

Rapport de 2^e soutenance
Issidi
Debil.OS();

Jérémy BEUVRY
Julien BOULICAUT
Clément FINCK
Sébastien FLEURY

2 mai 2016

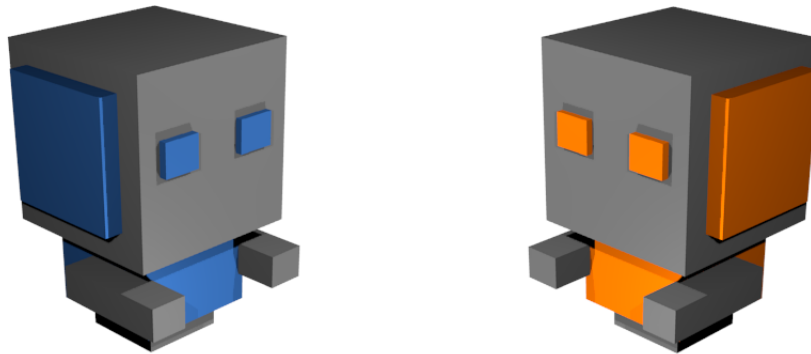


Table des matières

1	Introduction	4
2	Rappel du cahier des charges	5
2.1	Mécanique de jeu	5
2.2	Physique	5
2.3	Arme	5
2.4	Interface	6
2.5	Site Web	6
3	Répartition des tâches	6
4	Éléments déjà effectués	6
4.1	Interface	6
4.2	Physique	7
4.3	Particules	7
4.4	Personnage	7
4.5	Arme	7
4.5.1	Créations des modèles	7
5	Planning de la 2^e soutenance	8
6	Éléments effectués	8
6.1	Interface	8
6.2	Physique	8
6.3	Particules	8
6.4	Personnage	9
6.5	Arme	9
6.6	Créations des modèles	9
7	spécifications	9
7.1	Particules	9
7.2	Physique	10
7.3	Armes	10
7.4	Modèles	11
7.5	Interface	11
7.6	IA	12
7.7	Network	13

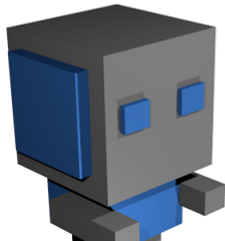
7.7.1	Adaptation des anciens scripts au reseau	13
7.7.2	Interpolation	13
7.7.3	Tir du joueur	14
7.7.4	Création des personnages	14
7.8	Autres	16
7.8.1	ColorModifier	16
7.8.2	Stats	16
7.8.3	Site web	16
7.8.4	Camera	16
7.8.5	Son	17
8	Planning de la 3^e soutenance	18
9	Récit de la réalisation	18
9.1	Issidi.shost.ca	18
9.2	animation	19
9.3	IA	19
9.4	Map	19
9.5	Réseaux	20
10	Conclusion	20

1 Introduction

Notre groupe, les "Debil.OS();" est fier de vous présenter son second rapport de soutenance du projet Issidi.

Ce projet prend la forme d'un jeu de tir à la troisième personne dans un univers post apocalyptique. Les principal mécaniques du gameplay et l'univers sont rappeler

Issidi



Project

2 Rappel du cahier des charges

2.1 Mécanique de jeu

Issidi est un jeu de tir à la troisième personne (aussi appelé *Third Person Shooter*), qui sera créé avec le moteur de jeu Unity3d, édition personnelle.

Pour les déplacements en plus de ceux classiques vous pourrez également esquiver à l'aide de petit saut rapide vers l'avant ou encore utiliser des doubles sauts qui permettront aussi d'atteindre certaines zones qui auraient été inaccessibles sinon.

Plusieurs armes seront disponibles pour varier le gameplay et permettre à tous les types de joueurs de prendre un maximum de plaisir. Que vous adoriez tout faire exploser avec une lance rocket, ou un rythme plus rapide avec un fusil d'assaut, même les fans de corps à corps pourront même s'essayer à la vivisection laser sur robots. ainsi que différentes cartes.

Le mode multijoueurs mettra en scène les batailles sanglantes entre pillards de ressources dont un seul sortira vainqueur, ou pas !

2.2 Physique

Cette section sera dédiée à la gestion des interactions des différents éléments avec la gravité. Ainsi que la gestion de tous les types de collisions : joueur/terrain, projectile/joueur, projectile/terrain (pour à terme, permettre d'avoir un environnement partiellement destructible). Parmi les interactions on notera en particulier la possibilité de marcher sur les murs et le plafond, via une rotation du vecteur gravité, qui constituera l'une des principales mécaniques du jeu.

2.3 Arme

Un robot sans défense n'a aucun intérêt s'il ne possède rien lui permettant d'occire ses ennemis, il doit donc disposer d'un arsenal suffisant. Il sera donc nécessaire de créer plusieurs armes afin d'ajouter une profondeur au gameplay.

2.4 Interface

Afin de rendre le jeu plus convivial, celui-ci doit posséder une interface composée de plusieurs menus. Celle-ci permettra notamment de naviguer dans les options du jeu, de lancer une partie en mode solo, ainsi que de choisir les paramètres des parties du mode multijoueur.

2.5 Site Web

Le site web nous permettra de montrer l'avancement de notre projet. Il contiendra une description du projet, des images de celui-ci, ainsi que des liens permettant d'avoir un accès rapide aux sources du projet.

3 Répartition des tâches

	Jérémy	Julien	Clément	Sébastien
Site Web		⊗		○
Physique			○	⊗
Multijoueur	⊗	○		
Animation		⊗	○	
IA		⊗	⊗	
Particules	○			⊗
Personnage	⊗			○
Arme	○			⊗
Son		○	⊗	
Modèles	⊗		⊗	
Interfaces		○	⊗	

Légende :

⊗ : s'occupe de

○ : aide

4 Éléments déjà effectués

4.1 Interface

- + Ajout du menu principal
- + Ajout d'un menu pause

- + création de la barre de vie

4.2 Physique

- + Ajout d'une boîte de collision pour le personnage, afin de lui permettre d'évoluer dans le monde
- + Modification de la gravité en fonction de son axe
- + Changement de l'axe du personnage lors de l'appui sur E .

4.3 Particules

- + Les balles sont suivit de particules, représentant la traînée.

4.4 Personnage

- + Possibilité de se déplacer
- + Échange possible de son axe de mouvement
- + Le personnage peut dasher (accélérer brutalement), impossibilité de changer de direction lors du dash
- + Double sauts possible
- + Caméra à la 3^e personne
- + Possibilité de se déplacer

4.5 Arme

- + lance-flamme
- + lance-rocket

4.5.1 Créations des modèles

- + Personnages
- + Armes variées
- + Point de respawn
- + Tourelles
- + Araignée suicidaire
- + Map prédécoupé

5 Planning de la 2^e soutenance

	Deuxième soutenance
Site Web	XXX
Physique	XXX
Multijoueur	XX
Animation	X
IA	X
Particules	XX
Personnage	XXX
Arme	XX
Son	X
Modèles	XX
interface	XX

Légende :

X : partie commencée

XX : partie à moitié faite

XXX : partie finie

6 Éléments effectués

6.1 Interface

- + refonte du menu principal
- + refonte du menu pause
- + création d'un menu de connexion en multijoueur
- + création de L HUD du joueur

6.2 Physique

- + L'obus utilise la gravité du personnage l'ayant lancé

6.3 Particules

- + Particules ajoutés lors de la destruction des balles.
- + Particules lors de l'annihilation des murs destructibles.

6.4 Personnage

- + Meilleure changement de gravité
- + La caméra ne passe plus à travers les murs.

6.5 Arme

- + Ajout d'un nom et des dégâts pour les armes.
- + Les balles font des dégâts à l'équipe adverse.

6.6 Créations des modèles

- + Refonte de certains modèles

7 spécifications

7.1 Particules

Si Michael Bay était un membre de notre projet, il serait fier de nos particules. La création de nouvelles armes, surtout le lance grenade, nous a poussé à chercher des effets un peu plus poussés que les premiers. Ceux-ci que nous avons dû manuellement adaptés aux réseau vu qu'ils n'étaient prévu que pour du single player. La partie la plus embetante des particules fut des les conserver jusqu'à leur mort naturelle. Pour ce faire, il a fallu separer l'objet, l'obus du generateur de particules en lui même, a l'impact faire que l'objet generateur de particules ne soit plus enfant de l'obus, arreter la generation de particules et enfin le supprimer plus tard.



7.2 Physique

La plus grosse partie de la physique était déjà disponible à la soutenance précédente, mais globalement, celle-ci est plus fluide mais surtout, est présente sur certaines munition. Dans un soucis de réalisme, car oui marcher sur les murs est réaliste, une gravité est maintenant présente sur les obus et cette gravité dépend du joueur, sur quel mur il est aligné.

7.3 Armes

En plus d'une refonte graphique, toutes les armes ont subi une refonte au niveau du code. Ainsi elles sont maintenant fonctionnelles en réseau. Le sniper, petit nouveau a cette soutenance dispose de son système de zoom, qui changera le FoV de la caméra. Le lance flamme maintenant complet inflige correctement les dégâts. Le lance grenade aussi, nouveau, s'ajoute au contenu du jeu. Les armes sont aussi maintenant dangereuses, les joueurs prendrons des dégâts des différentes munitions mais aussi et surtout, certains parties des cartes seront aussi destructibles et prennent des dégâts des armes.

7.4 Modèles

Tout comme le site, les modèles ont fait peau neuve. Toutes les armes ainsi que le personnage sont dotés de nouveaux modèles, en accord les uns avec les autres. Cette partie fut particulièrement longue pour respecter les critères graphiques du jeu, mais aussi pour que une sniper ressemble bien à un sniper, mais cubique.



7.5 Interface

Menu principal :

Nous avons finalement opté pour une charte graphique très simple basée sur le fait que les robots étant créés à partir de peu de moyen (le monde d'Issidi étant détruit) toutes les interfaces et même les retours fournis par le robot sont simplifiés à l'extrême. Notre refonte de toutes les interfaces déjà existantes et la création des nouvelles se basent sur l'utilisation des canevas et non plus du GUI.

Menu principal :

Ce menu permet de sélectionner les modes de jeu, et redirige vers les menus appropriés.

L'HUD :

L'interface de nos personnages est désormais composé de barres de vie et d'énergie qui fonctionnent sont toute deux des images qui utilise la très pratique propriété "filled" qui gère automatiquement le remplissage en fonction du ratio. Nous avons aussi l'indispensable jauge de munition, un affichage des scores qui n'est pas encore synchronisé au niveau du serveur, et une crosshair pour visé.

L'écran de mort

Pour l'écran de mort nous avons choisit d'afficher le pseudo écran de l'utilisateur du drone. Ainsi il voit donc un écran bleu lui permettant soit de créer un nouveau robot soit de quitter complètement l'application.

Le menu de sélection multi :

Ce menu est très important il permet de choisir son équipe et son arme et ainsi d'enclencher le processus de création du personnage.

7.6 IA

L'IA de notre jeu est pour le moment rudimentaire mais celle-ci s'annonce plutôt intéressante. Pour le moment seul deux ia sont développés et fonctionnels à leurs niveaux, la tourelle et la mine téléguidée. La tourelle utilise un système de raycast en direction du joueur ainsi que d'une zone de détection afin de savoir si celui-ci est perçu par la tourelle. Le point fort de cette méthode est que cela prend en compte la présence des murs qui seraient entre la tourelle et le joueur. Si une détection est faite alors la tourelle se tourne vers le joueur avec un délai de rotation permettant à celui-ci de se mettre à l'abri. Puis après un délai fixe la tourelle se met à faire feu sur le joueur. Par la suite si le joueur sort de la zone de vision ou de détection de la tourelle celle-ci arrête de tirer et se prépare de nouveau à le détecter. La mine, elle, ne prend en compte que la distance réelle entre elle et la totalité des joueurs. Elle prend alors le joueur le plus proche d'elle et en fait ça cible. Cependant si un autre joueur passe entre la mine et le joueur cible il devient alors la nouvelle cible. Elle n'est pour le moment pas capable de différencier les joueurs des 2 équipes. Une fois en collision elle détecte tous les joueurs dans une zone prédéfinie autour d'elle et leur inflige des dommages. Si celle-ci n'atteint pas de cibles avant un certain temps elle s'autodétruit et fait de même que si elle avait touché une cible.

7.7 Network

7.7.1 Adaptation des anciens scripts au reseau

Les anciens scripts étant conçus pour fonctionner uniquement en single-player, or lors de l'ajout du reseau, un problème est apparu : Si un personnage se déplaçait, en appuyant sur les touches directionnelles, les autres joueurs se déplaçaient aussi, ce qui n'est pas le comportement attendu. Afin de pallier à ce problème, nous avons du dériver tous nos scripts non plus de

`MonoBehaviour`

, qui propose une propriété intéressante :

`isLocalPlayer`

Ainsi, cela nous permet de désactiver tous les scripts qui n'appartiennent pas au joueur local en faisant :

`enabled = isLocalPlayer`

Cela à du être fait sur tous les anciens scripts, afin qu'ils fonctionnent en multijoueur.

7.7.2 Interpolation

Un problème que pose aussi le reseau, est le problème de lag (ou retard) qui peut arriver à tout moment dans une partie. Afin de ne pas voir les déplacements saccadés des autres joueur, nous avons créer un script d'interpolation, qui s'attache non pas uniquement au joueur, mais à toutes les entités dont nous voulons qu'elle apparaissent fluide en jeu (ce script fonctionne donc aussi sur les balles que tirent les personnages). Pour cela, nous avons utiliser la propriété

`hadAuthority`

`RigidBody`

`position`, afin de non plus rattrapper le "retard" a chaque frame, mais de pouvoir aussi prédire où le joueur vas aller, pour avoir une sensation en multijoueur d'un jeu plus réactif.

7.7.3 Tir du joueur

Un fonctionnalité importante en multijoueur est le tir (principalement sur les ennemis). Pour avoir des tirs de balles fonctionnelles, nous avons dû, en plus de faire

`Instantiate(bullet)`

est un préfabriqué de la balle à faire apparaître lors du tir, faire apparaître les balles sur le serveur, grâce à la commande suivante :

```
[Command]
void Cmd_InstantiateBullet(Vector3 position, Quaternion rotation, Vector3 up)
{
    GameObject bullet = (GameObject)Instantiate(bullet_template, position, rotation);
    if (!bullet)
        return;

    NetworkServer.Spawn(bullet);
}
```

L'ajout de

```
[Command]
```

par le serveur, et

```
NetworkServer.Spawn(bullet);
```

les clients connectés à la partie, au contraire de l'ancien système qui ne permettait l'apparition de la balle que sur le client ayant tiré celle-ci.

7.7.4 Création des personnages

Le fait de pouvoir changer de personnage à chaque mort étant un élément indispensable pour notre jeu afin de lui apporter plus de dynamisme, il a donc fallu avoir un système permettant ceci. Au contraire de la création des balles qui se faisait sur le serveur et dont celui-ci en avait l'autorité, la création des personnages doit se faire afin que le client voulu en ait l'autorité. Pour ce faire, il a fallu faire une commande qui crée le personnage et l'assigne au joueur de cette manière :

```
[Command]
public void Cmd_CreatePlayer(Stats.Team team, int weapon_index)
{
    GameObject character = (GameObject)Instantiate(character_and_weapons[weapon_index]);
    //Tous les test d'erreurs on été enlevés pour montrer cet exemple
    NetworkServer.ReplacePlayerForConnection(connectionToClient, character, player);
}
```



Dans cet exemple, team représente l'équipe que veut rejoindre le joueur, et weapon-index l'index représentant l'arme qu'il a choisit. character-and-weapons étant une liste de préfabriqué contenant à la fois le joueur, et l'arme associée. Ensuite, a la fin de cette fonction, nous récupéreront tous les

gameObject

tag de spawn (soit "WhiteSpawn" pour les spectateur, "OrangeSpawn" et "BlueSpawn"), et choisissent aléatoirement un des spawns, afin de placer le joueur sur celui-ci, grâce à :

```
spawns[Random.Range(0, spawns.Length)].GetComponent<Spawner>().Spawn(character);
//Encore une fois, tous les test d'erreurs on été enlevés pour plus de clareté.
```

7.8 Autres

7.8.1 ColorModifier

`{ColorModifier}`

le joueur est, il le colorie de la bonne façon, en récupérant tous les matériaux et en changeant les bons, de la façon suivante :

```
foreach (Transform child in transform)
{
    Material mat = child.gameObject.GetComponent<Renderer>().material;
    if (NeedChange(mat.color))
        mat.color = to_apply.color;
}
```

De plus, ce script a été conçu de façon à pouvoir être attaché à n'importe quel objet. Ainsi, cela nous permet de ne concevoir qu'un seul modèle, et de changer sa couleur, au lieu de devoir avoir un modèle pour chaque équipe. Cela nous permet donc de simplifier la création des personnages, ainsi que de leur gestion.

7.8.2 Stats

Ce script est un élément important du personnage : c'est lui qui gère la vie, l'énergie, les munitions du personnage. Ainsi, la plupart des scripts se réfère à lui pour savoir si le personnage peut tirer, dasher, ou tout simplement si le personnage est mort.

7.8.3 Site web

Avancement : A ce jour le site web est toujours en ligne à l'adresse : Issidi.shost.ca. Depuis la première soutenance le site web a subi une refonte complète dans le but de le rendre bien plus agréable à l'œil que sa précédente version. On peut bien entendu toujours avoir rapidement accès à toutes nos ressources.

7.8.4 Camera

La caméra a été retravaillée pour cette soutenance, afin de ne plus passer à travers les murs. Pour ce faire, nous récupéreront l'axe sur laquelle la caméra

doit se déplacer, et lançont un Raycast afin de voir la présence d'un mur. Si il y en a un, nous modifions la position de la caméra afin de placer celle-ci juste avant le mur.

7.8.5 Son

L'ajout du son lors des mouvements du personnage permet de donner une meilleure immersion dans le jeu. Afin d'ajouter cette partie imporante, nous avons créer une liste de sons de pas, prenons un son dans cette liste et le jouons losrque le joueur marche. Lorsque le son s'arrete mais le le joueur est toujours sensé marcher, nous prenons alors le son suivant et répétons cette procédure. Ainsi, le code à alors cette forme :

```
void Update ()
{
    if (!audio_source.isPlaying && deplacement.IsWalking())
    {
        //La gestion des erreurs a été enlevé pour plus de claireté
        audio_source.clip = footstep_sound[actual_sound];
        audio_source.Play();

        //Loop to next sound
        actual_sound = (actual_sound + 1) % footstep_sound.Count;
    }
}
```

8 Planning de la 3^e soutenance

	Troisième soutenance
Site Web	XXX
Physique	XXX
Multijoueur	XXX
Animation	XXX
IA	XXX
Particules	XXXX
Personnage	XXX
Arme	XXX
Son	XXX
Modèles	XXX
interface	XXX

9 Récit de la réalisation

9.1 Issidi.shost.ca

Toujours confronté à l'éternel problème de la page blanche même si celle-ci est restée virtuelle, nous avons tout de même en s'inspirant d'autre site réussi à obtenir un meilleur résultat que la première fois d'un point de vue visuel.

Interface

La réalisation de l'interface ne fut pas une tâche aussi aisée qu'elle n'aurait pu l'être pour une raison simple. Lors de la première réalisation la volonté de réellement "coder" quelque chose nous avait conduit à se tourner dans un premier temps vers le GUI qui est supposé être plutôt utilisé pour le debug, et n'aurait de toute façon pas été adapté pour l'implémentation du mode multijoueur par la suite. Nous nous sommes tout de même forcés à faire de l'UI qui, une fois qu'on arrête d'essayer de tout gérer par le code, se révèle plutôt efficace.

9.2 animation

La réalisation des animations nous a amené a nous confronter a blender logiciel que nous ne maîtrisions pas du tout. Et même si nous avons trouver de nombreux tutoriel qui expliquait comment faire. La présence de très nombreux raccourcis clavier pour changer l'angle de vue ou encore pour ouvrir les différents menus nous a beaucoup ralenti. D autre petite subtilité comme par exemple le fait de ne pas pouvoir rouvrir un fichier .fbx enregistrer par blender avec ce dernier.Nous ont bien ralenti, néanmoins nous commençons maintenant nous y habitué.

9.3 IA

L'IA a été très intéressante pour le moment à faire car il y a toujours un problème ou un cas auquel on n'a pas pensé qui ce produit. En effet lors du codage du script de la tourelle celle-ci faisait tout sauf ce que l'on lui demandait. En effet celle-ci se mettait à tourner sur elle-même sans jamais se fixer dès qu'un joueur était détecté. Quand celle-ci bougeait le référentiel n'était pas fixé et les coordonnées étant fixe la tourelle essayaient de rejoindre un point qu'elle ne pouvait rejoindre.

9.4 Map

Lors de la création de la map un problème est vite apparu : le trop grand nombre de setpasscall. En effet ceci est lié au nombre d'appel aux textures qui ont lieu sur la map. Le problème est que plus celui-ci est élevé plus le nombre d'image par seconde chute et moins le jeu est jouable. Le but était donc de réduire celui-ci au maximum. Le problème est que cela est très difficile de diminuer le nombre de setpasscall car nous essayons d'avoir un nombre d'entités suffisant pour que le décor soit destructible. De plus même si une texture est réutilisée, de nouveaux setpasscalls apparaissent. Par la suite nous n'avons pas réussi à faire une carte véritablement destructible car générant trop d'objet et donc de setpasscall. Notre carte solo ne sera donc un peu moins optimisé que celle en ligne pour le moment car celle-ci n'utilise pas beaucoup d'objet.

9.5 Réseaux

Unity est une formidable source de peines à propos du réseaux, où chaque mot de la documentation d'Unity est très important. Ainsi, la création du réseaux était un moment de tendre douleur, où ce qui marchait en mode un joueur, n,e marchait pas en mode multi. Une lecture attentive de toute la documentation est donc obligatoire.

10 Conclusion

Dans l'ensemble nous sommes très contents de ce qu'on a fait jusque la avec un gameplay très dynamique. le jeu marche en multi et est très fun , certaine personnes extérieur qui y ont joué se sont amusé aussi. Ces retours positif nous encourage donc a persévérer pour finir le jeu d'ici la dernière soutenance.