01/03/2025

**Project Definition Document**

Authors: Ali Emre Yenihayat

**Contents**

1. Project Name........................................................................................3

2. Project Summary..................................................................................3

3. Objectives.............................................................................................3

4. Scope....................................................................................................4

5. Target Audience....................................................................................4

6. Key Features.........................................................................................5

7. Deliverables..........................................................................................5

8. Budget and Resources..........................................................................6

9. Risks and Mitigation Strategies............................................................6

10. Project Success Criteria......................................................................7

11. Task Matrix…………………………...……………………………..8

**1. Project Name**

Demand Øf Momentum (DØM)

**2. Project Summary**

This project is a Python-based single-player racing game integrated with Unity. Players will compete against AI-driven bots with adjustable difficulty settings. The game employs deep reinforcement learning to train bots to maneuver efficiently on different race tracks. A top-down perspective will be used, with the track dynamically revealed as the player progresses. The objective is to win the race while strategically avoiding obstacles and utilizing speed boosters.

**3. Objectives**

- Developing an engaging single-player racing experience with AI-controlled bots.

- Implementing deep reinforcement learning to create bots with varying skill levels.

- Designing multiple maps in Unity for different racing experiences.

- Providing a dynamic track visualization system to enhance gameplay.

- Allowing players to configure the number of bots and their difficulty levels.

**4. Scope**

**These are included in the project:**

- Python-based AI bot training using deep reinforcement learning.

- Unity integration for map design and rendering.

- Adjustable difficulty levels for bots (Easy, Medium, Hard).

- Dynamic track rendering based on player movement.

- Speed-altering track features (boosters, slow zones).

- Multiple race tracks to choose from.

**These are not included in the project:**

- Online multiplayer functionality.

- Car customization options beyond basic selection.

**5. Target Audience**

- Racing game enthusiasts.

- AI and machine learning enthusiasts.

- Students and developers interested in reinforcement learning.

**6. Key Features**

- Single-player racing with AI-controlled bots.

- Adjustable bot difficulty (Easy, Medium, Hard) based on reinforcement learning performance.

- Multiple track maps designed in Unity.

- Dynamic camera system revealing only part of the track at a time.

- Speed-altering track elements affecting race strategies.

**7. Deliverables**

- A fully functional single-player racing game.

- Trained AI models for different bot difficulty levels.

- Unity-based track designs.

- Game documentation including installation, usage, and AI training details.

- GitHub repository containing source code and training models.

**8. Budget and Resources**

**Resources Needed:**

- Python for AI training.

- Unity for game development and map design.

- TensorFlow/PyTorch for reinforcement learning.

- GitHub for version control and project collaboration.

**Budget Considerations:**

- Potential cloud computing costs for AI model training.

- Unity license, PyCharm license, Aseprite for designing.

**9. Risks and Mitigation Strategies**

**- AI Training Complexity:** Training AI bots effectively may take longer than expected**. Mitigation:** Start with simpler AI models and refine them iteratively.

**- Integration Challenges:** Ensuring seamless Unity-Python integration could be difficult. **Mitigation:** Use commonly utilized communication protocols.

**- Performance Issues:** Reinforcement Learning models may impact game performance. **Mitigation:** Optimize models and use efficient inference techniques.

**10. Project Success Criteria**

- The game successfully integrates AI-driven bots with adjustable difficulty.

- Players can complete races on multiple tracks with dynamic rendering.

- AI bots exhibit noticeable differences in difficulty levels.

- The game provides a smooth and enjoyable experience without major performance issues.

**11. Task Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks \ Members | Ali Emre Yenihayat | Berk Ülker | Umut Baran Boztaş | Onur Erçen |
| Project Name | X | X | X | X |
| Project Summary | X |  |  |  |
| Objectives | X |  | X |  |
| Scope | X | X | X |  |
| Target Audience | X |  |  | X |
| Key Features | X |  | X |  |
| Deliverables | X |  |  | X |
| Budget and Resources | X | X |  | X |
| Risks and Mitigation Strategies | X | X | X |  |
| Project Success Criteria | X |  |  | X |
| Reviewing and Editing | X | X |  |  |