

# Réalisation d'un jeu vidéo et de son serveur portable

Daniel Roulin & Fabio Dubath

## Le problème

- Jouer à plusieurs sans Internet
- Nouveau type de multijoueur

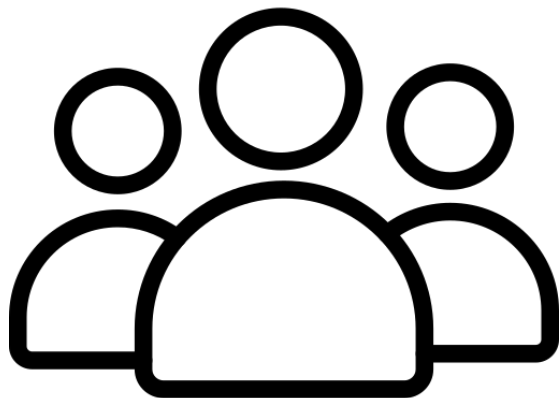


# Objectifs initiaux



- Création d'un serveur portable
  - Raspberry Pi
  - Batterie
  - Émission d'un réseau Wi-Fi
  - Hauts-parleur, LEDs, boutons
  - Boitier imprimé en 3D
- Développement d'un jeu
  - Style arcade
  - Partie rapide
  - Conception de sorts magiques
  - Jeu mobile
- Open Source

# Répartition initiale du travail



- Création d'un serveur portable
  - Raspberry Pi
  - Batterie
  - Émission d'un réseau Wi-Fi
  - Hauts-parleur, LEDs, boutons
  - Boitier imprimé en 3D
- Développement d'un jeu
  - Style arcade
  - Partie rapide
  - Conception de sorts magiques
  - Jeu mobile
- Open Source

**Daniel**

**Fabio**

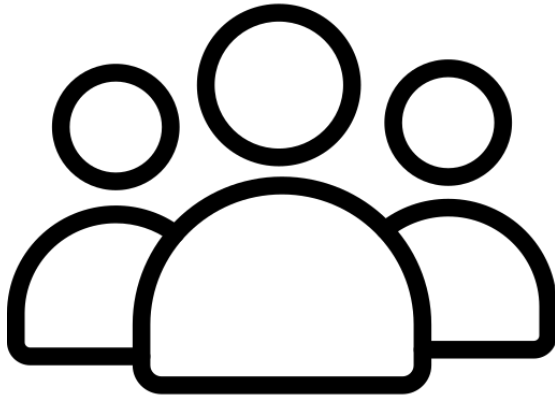
# Objectifs finaux



- Création d'un serveur portable
  - Raspberry Pi
  - Batterie
  - Émission d'un réseau Wi-Fi
  - Hauts-parleur, LEDs, boutons
  - Boitier imprimé en 3D
- Développement d'un jeu
  - Style arcade
  - Partie rapide
  - Conception de sorts magiques
  - Jeu mobile
- Open Source



# Répartition finale du travail



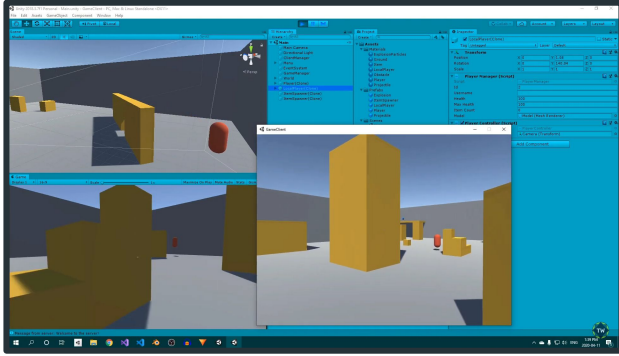
- **Daniel:**

- Serveur
  - Démarrage automatique
  - Réseau wifi
- Jeu
  - Logique
  - Networking

- **Fabio:**

- Jeu
  - Graphisme
  - Logique
  - Networking

# Démarche



+ Animations

+ Paquets + Logique

TM

+ UI + Scripts

+ Graphismes

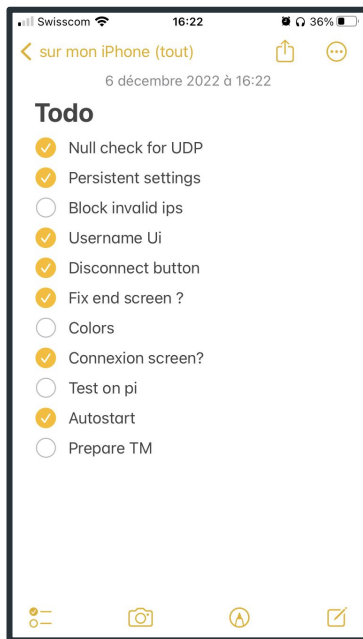
- Tutoriel de Tom Weiland
  - Réseau
  - 3D
  - Unity Server

Architecture

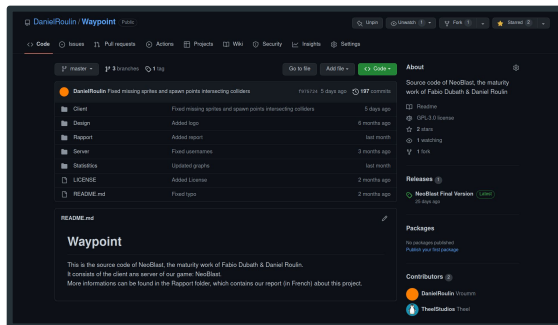
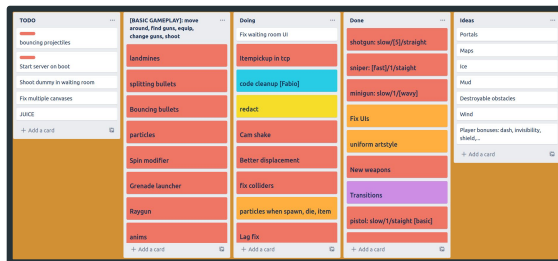
Notre projet

# Organisation

## Notes



## Trello



## Github

## WhatsApp





# Fonctionnement



```
public bool Hit(Player _by)
{
    if (invincibilityTimer > 0 || isRespawning) return false;
    if (!Server.gameStarted) return false;
    if (_by.id == id) return false;

    ServerSend.PlayerHit(this, _by);
    respawnTimer = respawnTime;
    isRespawning = true;

    return true;
}
```

Player.cs

# Serveur

```
foreach (Client client in Server.clients.Values)
{
    if (client.player != null)
    {
        if (collider.CheckCollision(client.player.collider))
        {
            if (client.player.Hit(owner))
            {
                Destroy();
                Return;
            }
        }
    }
}
```

Projectile.cs

```
public static void PlayerHit(Player _player, Player _by)
{
    using (Packet _packet = new Packet((int)ServerPackets.playerHit))
    {
        _packet.Write(_player.id);
        _packet.Write(_by.id);

        SendTCPDataToAll(_packet);
    }
}
```

ServerSend.cs

# Client

```
public static void PlayerHit(Packet _packet)
{
    int _id = _packet.ReadInt();
    int _by = _packet.ReadInt();

    GameManager.players[_by].kills += 1;
    GameManager.players[_id].Hit();
    GameManager.instance.leaderboard.IncreaseKillCount(_by);
}
```

ClientHandle.cs

```
public static Dictionary<int, PlayerManager> players = new Dictionary<int, PlayerManager>();
```

GameManager.cs

```
public void Hit()
{
    gameObject.SetActive(false);
    Debug.Log($"Player {id} got hit, ouch!");
}
```

PlayerManager.cs

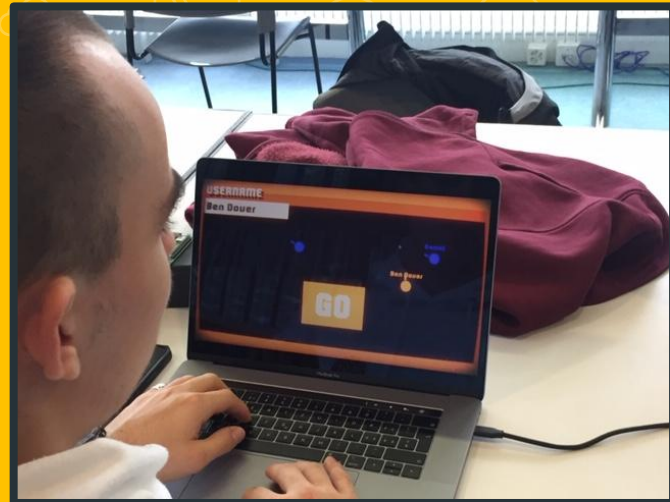


## PLAYER IS HIT



# DEMONSTRATIONS





# CRITIQUE POST-TEST

## Cyriel 8/10



Bonne mécanique  
Simplicité et clarté



Indication cooldown  
Spawn sans armes

## Hendrick 8/10

Amusant  
Convivial

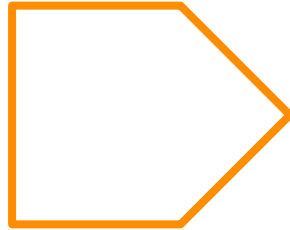
Spiroll, Lag, Maps  
Spawn armes

## Alam 8/10

Facile à comprendre  
Compétitif

Plein écran, Lag  
Contraste

# GAMEPLAY



## DEMO



# 01

## WIFI LOCAL

Se connecter au wifi  
"Waypoint"

# 02

## LANCER LE JEU

Ouvrir "neoblast.exe"

# 03

## CONNECTION

Dans les options, verifier que  
l'IP soit "192.168.4.1" et jouer

# **EVENEMENTS RÉCENTS**

Le scandale Unity et les  
alternatives



**Unity**®



**UNREAL  
ENGINE**



**THE END**