**RS**

RICE SHOWER

***Thieves Racer***

***TECHNICAL DESIGN DOCUMENT***

**SUMMARY**

1. **Project goal**
2. **Innovation**
3. **Provided Services (Beside the Game)**
4. **Client side**
   1. Hardware Requirements
   2. Software Requirements
5. **Workload Estimation**
6. **General Architecture**
   1. Frontend
      1. Hardware
      2. Software
      3. Services
   2. Backend
      1. Hardware
      2. Software
      3. Services
   3. Workload Capacity
7. **Connection**
   1. Global Infrastructure
   2. Network Requirements
8. **Delivery**
   1. Estimated Delivery Time
   2. Delivery Platform
9. **Cost Estimation**

**TEAM MEMBERS**

|  |  |
| --- | --- |
| **A person wearing sunglasses  Description automatically generated with low confidence** | **Lin Liya**  Game Design  UI Design  Gameplay programmer  [liya.lin@studenti.unimi.it](mailto:liya.lin@studenti.unimi.it) |
| **A person wearing a hat and holding a cell phone  Description automatically generated with low confidence** | **Aragon McAyrton**  Art  Video editing  AI  [mcayrton.aragon@studenti.unimi.it](mailto:mcayrton.aragon@studenti.unimi.it) |
| **A person wearing glasses  Description automatically generated with medium confidence** | **Chen Xiao Oscar**  Game Design  Technical Design  Programming  [xiaooscar.chen@studenti.unimi.it](mailto:xiaooscar.chen@studenti.unimi.it) |

**1. Project goal**

**Online multiplayer**

Beside the normal offline mode, the game allows a maximum of fourplayers to be connected and play the co-op multiplayer mode. Each player’s progress is saved locally on their phone, the server will only manage the coordination of the game and must be always running. In case of maintenance, the server will be shut down and it’s not possible to play the game.

**2. Innovation**

The project provide customers/players to play the game together while one of them host the game.

Of course the game is playable offline.

The game aim for people who are friends, it is available as 2 for now.

The connections uses a KCP Transport.

The max connection is 2 player.

Graphical user interface, text

Description automatically generated

This may not be the ultimate values.

Why KCP?

 100% C#.

 Works on all platforms except WebGL.

 Heavy test coverage (83.5%).

 Extremely fast.

 Extremely simple.

 Nearly allocation free\*.

"KCP is a fast and reliable protocol that can achieve the transmission effect of a reduction of the average latency by 30% to 40% and reduction of the maximum delay by a factor of three, at the cost of 10% to 20% more bandwidth wasted than TCP. It is implemented by using the pure algorithm, and is not responsible for the sending and receiving of the underlying protocol (such as UDP), requiring the users to define their own transmission mode for the underlying data packet, and provide it to KCP in the way of callback. Even the clock needs to be passed in from the outside, without any internal system calls."

**3. Provided Services (beside the game)**

* **Customer support:** we’ll provide the player the assistance about technical issues like account management, device configurations and platform compatibility.
* **Forum:** we’ll create a place where the players can interact and help out between them.
* **Payment system:** The game contains in-app purchases and we want to make sure the player can make transactions successfully.

**4. Client side**

**4.1 Hardware Requirements**

**4.2 Software Requirements**

**5. Workload Estimation**

**Go watch gdd**

**6. General Architecture**

**6.1 Frontend**

**6.1.1 Hardware**

**6.1.2 Software**

**6.1.2 Services**

**6.2 Backend**

**6.2.1 Hardware**

**6.2.2 Software**

**6.2.3 Services**

**6.3 Workload Capacity**

**7. Connection**

**7.1 Global Infrastructure**

**7.2 Network Requirements**

**8. Delivery**

**8.1 Estimated Delivery Time**

**8.2 Delivery Platform**

**9. Cost Estimation**