

## ***Rapport de soutenance 1***

### Sommaire

<a href="#">1 / Rappel du projet</a> .....	1
<a href="#">2/ Rappel du cahier des charges</a> .....	2
<a href="#">3/ Taches individuelles et ressentis</a> .....	2
<a href="#">3.1/ Joric HANTZBERG</a> .....	2
<a href="#">3.1.1/ Multijoueur</a> .....	2
<a href="#">3.1.2/ Barre de vie</a> .....	4
<a href="#">3.1.3/ Autres implémentations minimales</a> .....	6
<a href="#">3.1.4 / Ressentis</a> .....	6
<a href="#">3.2/ Arthur Gallier</a> .....	6
<a href="#">3.2.1/ Système de compte</a> .....	6
<a href="#">3.2.2/ Site web</a> .....	7
<a href="#">3.2.3/ Système d'inventaire</a> .....	8
<a href="#">3.2.4 / Ressentis</a> .....	9
<a href="#">3.3 / Raoul Olaru</a> .....	9
<a href="#">3.3.1/ Déplacements, attaques et animations</a> .....	9
<a href="#">3.3.2/ BREAKING POT !</a> .....	11
<a href="#">3.3.3/ Reprise du design du niveau et collisions</a> .....	12
<a href="#">3.3.4/ Design des armes, items et animations</a> .....	13
<a href="#">3.3.5/ Ressentis :</a> .....	15
<a href="#">4/ Avancement du projet</a> .....	16
<a href="#">5 / Conclusion et perspective</a> .....	17
<a href="#">6 / Source</a> .....	17

### 1 / Rappel du projet

Notre projet de deuxième semestre intitulé « The legendary world : Eldorado » est un jeu 2D de type MMORPG (Massive Multiplayer Online Rôle Play Game). Le but du jeu est d'explorer le monde et d'améliorer soon personnage en combattant différents monstres. Pour ce faire, le joueur va pouvoir utiliser différentes armes qu'il récupèrera pendant son aventure.

## 2/ Rappel du cahier des charges

Pour rappel, ce jeu contiendra plusieurs fonctionnalités importantes. On retrouve parmi celle-là : le multijoueur, les maitrises qui permettent d'améliorer les statistiques du joueur, les coffres et les marchands qui permettent de récupérer de nouvelles armes ou encore des monstres à combattre munis d'une intelligence artificielle.

Répartition des tâches et planning d'avancement :

- Début de création de la map (environ 25%) => Raoul et Joric
- Mouvement et animations du personnage => Raoul
- Attaque à l'épée du personnage => Raoul
- Système de multijoueur permettant aux utilisateurs de se rejoindre sur une même partie => Joric
- Création d'un inventaire pour stocker les items => Arthur
- Barre de statistique du joueur => Joueur

## 3/ Taches individuelles et ressentis

### 3.1/ Joric HANTZBERG

#### 3.1.1/ Multijoueur

Durant cette première partie de projet, j'ai eu à ma charge de réaliser une partie du multijoueur, la partie qui consiste simplement à permettre à deux joueurs de se retrouver sur la même partie. Cette tâche peut paraître simple dans un premier abord mais elle est enfaite assez complexe. En effet cela implique d'apprendre un toute une nouvelle partie de la programmation : le réseau. Pour réaliser cela, j'ai donc choisi d'utiliser photon. C'est un package Unity qui permet d'avoir un petit serveur de 20 places qui peut accueillir du multijoueur.

Dans un premier temps, j'ai créé l'interface de connexion, un simple bouton <<Play>> mais pas si simple qu'on ne pourrait le croire.

En effet, l'un des problèmes rencontrés avec photon est l'absence de <<room persistante>>. C'est-à-dire que une fois que tous les joueurs sont partis de la partie, le serveur s'arrête. Pour pallier ce premier problème, il a fallu que je mette une petite procédure en place dont je vais maintenant décrire les quelques étapes.

- 1- Le joueur appuie sur le bouton Play (Voir figure 1 pour le code)
- 2- Le joueur se connecte au serveur principal puis au lobby (une zone dans laquelle le joueur attend pour se connecter à la partie)
- 3- Le programme essaye de créer une partie « Master », si elle existe déjà, la fonction « OnCreateRoomFailed » est appelée et cette fonction va faire rejoindre la partie «Master» au joueur.
- 4- Ainsi, peu importe qu'il y ait déjà des joueurs ou non, le joueur qui essaye de se connecter va bien arriver sur la partie.
- 5- Ensuite, il y a un petit écran « Loading » le temps que la connexion se fasse pour que l'utilisateur n'appuie pas frénétiquement sur le bouton Play pour essayer d'accélérer le processus.
- 6- Pour finir, une instance de joueur est créé sur la map (voir figure 2)

```
public class MainMenu : MonoBehaviourPunCallbacks
{
    public void PlayGame()
    {
        PhotonNetwork.JoinRoom("Master");
    }
    public override void OnJoinRoomFailed(short returnCode, string message)
    {
        PhotonNetwork.CreateRoom("Master");
    }
    public override void OnCreateRoomFailed(short returnCode, string message)
    {
        SceneManager.LoadScene("MainMenu");
    }
    public override void OnJoinedRoom()
    {
        PhotonNetwork.LoadLevel("Level 1");
    }

    public override void OnConnectedToMaster()
    {
        PhotonNetwork.JoinLobby();
    }

    public void QuitGame()
    {
        Application.Quit();
    }
}
```

Figure 1 : Code du bouton Play

```
public GameObject playerPrefab; // Player_1
// Event Function
void Start()
{
    Vector2 vect = new Vector2(0, 0);
    GameObject instance = PhotonNetwork.Instantiate(playerPrefab.name, (Vector3)vect, Quaternion.identity);
    Camera child = instance.GetComponentInChildren<Camera>();
    child.enabled = true;
}
```

Figure 2 : Script d'instanciation des joueurs sur le serveur



Figure 3 : Menu de lancement du jeu

### 3.1.2/ Barre de vie

Dans cette partie, je vais parler de l'UI. UI est l'abréviation anglaise de user interface. Ces interfaces sont utilisées pour donner des informations aux joueurs ou leur permettre d'interagir avec son environnement.

La deuxième tâche à laquelle je me suis attelé est la barre de vie car en effet, tout bon jeu se doit d'avoir une UI propre et claire. Cette barre de vie ne contient en réalité pas que la vie. C'est en fait un groupe de 3 barres représentant respectivement : la vie, le mana et l'expérience acquis pour le niveau en cours. Pour finir, on peut également voir son niveau actuel en haut de ces barres de statistiques (voir figure 4).



Figure 4 : Barres de statistiques

Ces différentes barres utilisées sont remplies avec des images de type « filled ». Ce qui permet d'avoir un champ scale (un champ qui est échelonné) avec pour valeur maximum la valeur maximum de la statistique en question et en « fillAmount » la valeur actuelle de cette même statistique. Ceci rend très simple son utilisation comme le montre le script associé ci-dessous (voir figure 5) :

Joric HANTZBERG

Arthur GALLIER

Raoul OLARU

```
public class PlayerStatistics : MonoBehaviour
{
    public int money; // Unchanged
    public int level = 1; // Unchanged
    public LvlNumber lvl; // StatusPattern (LvlNumber)

    public int maxHealth = 10; // Unchanged
    public int currentHealth; // Unchanged
    public HealthBar healthBar; // HealthBar (HealthBar)

    public int maxMana = 200; // Unchanged
    public int currentMana; // Unchanged
    public ManaBar manaBar; // ManaBar (ManaBar)

    public int maxXp = 3000; // "2000"
    public int currentXp; // Unchanged
    public XpBar xpBar; // XpBar (XpBar)

    // Start is called before the first frame update
    // Event function
    void Start()
    {
        currentHealth = maxHealth;
        healthBar.SetMaxHealth(maxHealth, add: 0);

        currentMana = maxMana;
        manaBar.SetMaxMana(maxMana, add: 0);

        currentXp = maxXp;
        xpBar.SetMaxXp(maxXp, add: 0);

        lvl.ChangeSprite(level);
    }
}
```

```
// Update is called once per frame
// Event function
void Update()
{
    healthBar.SetHealth(currentHealth);
    xpBar.SetXp(currentXp);
    manaBar.SetMana(currentMana);
}
```

Figure 5 : Scripte barres de statistiques

Dans ce script, la fonction start va initialiser les trois barres de statistiques

Ensuite la fonction update remplit les barres de statistiques toutes les secondes avec les données actuelles du joueur

On peut ainsi voir que l'implémentation de cet élément n'a rien de très compliqué. Cependant, il y a eu un petit problème à régler. La taille de cet élément est fixe et est en pixels. Ce qui veut dire que le rendu n'est pas le même suivant la résolution de l'écran. Pour parer ce problème, j'ai fait un petit script qui positionne l'élément dans le cadre et règle la taille en fonction de la résolution actuelle et de la résolution d'échelonnage. Cela est fait grâce à un petit produit en croix que l'on peut voir dans le script ci dessous.

```
public class StatusSizer : MonoBehaviour
{
    private RectTransform placement;
    private Resolution _resolution;

    // Event function
    void Start()
    {
        _resolution = Screen.currentResolution;
        placement = GetComponent<RectTransform>();
        placement.localScale = new Vector3((float)(0.5 * _resolution.width) / 1920, (float)(0.5 * _resolution.height) / 1080);
        placement.anchoredPosition = new Vector2((float)(250 * _resolution.width) / 1920, (float)(-175 * _resolution.height) / 1080);
    }
}
```

Figure 6 : Script d'adaptation à la résolution de l'écran

### 3.1.3/ Autres implémentations minimales

Pour ce début de projet, j'ai également implémenté plusieurs petites fonctionnalités ensuite reprises par mes camarades comme les déplacements, ou bien la création des tilepalettes (endroit regroupant les textures permettant la création du niveau) utilisées pour réaliser notre map.

### 3.1.4 / Ressentis.

J'ai beaucoup apprécié faire ce projet avec mes camarades. Pour l'instant les délais sont tenus et l'ambiance de groupe est agréable. Cependant la charge de travail est conséquente dû à l'importance du projet et notre petit groupe de trois personnes. Pour l'aspect programmation, je trouve cela très intéressant de devoir se débrouiller seul sur ce projet, cela nous apprend à chercher par nous-même et c'est très important car ce sera comme cela que nous devons travailler plus tard.

## 3.2/ Arthur Gallier

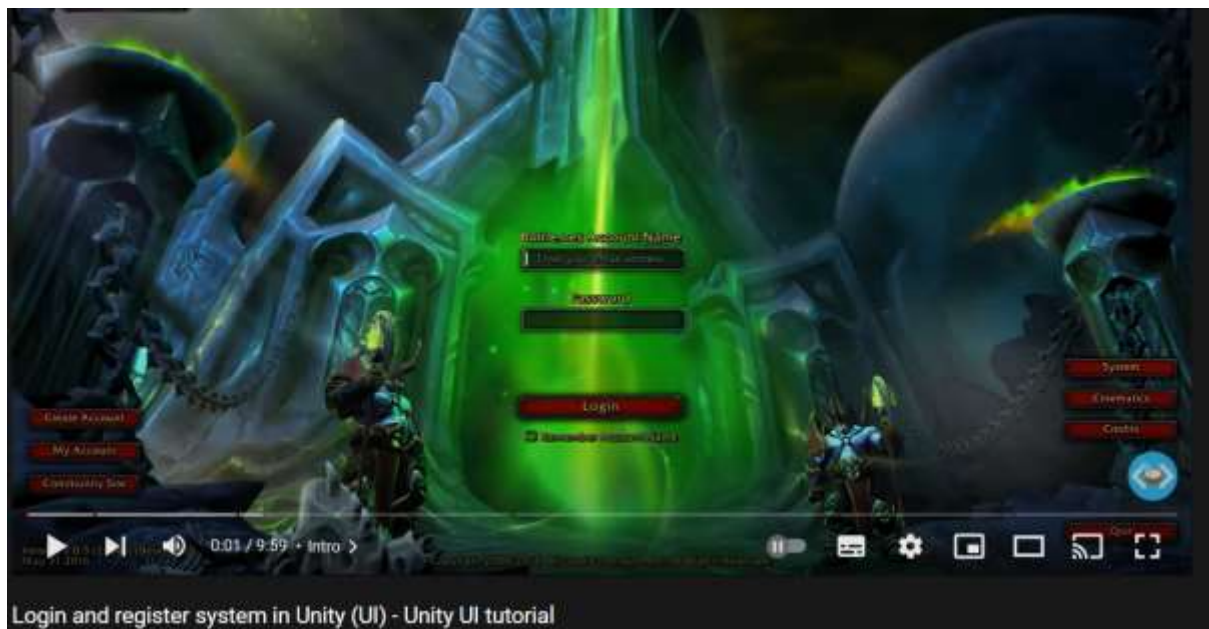
### 3.2.1/ Système de compte

Même si ce n'est pas encore mis au point, j'ai voulu commencer par ceci que je considère comme le plus important, le système de compte, qui n'est pas encore implémenté mais documenté.

C'est en effet ce qui va permettre de synchroniser les inventaires des joueurs ainsi que tout leurs items.

Il faut premièrement assurer un système de connexion, je me suis premièrement basé sur la vidéo de Coco Code(voir source), un anglophone qui explique simplement comment créer un "login", "register", "password forgotten" etc.

<https://youtu.be/PIA-4BUJfo0>



En complément des informations trouvées sur internet j'ai aussi regardé la vidéo de Brackeys

« <https://youtu.be/ZX8-1zodEFU> » qui explique plus en profondeur la gestion des comptes

Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU

### 3.2.2/ Site web

Le premier point était purement théorique mais là nous pouvons passer dans le concret puisqu'on va aborder le sujet du site web. Il n'est pas fini mais avancé.

Quand nous lançons le programme html nous tombons sur cette page et lorsque nous passons notre souris sur chacun des titres nous pouvons voir une animation avec des couleurs différentes faites en HTML et en Css.



Figure 1 : Accueil du site web

Le plus dur n'était pas le HTML comme nous pouvons le voir ci-dessous qui ne fait que 40 lignes



Joric HANTZBERG

Arthur GALLIER

Raoul OLARU

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mon site</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/styles.css">
  </head>
  <body>
    <nav>
      <ul>
        <li>
          <a href="anin.html">
            The legend of Eldorado
          </a>
        </li>
        <li>
          <a href="presentation.html">
            Présentation du projet
            <link rel="stylesheet" type="text/css" href="css/stylehub.css">
          </a>
        </li>
        <li>
          <a href="download.html">
            télécharger le pdf
          </a>
        </li>
        <li>
          <a href="liens.html">
            Liens
          </a>
        </li>
      </ul>
    </nav>
    <link rel="stylesheet" type="text/css" href="style.css">
  </body>
</html>
```

Mais c'est plutôt le css qui prend des centaines de lignes pour faire les animations car il faut gérer les modifications d'affichage des boutons à 10% de l'animation 20%, 30% ... Pour faire en clair on doit gérer l'affichage du bouton à 10% de l'animation , à 20% et ainsi de suite .

### 3.2.3/ Système d'inventaire

Si le site web n'est pas fini c'est parce qu'il fallait apporter du contenu sur notre jeu, en créant, par exemple, un inventaire.



Ce n'est que la version bêta de notre inventaire mais on y trouve déjà la plupart des fonctions implémentées : 24 cases où les inventaires peuvent se stocker, des boutons permettant de se rendre dans les autres onglets. Vous pouvez voir en haut à gauche un papier vous êtes en droit de vous demander ce que c'est.





C'est un papier qui nous informe sur le nom de l'item sur lequel on passe notre souris dessus, son prix ainsi que sa description. C'est ce qu'on appelle le "tooltip".

```
public void OnPointerEnter(PointerEventData eventData)
{
    GetThisItem();
    if(thisItem != null)
    {
        tooltip.ShowTooltip();
        tooltip.UpdateTooltip(GetDetailText(thisItem));
        RectTransformUtility.ScreenPointToLocalPointInRectangle(canvasRect, Input.mousePosition, null, out position);
        tooltip.SetPosition(position);
    }
}
```

C'est en effet ce programme qui nous permet d'afficher le « tooltip ».

### 3.2.4 / Ressentis.

Pour ce qui est du ressenti je me suis intégré à mon groupe même si je me sens quelquefois un peu oublié, l'ambiance est vraiment bien, donne envie de travailler.

Pour la gestion d'Unity elle est catastrophique. 1 fois sur 2 Unity ne veut pas sauvegarder et "push" aux autres ce que j'ai fait mais on me dit pourtant le contraire, je quitte donc Unity et quand je le relance, plus rien. J'ai perdu donc précieuses heures à devoir tout refaire, cependant le refaire une seconde fois m'a permis de mieux développer ce que je faisais.

## 3.3 / Raoul Olaru

### 3.3.1/ Déplacements, attaques et animations.

Pour les premiers jours de cette semaine de projet, j'ai eu à refaire les animations de déplacement de notre personnage, implémenter les animations de coups d'épée, donc d'attaque, et commencer à développer le système d'attaque.

Commençons par les animations de déplacement. Unity met à disposition des outils permettant l'implémentation assez simple d'animations, la seule difficulté étant de trouver un bon équilibre pour le changement d'image, pour rendre l'animation « fluide ».

Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU



Figure 1 : Image utilisée pour l'animation de déplacement.

```
public enum PlayerState
{
    walk,
    attack
}

public class PlayerMovement : MonoBehaviour
{
    public PlayerState currentState;
    public float speed;
    private Rigidbody2D myRigidbody;
    private Vector3 change;
    private Animator animator;
    private PhotonView view;

    // Start is called before the first frame update
    void Start()
    {
        view = GetComponent<PhotonView>();
        currentState = PlayerState.walk;
        animator = GetComponent<Animator>();
        myRigidbody = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    void Update()
    {
        if (view.IsMine)
        {
            change = Vector3.zero;
            change.x = Input.GetAxis("Horizontal");
            change.y = Input.GetAxis("Vertical");
            if (Input.GetButtonDown("attack") && currentState != PlayerState.attack)
            {
                StartCoroutine(AttackCo());
            }
            UpdateAnimationAndMove();
        }
    }
}
```

Figure 2 : Aperçu du script "PlayerMovement", gérant donc l'animation de déplacement.

Vient ensuite l'attaque, dans un premier temps, j'ai fait l'animation, en appuyant sur un bouton, l'animation d'attaque était lancée.



Figure 3 : Image utilisée pour les animations d'attaque.

### 3.3.2/ BREAKING POT !

Ensuite, il fallait rendre l'attaque fonctionnelle, et pour de simples tests, j'ai préféré faire un pot qui se casse quand on tape dessus. Évitant ainsi de devoir implémenter tout le système de monstre juste pour tester notre attaque. J'ai donc implémenté un pot, et son animation de destruction, quand on tape dessus.



Figure 4 : Le pot en jeu

J'ai rajouté dans le script PlayerMovement des lignes, dédiées à l'attaque. Mais il fallait aussi un script pour le pot ! Le voici :

Joric HANTZBERG

Arthur GALLIER

Raoul OLARU

```
public class pot : MonoBehaviour
{
    private Animator anim;

    // Start is called before the first frame update
    void Start()
    {
        anim = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {

    }

    public void Smash()
    {
        anim.SetBool("smash", true);
        StartCoroutine(breakCo());
    }

    IEnumerator breakCo()
    {
        yield return new WaitForSeconds(.3f);
        this.gameObject.SetActive(false);
    }
}
```

Figure 5 : Script du pot

Nous avons donc notre pot, il ne restait plus qu'à définir une zone autour du joueur, représentant la "hitbox" de l'épée, et donc quand un objet est dans cette zone et que l'animation d'attaque est lancée, cet objet se casse (s'il a été programmé pour se casser bien évidemment).

### 3.3.3/ Reprise du design du niveau et collisions

Joric avait fait les bases du monde, pour se familiariser avec l'usage de la "Tile Palette" (qui est une palette où tous les morceaux d'une carte sont divisés en 16 pixels par 16 pixels ce qui nous permet de les poser sur notre carte à nous), tâche que j'ai passé le reste de la semaine à faire. À première vue, on se dit que c'est une tâche faite rapidement, j'étais le premier à dire ça. Mais quand on commence à le faire, il faut prendre en main la "Tile Palette", et une fois fait, on se rend compte que tout n'est pas si rapide. Voulant avoir un beau rendu, il faut recommencer plusieurs fois certaines zones avant d'avoir un bon résultat qui nous plaît, chaque petite partie étant à faire manuellement, le temps passe très vite en le faisant.

Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU



*Figure 6 : Avancement du monde de notre jeu.*

### 3.3.4/ Design des armes, items et animations

Je faisais généralement ça le soir, en rentrant chez moi, chose qui prenait aussi mine de rien pas mal de temps. Premièrement les armes, j'ai commencé de design de quelques épées à mettre dans le jeu.



Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU



Les épées apparaissent ici en flou car elles ont été exportées en image 16x16, en jeu les items ne seront évidemment pas flous.

Qui dit nouvelle épée, dit aussi nouvelle animation, j'ai modifié l'image d'attaque pour que l'animation change en fonction de l'épée que le joueur à équiper.



*Image pour l'animation du katana (à revoir, car sur fond blanc, l'épée ne se voit pas).*

Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU



*Image pour l'animation de l'épée de feu.*

En plus de cela, j'ai également commencé à chercher, et faire moi-même des items, que plus tard nous implémenteront dans le jeu. En voici quelques exemples :



### 3.3.5/ Ressentis :

J'ai passé une bonne semaine, nous nous retrouvions tous les jours à EPITA pour travailler ensemble, l'ambiance de travail est très agréable, et nous avons les mêmes idées concernant l'avancée du jeu. Même si on a plus de travail, étant un groupe de 3, notre groupe en sera capable. Cette semaine a également permis une meilleure prise en main de Unity, nous faisant gagner du temps pour les



Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU

prochains jours de travail. Ma partie était un petit peu moins basé sur de la programmation « pure » comme la partie de Joric ou de Arthur, mais il faut aussi de ça pour un bon jeu, sans images on n'a rien ! J'ai pris énormément de plaisir à avancer ce projet durant cette semaine.

#### 4/ Avancement du projet.

Pour l'instant, le projet est dans les temps par rapport à ce qui était convenu dans le cahier des charges.

Taches réalisées :

Tâche	Chargé de développement	Avancement
Multijoueur	Joric	Les joueurs peuvent se rejoindre dans une partie
Mouvement et animation du personnage	Raoul	<b>Finis</b>
Attaque du personnage	Raoul	Le joueur peut donner des coup d'épée et casser des objets de type <<Breakable>>
Barres de statistiques	Joric	<b>Finis</b>
Création de la map	Raoul	La map est bien commencé (25%) , nous avons réussi a bien prendre en main l'outil de création

Les items sont également en cours d'implémentation ce qui nous permet d'avoir une légère avance sur le projet en cas de problème inattendu.

Pour la deuxième soutenance, les taches à réaliser sont :

Tâche	Chargé de développement	Avancement
Création de la map	Raoul	60% de la map est crée
Inventaire	Arthur	L'inventaire est fini
Design d'arme	Raoul	10 épées différentes sont créées (uniquement design)
Attaque à l'épée et à l'arc	Raoul	Attaque à l'épée finis et les attaques à l'arc sont commencés
Création des monstres	Raoul	Les monstres ont une IA et détectes les joueurs proches d'eux
Système de coffre	Arthur	Les coffres sont terminés
Système de compte et sauvegarde	Joric	Les joueurs se connecte au jeu avec un compte et les données

Joric HANTZBERG  
Arthur GALLIER  
Raoul OLARU

		du joueur sont sauvegardé sur son compte
Création des sons	Arthur	Création de bruitages pour le jeu
Site web	Arthur	<b>Finis</b>
Maitrises	Joric	Les maitrises sont en cours de développement

## 5 / Conclusion et perspective

Le projet avance comme le prévoyait nos estimations. Les délais ont été respectés même si compte tenu de notre nombre de 3, il est plus compliqué de faire un projet équivalent à des groupes de 4 et la charge de travail est importante. Nous espérons continuer sur cette lancée pour la suite du projet afin de réussir à finir notre projet dans les temps

## 6 / Source

Unity : <https://unity.com/fr>

Documentation Unity : <https://docs.unity.com>

Asset : <https://gif-superretroworld.itch.io>

Gui : <https://assetstore.unity.com/packages/2d/gui/simple-fantasy-gui-99451>

Multiplayer : <https://www.photonengine.com>

Doc photon : <https://doc.photonengine.com/en-us/realtime/current/getting-started/quick-start>

Beaucoup de forums, pages de docs et tuto youtube

Playlist avec pleins de vidéos utiles :

<https://youtube.com/playlist?list=PL4vbr3u7UKWp0iM1WlfrjCDTI03u43Zfu>

Quelques "assets" d'ici : <https://opengameart.org/content/zelda-like-tilesets-and-sprites>

téléporteurs : <https://youtu.be/0JXVT28KClg>