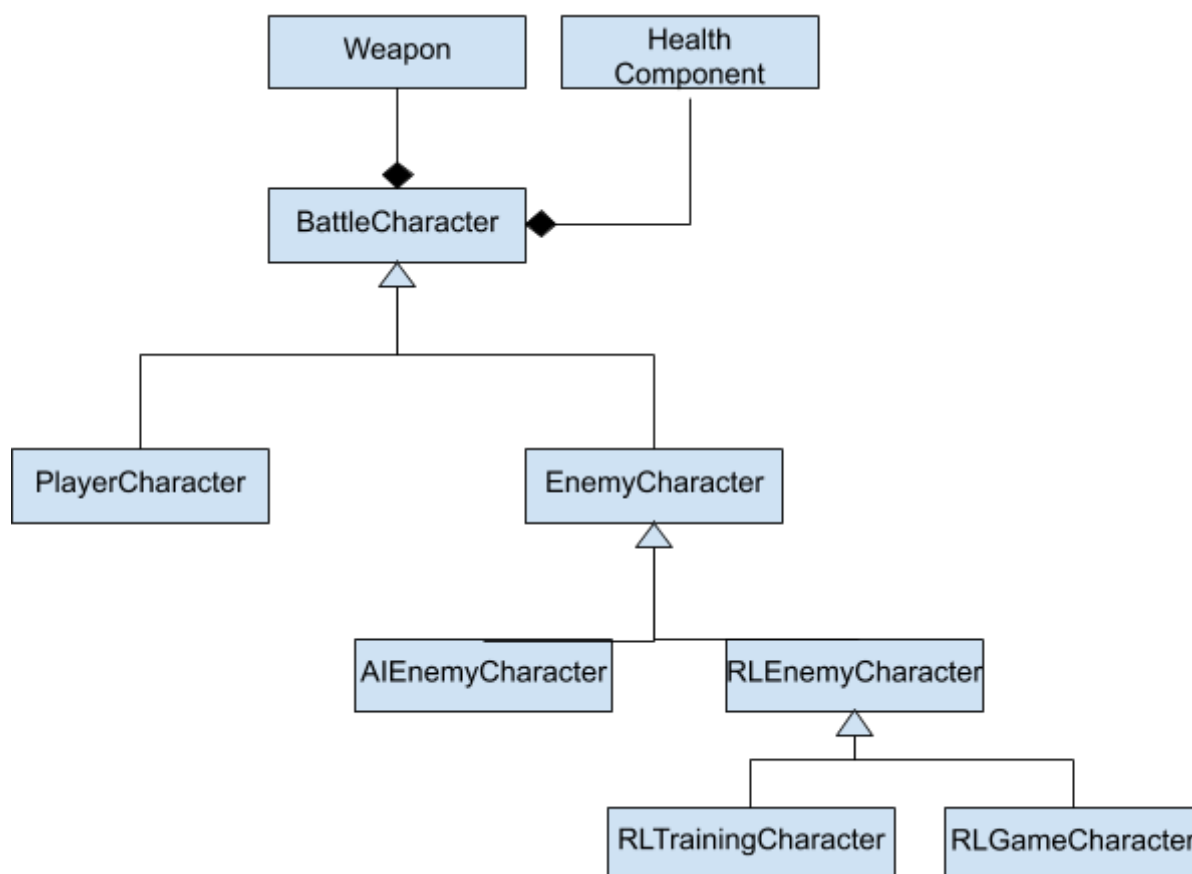
3. Ниво на сложност на проекта – основни проблеми при реализация на поставените цели

Има две основни сложности в реализацията на проекта - имплементацията на мултиплейър кооперативен режим на играта и имплементацията на изкуствен интелект, използващ машинно самообучение чрез утвърждение.

Основните проблеми се изразяват в тестването на мултиплейър игра на локалната машина на разработчика както и тестване на изкуствения интелект и промяна на средата с цел адаптация и по-добро и желано действие от агента.

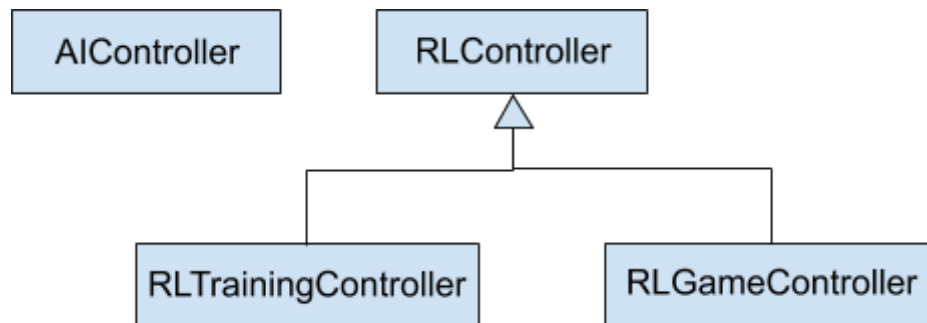
4. Логическо и функционално описание на решението – архитектура, от какви модули е изградено, какви са функциите на всеки модул, какви са взаимодействията помежду им и т.н.ю.

В основата на йерархията е базовият клас за играч “BattleCharacter”. Той предоставя функционалност за взаимодействие с оръжия, умирање и други основни функции, споделяни от всеки играч, Композира различни главни функционалности, като базовия клас за оръжие и компонент за кръвта на героя. От него наследяват класовете “PlayerCharacter” и “EnemyCharacter”, които допълват функционалността на базовия клас. Противниковия клас се разделя на “AIEnemyCharacter” и “RLEnemyCharacter”. Вторият е допълнително разделен на “RLTrainingCharacter” и “RLGameCharacter”:



Всеки играч е контролиран от контролер клас. При тях също има йерархия. Има ‘EnemyAIController’, който управлява противниците, имплементирани чрез автомат на състоянията. Имплементирани са и базов клас “RLController”, който включва базовата

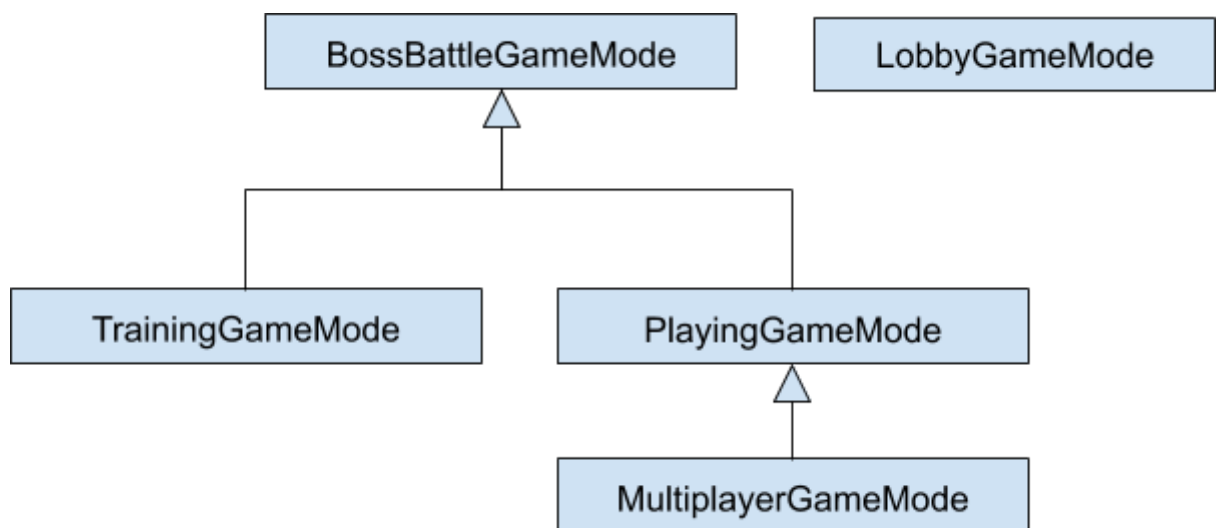
функционалност за противник, използващ машинно самообучение с утвърждение. От него наследяват двата контролера “RLTrainingController”, който служи за трениране, и “RLGameController”, който ползва натренираната таблица за същинската игра.



Потребителският интерфейс е съставен от плоска йерархия от “HUD” и “Widget” класове.

За анимацията се грижи единствен C++ клас “CharacterAnimInstance”, което значи, че анимацията на всички играчи е еднаква.

Режимът на игра се диктува от “GameModее” класовете. Имплементиран е базов клас “BossBattleGameMode”, който декларира виртуални функции, които имат различна имплементация в различните наследяващи класове:



5. Реализация – обосновка за използвани технологични средства, алгоритми, литература, програмни приложения и др.

С невероятната си графична мощ и изключително лесни за използване инструменти, Unreal е идеалният продукт, който ще позволи постигането на естетически приятна игра, която също има добре изградена функционалност и здрави игрови механики в своята опора. Освен това игровия двигател предлага и вече изработени текстури, които са за свободно ползване от потребителя. Unreal също може да се компилира за широк диапазон от устройства. От PC до Xbox One, PS4, Android, iOS, както и много от модерните VR платформи. Гъвкавостта на програмата ѝ позволява да изпълнява един и същ код на множество различни машини. Unreal също притежава способността да използва DirectX 12, което му позволява да бъде оптимално ефективен в работата си, тъй като, в сравнение с предшественика си, DirectX 12 е по-оптимизиран и позволява още по-директен контрол върху хардуера, което означава, че едни и същи обекти се изобразяват на екрана за по-кратко време, когато се използва DX12. В новите ъпдейти на софтуера, Epic Games също добавят изцяло нов инструмент за създаване на анимации по време на игра, както и подобрения върху изкуствения интелект на героите.

C++ е неспециализиран език за програмиране от високо ниво. Той е обектно-ориентиран език със статични типове. От 1990-те, C++ е един от най-популярните комерсиални езици за програмиране. C++ съдържа вградена в езика поддръжка на обектно-ориентирано програмиране. В C++ са добавени класове, множествено наследяване, виртуални функции, overloading, шаблони (templates), обработка на изключения (exceptions) и вградени оператори за работа с динамична памет. Единствената част, в която има повече смисъл да бъде използван BP е потребителския интерфейс. Unreal Engine позволява много удобна разработка без използване на специалния език за разработка на потребителски интерфейс - Slate.

Програмирайки на C++ ние имаме възможността да представяме по-лесно взаимоотношенията и йерархията между класовете в играта.

Основния използван алгоритъм за играта е този за машинно самообучение чрез утвърждение. Представява табличен Q-learning, имплементиран изцяло чрез йерархия от C++ класове.

Използваната литература се състои от документацията на Unreal Engine 4, статии от Unreal Engine Wiki, UE4 форум и работата на Michael Boyd свързана с изкуствен интелект.

6. Описание на приложението – как се стартира и/или инсталира, как се използва, как се поддържа

Приложението е компилирано и бинарните файлове на дедикирания сървър на играта и клиентската част могат да бъдат открити в гитхъб репото на проекта. Стъпките за инсталация са следните:

- Отидете в секция “releases” на репото на играта в Github: <https://github.com/mikirov/Elsys-Thesis-Work-2019-2020/releases>
- Изтеглете последната версия на WindowsNoEditor.rar
- Разархивирайте WindowsNoEditor.rar
- Навигирайте до WindowsNoEditor\BossBattle\Binaries\Win64
- Създайте препратка към файла “BossBattleServer.exe”
- Отворете настройките “Properties” на препратката и добавете накрая на “Target” опцията “-log”.
- Пуснете сървъра на играта чрез файла “BossBattleServer.exe - Shortcut”
- Пуснете клиентска инстанция на играта чрез файла “BossBattle.exe”

7. Заключение – какъв е основният резултат, дали има приложения до момента, какви възможности съществуват за развитие и усъвършенстване

Машинното самообучение чрез утвърждение е много капризно към средата, с която агента взаимодейства. Тя може да бъде променена, за да бъдат получени нови интересни поведения от него.

Разработена бе кооперативна стрелкова игра, в която няколко играча играят срещу противник с изкуствен интелект. Имплементиран бе потребителски интерфейс за управление на тренирането на агента и показващ всичката релевантна информация за процеса на самообучение.

Бъдещото развитие също би включвало добавяне на произволен брой действия, състояния на играта и награди, които да могат да бъдат контролирани от разработчика.

Този проект до голяма степен успява да имплементира иновативен изкуствен интелект, който при правилно трениране може да даде на потребителите по-интересни и реалистични противникови герои.