# Plan project: Trivia game

**By:** Simeon Markov

**Department:** ICT, Fontys UAS

**Course Name:** STA-OIL430-CMK

**Class:** EN08

## Introduction

The game emphasizes on education, exploring different fields like history, physics, geography, science, etc. Its purpose is to put someone's general knowledge into test and to enrich it. The game falls under the category of puzzle games, targeting skills like logic and general awareness. I will be implementing the game using modern technologies, tools, libraries and techniques and the user experience would the game's priority.

*Keywords*: education, realization, goal

## Objectives of the project

The goal is to design and implement a functioning game within the timespan of four weeks, providing:

**INTUITIVE USER INTERFACE**

Interface that is easy for navigation, following the layout concepts.

**PERFORMANCE**

Reasonable responsive time for data loading.

**CONTENT**

Fetching random questions from different categories using openly distributed APIs.

All functions are based on the *user requirements specifications* that the game meets.

My contributions are:

- Setting up the development environment.
- Designing, wireframing and prototyping.
- Implementation.
- Integrating database into the project.
- Data visualization, reverse engineering.

## Main questions

1) How could I come up with a product that is entertaining and at the same time educational.
2) What kind of technologies, game engines, would work better and more efficiently for my game and why.
3) What would be my learning outcomes during and after the realization.

**MoSCoW method**

| | |
|---|---|
| Must have | Multiple-choice questions, levels, category, progress, scoring system |
| Should have | Tailored feedback, educational explanations |
| Could have | Timer for counting down the remaining time for the game completion. |
| Will/Wish have | AI chatbot for providing further insights into the game |

**Scrum methodology**

| Phase | Description |
|---|---|
| Week 1 – Research | Doing research on the topic, exploring methods of realization, technologies, etc. |
| Week 2 – Design | Wireframing the conceptual idea and coming up with a prototype. |
| Week 3 - Implementation | Working on the actual realization of the game, considering all of the requirements. |
| Week 4 – Test and evaluation | Testing small portions of the product and improving where necessary. |

**Indirect Stakeholders**

### Educational Institutions

- **Why they matter**: Schools, universities, and training centers may adopt the game as a learning tool or recommend it to students.
- **Impact**: If the game proves effective in boosting engagement or comprehension, institutions might integrate it.
- **Interest**: They'll care about educational value, alignment with learning outcomes, and accessibility.

### Parents and Guardians

- **Why they matter**: Especially relevant if younger users are involved.
- **Impact**: Parents may influence whether their children use the game, monitor usage, or even provide feedback.
- **Interest**: Safety, age-appropriate content, educational benefit.

## Preconditions

- The project is extensible and maintainable.
- Development duration: 4 weeks.
- The project complies with the AI transparency regulations.

## Applied areas of knowledge

| *Field* | *Application* |
|---|---|
| *Media design* | Frontend, UI/UX designing |
| *Software development* | Backend with C# |
| *Game Development* | Working with game engine |

## System requirements

- Installing the latest version of Visual Studio 2022
- Installing the latest version of Unity

**Techstack**

**Includes**: Unity game engine, C#, open APIs, Figma, Visual Studio, Unity UI components.

| Feature | Unity | Unreal Engine | Godot |
|---|---|---|---|
| 1 Ease of use | Most beginner-friendly; intuitive editor | Steeper learning curve; advanced terminology | Visual scripting helps, but still requires coding |
| 2 Graphics & Rendering | Good, improving with SRP; not AAA-level | Best-in-class photorealism; global illumination | Decent for indie games; |
| 3 Platform Support | 25+ platforms; easy porting | Broad support; | Limited console support |
| 4 Programming Language | C# (easy to learn) | C++ (powerful but complex) | GDScript |
| 5 Asset Store | Massive marketplace | Growing asset library | Smaller, community-driven assets |
| 6 Learning Resources | Extensive tutorials and forums | Good documentation | Improving, but less beginner-focused |
| 7 Community Size | Largest and most active | Large and skilled | Passionate but smaller |

*Table 1: Comparison between different game engines*

**Learning outcomes**

Setting up environments

- Installing MS Visual Studio, Unity Editor.

Reporting

- Doing research on the topic, analyzing users, technical requirements.

## Working with Frameworks

- Using Unity testing framework for making unit tests.

## Wireframing, prototyping

- Using platforms like Figma

## Planning

- Utilizing apps like Notion for managing tasks, schudeling.

## Conclusion

With the implementation of the final product with the help of modern technologies and techniques, it highlights cognitive growth and places an emphasis on general awareness. The game offers intuitive design and experience that tests and enriches the user's knowledge. As far as future updates are concerned, features as *adaptive levels* tailored for users by AI model*, category expansion* and *accessibility regime* might follow.

**References**


- Courage, C., & Baxter, K. (2005). Understanding Your Users: A Practical Guide to User Requirements Methods, Tools & Techniques. Morgan Kaufmann.
- Copilot AI, chat for Indirect stakeholders.
- Jude Maxwell. Why Unity is the Future of Game Development: Key Features & Benefits: https://dev.to/jude_maxwell_fe6a32b5a286/why-unity-is-the-future-of-game-development-key-features-benefits-2k91, accessed on 22/09/2025.
- Microsoft documentation. What is Visual Studio? https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022, accessed on 22/09/2025.
- Akshita Zarbade. Why .NET 8 remains the backbone of backend development in 2025 — fast, scalable, cloud-ready, and powered by AI: https://www.c-sharpcorner.com/article/why-net-8-remains-the-backbone-of-backend-development-in-2025-fast-scalable/, accessed on 22/09/2025.