



Софийски университет „Св. Кл.
Охридски”

Факултет по математика и информатика



Бакалавърска програма
„Софтуерно инженерство”

Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2025/2026 год.

Тема №53 “Учебна видео игра от тип стрелба”

Курсов проект

Автори:

Владимир Коцев, фак. номер 6MI0600278

Радин Тихолов, фак. номер 4MI0600310

януари, 2026 г.

София

Съдържание

1	Въведение.....	3
2	Анализ на решението.....	3
2.1	Игрови процес.....	3
3	Дизайн.....	3
3.1	Структура на входното и изходното текстово съдържание.....	3
3.2	Тип и представяне на аудио-визуалното съдържание.....	4
3.3	Описание на проекта в Unity.....	4
3.4	Дизайн на нивата на играта.....	6
4	Използване на изкуствен интелект.....	6
5	Тестване.....	6
6	Заключение и възможно бъдещо развитие.....	10
7	Разпределение на работата.....	10
8	Използвани литературни източници и Уеб сайтове.....	11

1 Въведение

Разработената система е триизмерна компютърна игра от жанра "Шутър от първо лице" (First-Person Shooter - FPS), реализирана чрез Unity. Проектът комбинира аркадни елементи на стрелба с образователен компонент за събиране на исторически данни. Играчът влиза в ролята на изследовател, чиято задача е да свали въздушни балони, транспортиращи ценни исторически артефакти, и да събере съдържанието им, за да

попълни своята колекция (Инвентар). Играта акцентира върху прецизността на стрелбата спрямо дистанцията на мишените.

2 Анализ на решението

2.1 Игрови процес

Основната цел на играча е да премине през поредица от нива, дефинирани във външен XML файл, като събере всички заложени артефакти с минимален брой грешки.

Цели на играта:

- Унищожаване на балоните с горещ въздух. Избягване на стрелба по сандъка, който носи артефактите. Събиране на предмети от падналите сандъци (Loot Crates) за напредване в нивото.

Точкова система (Scoring):

- Резултатът е мярка за ефективността на играча и се променя динамично:
 - +1 точка: При успешно попадение в балона (Hit)
 - -.1 точка: При изстрел в празно пространство (Miss).
 - -5 точки: При попадение по сандъка(Penalty). Това е основното предизвикателство за точност.

Нива и условия за край:

- Играта зарежда последователно нива от xml файла. Всяко ниво има различен брой балони и разположение. Нивото приключва успешно само когато играчът е събрал в инвентара си всички предмети, асоциирани с текущото ниво. След завършване на последното ниво от заредените нива, играта приключва и записва финалния резултат в текстов файл (GameResults.txt).

Действия на играча (Управление):

Играчът взаимодейства със света чрез стандартна периферия – мишка и клавиатура. Използвана е класическата схема за FPS управление

Стрелба:

Различните оръжия имат различна настройка за стрелбата.

- Пушка - Най-бърза при всички аспекти, но разстоянието, което покрива е най-малко.
- Снайпер - Най-бавен при стреляне и презареждане, но достига до най-далечно разстояние и може да се zoom-ва.
- Пистолет - Средно по бързина и дистанция.

3 Дизайн

3.1 Структура на входното и изходното текстово съдържание

Системата за управление на данни в играта е проектирана да работи с два основни потока на информация: входни конфигурационни данни и изходни данни за резултатите.

Входни данни (XML Конфигурация): Инициализацията на нивата и обектите в тях не е твърдо кодирана (hard-coded), а се извлича от външен файл game.xml, разположен в директорията StreamingAssets. Файлът game.xml допълнително се валидира от файла game.xsd. Структурата на game.xml следва йерархичен модел:

`<level>`: Коренов елемент за всяко ниво.

`<balloon>`: Дефинира вражеска единица (балон), включващ атрибути за hitpoints

`<item>`: Представлява името на предмета

`<description>`: Информация за дадения предмет

`<icon>`: Снимка на предмета

`<trajectory>`: Траекторията на балон, заедно с неговите координати

`</balloon>`

`</level>`

Изходни данни (TXT Резултати): При успешно приключване на играта (събиране на всички артефакти), системата генерира или презаписва файл GameResults.txt. Файлът съдържа структуриран текст с финалния резултат на играча и времето, в което играта е завършена.

3.2 Тип и представяне на аудио-визуалното съдържание

Аудио-визуалното съдържание е разделено на три основни слоя, които осигуряват обратна връзка на играча:

- **3D Графична среда:**

- Използва се перспектива от първо лице (FPS).
- Основните обекти включват: терен (Terrain), модели на оръжия, динамични обекти (балони с физика на движение) и интерактивни обекти (сандъци и награди).
- Използвани са стандартни Shaders на Unity за осветление и текстуриране.

- **Потребителски интерфейс (UI - 2D):**

- HUD (Heads-Up Display): Показва текущия мерник (Crosshair), натрупаните точки и избраното оръжие.

- Инвентарна система: Визуализира се чрез статични 2D спрайтове (изображения) на артефактите. Въпреки че спрайтовете може да са повтарящи се, текстовото поле до тях се попълва динамично с информацията от XML файла (напр. „Антична ваза от 300 г. пр.н.е.“).
- **Аудио ефекти (SFX):**
 - Звукова индикация при стрелба (различна за пистолет, снайпер и пушка).
 - Звук при пукане на балон и падане на сандък.
 - Звук за грешка/наказание при уцелване на кошницата.

3.3 Описание на проекта в Unity

Проекта в unity има няколко основни папки:

- Effects – Съдържа визуални ефекти, реализирани чрез Unity Particle System, използвани при стрелба, експлозии и разрушаване на балони.
- Image – Съдържа статични изображения, използвани в потребителския интерфейс и менютата на играта.
- Models – Включва триизмерни модели, използвани за обекти и елементи в игровия свят.
- PlayerAssets – Съдържа всички ресурси, свързани с оръжията на играча, включително 3D модели, prefabs, анимации и звукови ефекти.
- Resources – Съдържа папка ItemIcons, в която се намират икони на предмети, зареждани динамично по име, съответстващо на стойността в XML файла (напр. <icon>Coin</icon>). Ако трябва да се добави нова иконка трябва да се добави изображението в папката като Sprite и да се реферира по име в XML файла.
- RPG_FPS_game_assets_industrial – Съдържа тар-а.
- Scenes – Съдържа единствената игрова сцена, в която се развива цялата игра.
- TargetAssets – Съдържа prefab-и за различните типове балони, използвани като цели в играта.

Програмни скриптове:

- Скриптове за XML. Имаме класове, която описва xml йерархията и позволява работа с пернатите обекти. Допълнително има XmlLoader и XmlValidator, които се използват за зареждане и валидиране на xml файла.
- UIManager, който менажира в какъв етап се намира потребителския интерфейс чрез използването на UIState. Идеята е да може динамично да се разширява интерфейса.
- Target скриптове. Имаме по един скрипт за всеки тип позволено движение на балон. (LinearTarget, CurvedTarget, ExpertTarget)
- За оръжията имаме базов клас Gun, който се разширява от Gun, Pistol и Shotgun, като по този начин се преизползва логиката. Допълнително имаме

GunManager, който отговаря за използването и промяната на оръжията по време на играта.

- Reward - Отговаря за всичко свързано с наградата за сваляне на балон.
- MouseMovement и PlayerMovement изцяло отговарят за логиката по контролиране на играча. Оказват как се движи играча и как се контролира мишката.
- В Inventory папката е всичко свързано с инвентара. Имаме InventoryItem, който описва елемент от инвентара и допълнително имаме класове, които го използват за да постигнем искания резултат. В папката се намират скриптовете за управлението на UI-а на инвентара.
- Имаме и GameScoreboard, който изцяло отговаря за точките на играча.

Аудио файловете, които се използват в играта са в папката PlayerAssets, като имаме по един ефект за всяко оръжие и имаме ефекти за стъпките на героя. Допълнително в папката Prefabs имаме звуков ефект за падането на сандък от височина.

3.4 Дизайн на нивата на играта

Нивата на трудност се задават във входния XML документ и играчът успява да премине на следващо ниво, когато успее да свали всички балони за даденото ниво. Даден балон има точки на кръв, които също са зададени в XML файла. Допълнително в папката Prefabs имаме звуков ефект за падането на сандък от височина.

4 Използване на изкуствен интелект

В процеса на разработка на проекта беше използван изкуствен интелект като помощен инструмент при реализирането на определени игрови механики, с цел по-ефективно изпълнение на условията на заданието. Използваният модел представлява голям езиков модел (Large Language Model), базиран на архитектура тип transformer, достъпен чрез чат-базиран интерфейс (ChatGPT).

Контекстът на решението беше зададен чрез подробно описание на игровата концепция, използвания game engine (Unity), както и конкретните изисквания за механиките, които трябва да бъдат реализирани. Формулирането на промптовете беше насочено към получаване на предложения за алгоритми и логика за движение на различни типове балони (линейно, криволинейно, експертно), както и за визуален и логически индикатор за стрелба, подпомагащ играча при прицелване и оценка на точността.

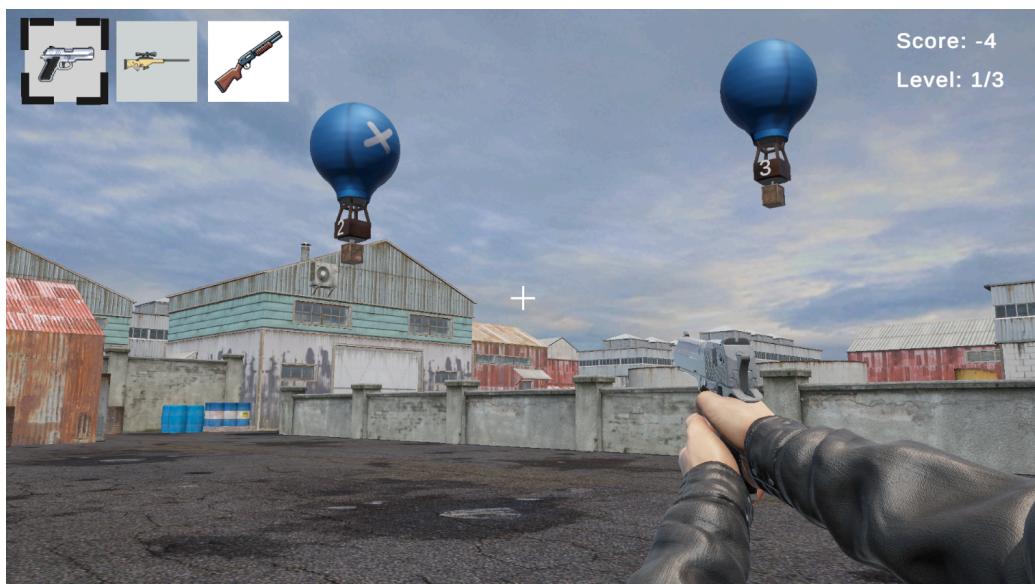
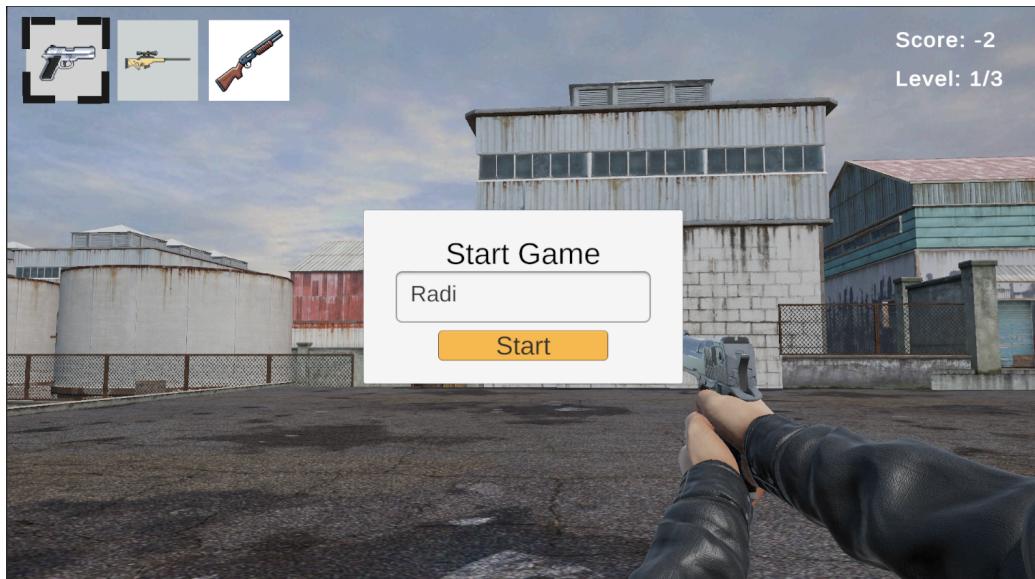
Чрез използването на ИИ бяха генериирани идеи и примерни реализации за поведението на балоните, включително параметризираны траектории, които впоследствие бяха адаптирани в XML-базираната система за нива. Освен това ИИ беше използван и за подобреие на начина по който се индицира дали даден балон е уцелен.

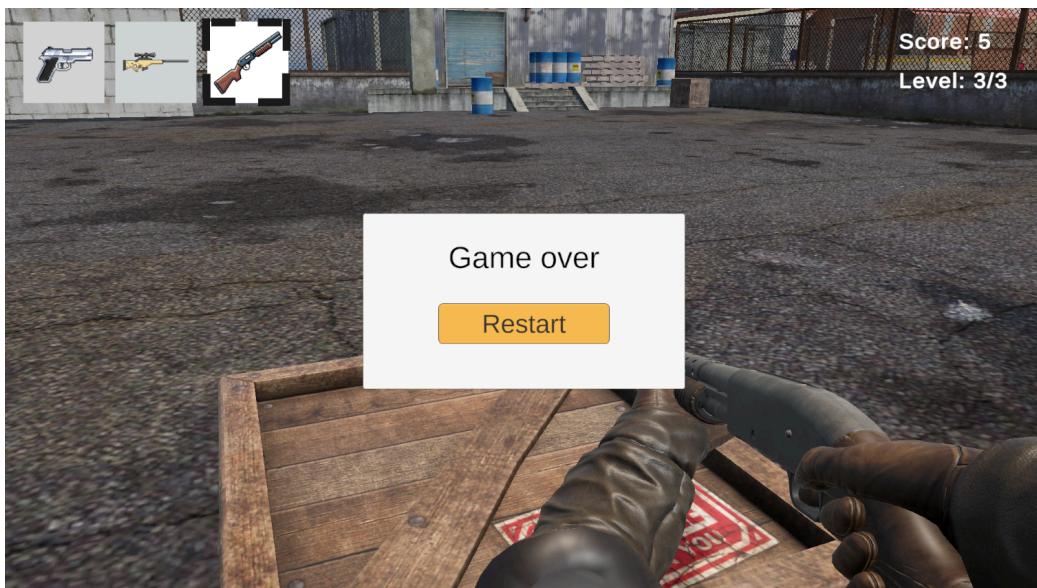
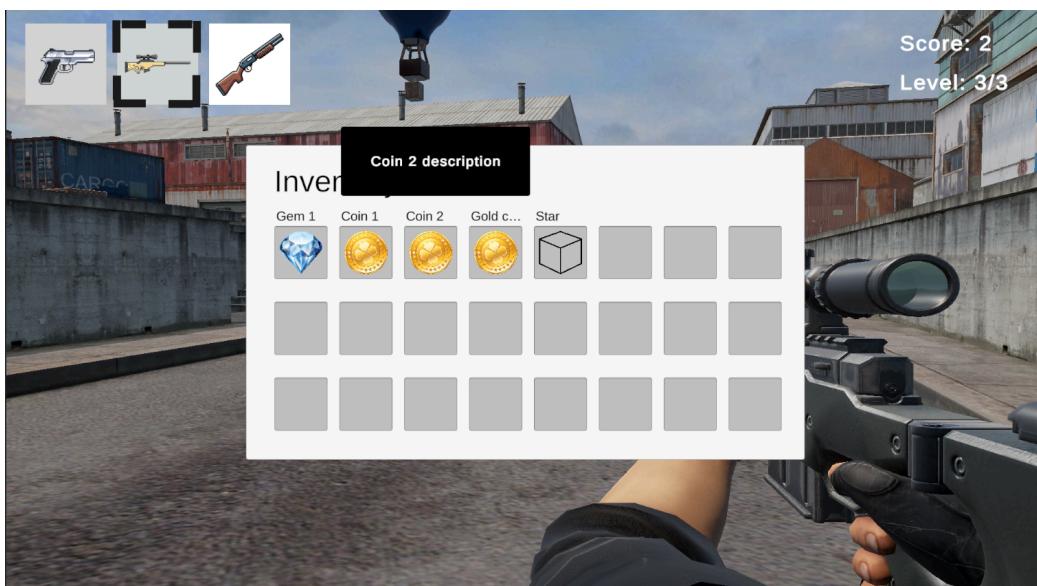
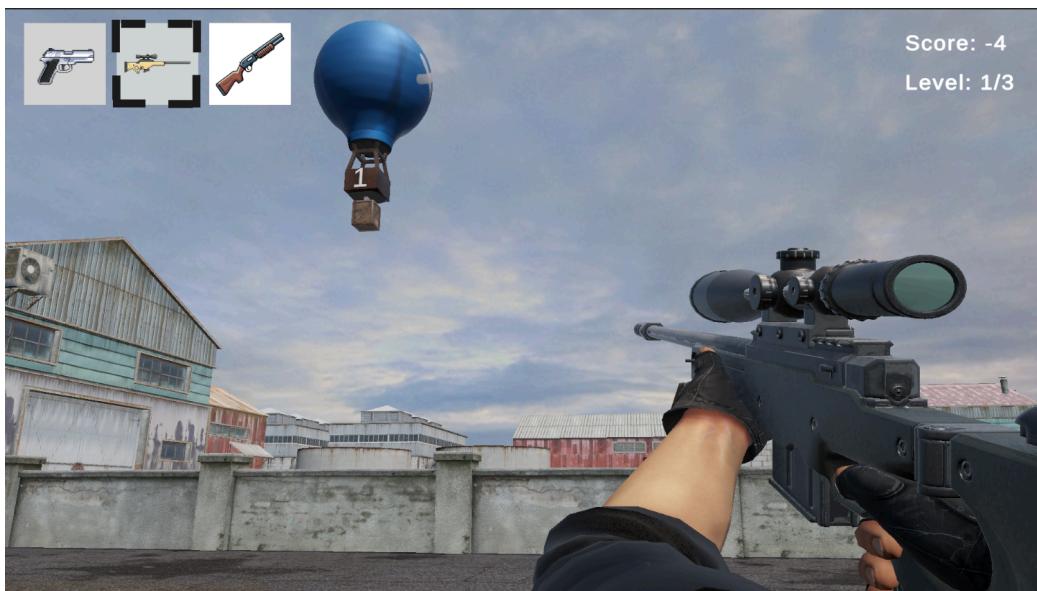
5 Тестване

Играта е тествана в редактора на Unity, като сме използвали различно настроени XML файлове за да симулираме различни нива. Тестове които направихме:

- За правилно четене и валидиране на XML документа
- Правилно сериализиране на изходния файл с информацията за играта
- Движението на играчът в различни ситуации по време на игра
- Механиките свързани с инвентар и сандъци
- Промяна на избрано оръжие
- Стрелба и сваляне на даден балон
- Прицелване със снайпер
- Обхват на стрелбата с различните оръжия
- Правилно премахване на точка при неуцелване и правилно премахване на пет точки при уцелване на сандък

Build-а на играта е тестван в Windows среда, като работи безпроблемно.







6 Заключение и възможно бъдещо развитие

В заключение, резултатът от извършената работа е разработването на FPS игра с обучителен характер, която успешно съчетава развлекателни елементи с образователно съдържание. Чрез динамичен геймплей и система за събиране на артефакти, играта има потенциал да привлече и задържи вниманието на учениците, като едновременно с това подпомага усвояването на знания по интерактивен и ангажиращ начин. Използването на XML за дефиниране на нивата позволява гъвкавост, лесна поддръжка и разширяемост на съдържанието без необходимост от прекомпилиране на кода, което е съществено предимство на избрания подход.

Като алтернативи за разработка на играта могат да бъдат използвани други game engines като Unreal Engine и Godot Engine. Unreal Engine предлага по-реалистична графика и мощни инструменти за визуално програмиране чрез Blueprint система, но е по-ресурсоемък и по-сложен за начинаещи. Godot Engine, от своя страна, е с отворен код и по-лек, което го прави подходящ за по-малки проекти, но разполага с по-ограничена екосистема и по-малко готови ресурси в сравнение с Unity.

Като насоки за развитие на играта могат да се предложат добавяне на повече нива с разнообразни исторически периоди и артефакти, подобряване на инвентарната система с допълнителна информация и мултимедийно съдържание, както и създаване на система за оценяване на знанията на играта. Допълнително може да се реализира мултиплейър, система за постижения, която да се настройва спрямо уменията на играта.

7 Разпределение на работата

Радин:

- Inventory system
- XML parsing
- XML validation
- User Interface
- Dynamic levels mechanics and levels flow
- Integrated map
- Reward object fall dynamics

Владимир:

- Weapon mechanics
- Animations
- Balloon mechanics
- Sound effects
- Partially HUD UI

8 Използвани литературни източници и Уеб сайтове

1. <https://docs.unity3d.com/>
2. <https://discussions.unity.com/>
3. <https://stackoverflow.com/>
4. <https://assetstore.unity.com>

Линк към GitHub репото:

<https://github.com/VladimirKotsev/XMLCourseGame>