Berk Ülker, Onur Erçen

**Project Plan Document**

**Table of Contents**

Project Scope……………………………………………………………………………………………2

Project Organization…………………………………………………………………………………2

Objectives………………………………………………………………………………………………..2

Key Phases and Timeline………………………………………………………………………….3

Resource Planning……………………………………………………………………………………4

Risk Management…………………………………………………………………………………….4

Communication Plan………………………………………………………………………………..4

Change Management Plan……………………………………………………………………….4

Budget Plan……………………………………………………………………………………………..4

Effort Estimation for Each Phase...……………………………………………………………5

Estimated Effort Per Task………………………………………………………………………….6

Total Project Effort……………………………………………………………………………………7

Task Matrix………………………………………………………………………………………………8

**Project Scope:**

Since the project is a 2D car racing game; it will developed on Unity and include 2D cars, a pannable screen on map, bots which trained with Deep Reinforcement Learning and control non-player cars, and finally and optionally upgrades which will be takable on road by cars. It will not include multiplayer and LAN, therefore there will not be any local or remote servers.

**Project Organization (Task Assignment):**

Ali Emre Yenihayat (AI Developer): Using Swig, TensorFlow, Pandas, PyTorch on Python to train AI for the car bots.

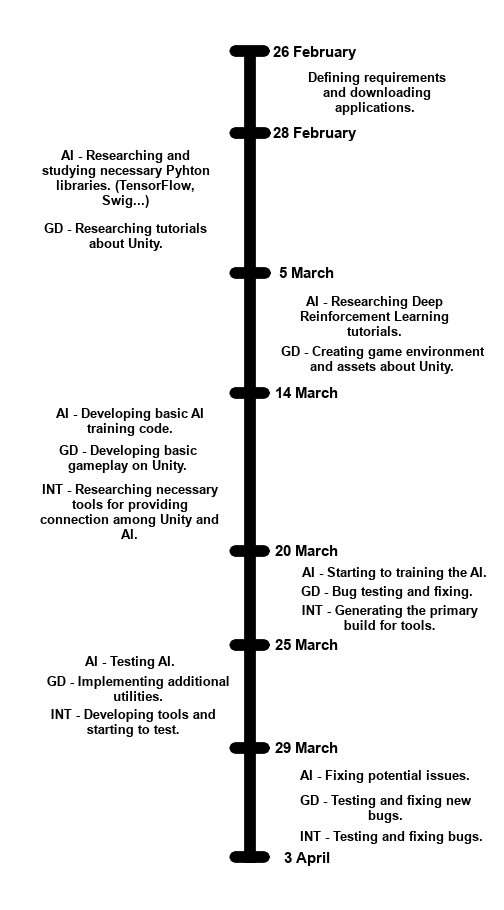
Berk Ülker (AI Developer and Asset Designer): Using Swig, TensorFlow, Pandas, PyTorch on Python to train AI for the car bots. Designing assets for game.

Onur Erçen (Integrator and Game Developer): Creating a connection between AI bots and game engine (Unity). Supporting game development.

Umut Baran Boztaş (Game Developer): Using Unity to create game environment, simulation, and design maps and assets.

**Objectives:**

* Using agile method to create detailed documentation.
* Splitting tasks according to member’s abilities and interests to increase performance.
* Checking all member’s development regularly at scrum meetings.
* Supporting member’s which encounters with bottleneck.

**Key Phases and Timeline:**

**AI : AI Development – Ali Emre Yenihayat, Berk Ülker**

**GD : Game Development – Onur Erçen, Umut Baran Boztaş**

**INT : Integration – Onur Erçen**

**Resource Planning:**

* Python (TensorFlow, PyTorch…) (AI Training Environment)
* Unity (Game Engine)
* Github
* GIMP (Photoshop Program)
* Aesprite (Design Program)
* Steam (Communication)

**Risk Management:**

* AI training might delay the project. – More efficient algorithms can be used.
* There may be unexpected bugs in Unity. – Other members can support game development and some research may be done.
* Integration tests might not be done because of other delays. – Integrators can be appointed to other tasks to minimize the delay.

**Communication Plan:**

Every Tuesday and Thursday, there will be a scrum meeting and every Sunday, there will be a sprint meeting on Steam.

**Change Management Plan:**

* Change demands will be noted to Github.
* Team will evaluate the changes’ effect on project.
* Major changes will be applied after approved.
* Applied changes will be updated on documents and code.

**Budget Plan:**

Application Licences (Unity, PyCharm, Aseprite) to develop game environment, train AI and design assets.

Labor and time.

**Effort Estimation for Each Phase**

We are designing a game called **Demand Øf Momentum** with a four-people team: **Ali Emre Yenihayat, Berk Ülker, Onur Erçen, and Umut Baran Boztaş**.

Ali Emre and Berk will work on AI development by using : **Python, PyTorch, Pandas, GIMP, Aseprite** and **SensorFlow.**

Onur and Umut Baran plan to design the game environment by using **Unity**.

The game will feature a map where players race against AI bots trained with **Deep Reinforcement Learning**.

Our target audience is **12 to 24-year-old individuals**.

For effort estimation, we used an **analogy-based approach** based on similar projects.

The **requirements gathering** will take approximately 5 person-days, where we define our game mechanics, AI requirements, and the map structure.

The **game environment design** will last about 25 person-days, which include map creation, modeling, and implementation of basic mechanics.

The **AI bot development** will require 30 person-days, where Deep RL models will be trained and optimized.

The **data processing and AI optimization** is expected to last about 10 person-days to improve the model’s performance.

The **integration and testing phase** will take around 20 person-days to ensure AI bots interact correctly with the game environment.

The **final tests and deployment** phase will require around 10 person-days, covering bug fixing and performance improvements.

For **project management**, we allocate 10 person-da**ys** for daily coordination and progress tracking.

The **total estimated effort** is considered to be about **110 person-days**, which means the project must be completed in **27-28 days** with our four-people team.

With multiple phases in parallel, our target is to finish within our **25-30 day deadline**.

**Estimated Effort Per Task**

* **Requirements Gathering** → **5 person-days** (All team members)
* **Game Environment Design (Unity)** → **25 person-days** (Onur & Umut Baran)
* **AI Bot Development (Deep Reinforcement Learning, Python, PyTorch)** → **30 person-days** (Ali Emre & Berk)
* **Data Processing & AI Optimization** → **10 person-days** (Ali Emre & Berk)
* **Integration & Testing** → **20 person-days** (All team members, mainly Onur & Umut Baran for Unity, Ali Emre & Berk for AI)
* **Final Adjustments & Deployment** → **10 person-days** (All team members)
* **Project Management & Coordination** → **10 person-days** (All team members, rotating responsibility)
* **Total Effort Per Person**

| **Team Member** | **Main Responsibilities** | **Estimated Effort (Person-Days)** |
| --- | --- | --- |
| **Ali Emre Yenihayat** | AI Development with Python, Data Processing, Testing | ~25 PD |
| **Berk Ülker** | AI Development with Python, Data Processing, Testing | ~25 PD |
| **Onur Erçen** | Unity Development, Integration, Testing | ~30 PD |
| **Umut Baran Boztaş** | Unity Development, Integration, Testing | ~30 PD |

**Total Project Effort – Demand of Momentum**

To calculate the **total project effort**, we sum up the effort estimations for every phase:

* **Requirements Gathering** → **5 PD**
* **Game Environment Design (Unity)** → **25 PD**
* **AI Bot Development (Deep RL, Python, PyTorch)** → **30 PD**
* **Data Processing & AI Optimization** → **10 PD**
* **Integration & Testing** → **20 PD**
* **Final Adjustments & Deployment** → **10 PD**
* **Project Management & Coordination** → **10 PD**

**Total Effort Calculation:**

**5 + 25 + 30 + 10 + 20 + 10 + 10 = 110 Person-Days**

**Task Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks \ Members | Ali Emre Yenihayat | Berk Ülker | Onur Erçen | Umut Baran Boztaş |
| Project Scope | X | X |  |  |
| Project Organization | X | X | X | X |
| Objectives |  | X |  |  |
| Key Phases and Timeline |  | X | X |  |
| Resource Planning | X | X | X | X |
| Risk Management | X | X |  |  |
| Communication Plan |  | X |  |  |
| Change Management Plan | X | X | X | X |
| Budget Plan |  | X |  |  |
| Effort Estimation for Each Phase | X | X | X | X |
| Estimated Effort Per Task | X |  | X |  |
| Total Project Effort |  |  | X |  |