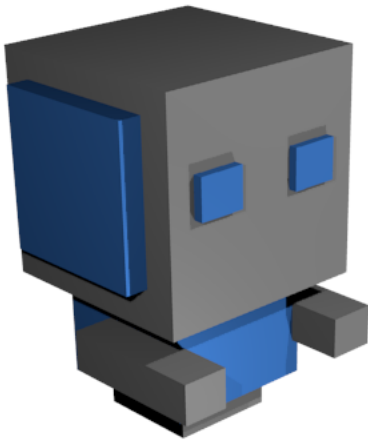


Rapport de 1<sup>re</sup> soutenance  
Issidi  
Debil.OS();

Jérémy BEUVRY  
Julien BOULICAUT  
Clément FINCK  
Sébastien FLEURY

14 mars 2016



## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Présentation du projet</b>	<b>5</b>
2.1	Scenario . . . . .	5
2.2	Mécanique de jeu . . . . .	6
<b>3</b>	<b>Découpage du projet</b>	<b>6</b>
3.1	Physique . . . . .	6
3.2	Mode un joueur . . . . .	7
3.3	Multijoueur . . . . .	7
3.4	IA . . . . .	7
3.5	Particules . . . . .	7
3.6	Personnage . . . . .	7
3.7	Arme . . . . .	8
3.8	Son . . . . .	8
3.9	Créations des modèles . . . . .	8
3.10	Interface . . . . .	8
3.11	Site Web . . . . .	9
<b>4</b>	<b>Répartition des tâches</b>	<b>10</b>
<b>5</b>	<b>Planning</b>	<b>10</b>
5.1	1 <sup>re</sup> soutenance . . . . .	10
5.2	Éléments effectués . . . . .	10
5.2.1	Physique . . . . .	10
5.2.2	Mode un joueur . . . . .	11
5.2.3	Particules . . . . .	11
5.2.4	Personnage . . . . .	11
5.2.5	Arme . . . . .	11
5.2.6	Créations des modèles . . . . .	11
5.2.7	Interface . . . . .	12
5.2.8	Site Web . . . . .	12
<b>6</b>	<b>Présentation des éléments effectués individuellement</b>	<b>12</b>
6.1	Personnage . . . . .	12
6.1.1	Déplacement . . . . .	12
6.1.2	Physique du personnage . . . . .	12
6.1.3	Double saut . . . . .	14
6.1.4	Dash . . . . .	14
6.2	Modèles . . . . .	14
6.2.1	Description . . . . .	14
6.2.2	Exemples . . . . .	15
6.3	Site Web . . . . .	16
6.3.1	Intérêt du site . . . . .	16
6.3.2	Structure du site . . . . .	16

6.3.3	Realisation du site . . . . .	17
<b>7</b>	<b>Planning de la 2<sup>e</sup> soutenance</b>	<b>17</b>
<b>8</b>	<b>Récit de la réalisation</b>	<b>17</b>
8.1	Issidi.shost.ca . . . . .	18
<b>9</b>	<b>Conclusion</b>	<b>18</b>

## 1 Introduction

Notre groupe, les "Debil.OS();" est fier de vous présenter le cahier des charges de son magnifique, que dis-je, génialissime projet : Issidi.

Les auteurs de ce projet sont quatre compères de première année animés d'une motivation à toute épreuve.

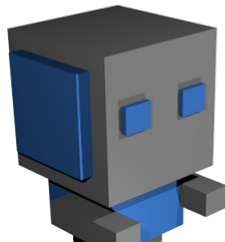
Dans ce document nous allons vous expliquer quels sont les moyens que nous comptons mettre en oeuvre pour en faire un succès.

Ce projet prendra la forme d'un jeu de tir à la troisième personne dans un univers post apocalyptique. Les différentes mécaniques du gameplay et l'univers seront bien entendu détaillés dans la partie "Découpage du projet".

Nous sommes tous très impatients de travailler sur ce projet d'une envergure bien supérieure aux divers Morpion, Racer, Sudoku, Packman que nous avons eu l'occasion de coder précédemment en Tp.

La réalisation d'Issidi sera aussi l'occasion de nous confronter à ce qui sera très probablement la réalité de notre futur métier d'ingénieur : le travail en groupe. En effet, même si nous avons déjà eu l'opportunité de travailler en groupe au Lycée ou pendant le premier semestre, l'ampleur du projet qui nous est demandés aujourd'hui exige plus d'efficacité, dans la répartition des tâches comme dans la résolution des conflits.

*Issidi*



Project

## 2 Présentation du projet

### 2.1 Scenario

Dans un monde ravagé par la guerre tous les hommes sont morts ou enterrés, mais pas nécessairement dans cet ordre. En effet, les radiations résultant du grand bombardement ont rendu la surface de la Terre létale pour l'humanité telle que nous la connaissons. Les rares survivants sont contraints de se terrer dans des bunkers antiatomiques pour pouvoir espérer qu'il y ait un jour lointain un retour possible à la surface. En attendant, seuls leurs drones peuvent se risquer au-dehors pour récupérer des matériaux quand il en reste. Très vite, les ressources se font rares et des luttes éclatent peu à peu. Incarnez un pilote de drone de combat en récupérant un maximum de matériaux nécessaires à la survie des camarades de votre abri et en déjouant les assauts des autres pillards assoiffés de métal. Vous représenterez l'ultime rempart entre la vie et la mort pour vos frères, alors, laisserez-vous l'humanité sombrer sans agir ?

## 2.2 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. A la différence des shooters traditionnels, plutôt horizontaux (souvent ramenés à un simple et large plan horizontal avec des obstacles), Issidi a pour but d'exploiter réellement les trois dimensions en permettant au joueur de marcher sur les murs, mais aussi sur le plafond rendant le jeu beaucoup plus dynamique !

Ainsi, les possibilités de déplacements du joueur seront démultipliées pour proposer une expérience unique. En plus des déplacements 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. Attention toutefois à ne pas oublier où se trouve le sol car un saut depuis un mur ou le plafond vous précipitera à terre et même si vous êtes solide, selon la hauteur, une telle chute vous sera sans doute fatal.

De plus, dans le mode solo, la caméra à la troisième personne vous permettra de mieux appréhender votre environnement ce qui vous permettra après réflexion de choisir votre chemin que ce soit : au plafond, sur les murs ou tout simplement au sol. Et ne croyez pas que ce sera une tâche aisée, votre environnement étant en ruine et partiellement truffé d'appareils en tous genres présent uniquement pour vous réduire en pièces détachées, toutes les routes ne seront pas empruntables soit parce qu'elles seront bouchées, soit à cause des diverses tourelles et autres joyeuses défenses automatiques qui vous bloqueront définitivement le passage, vous forçant à rebrousser chemin, vous forçant à rebrousser chemin dans l'espoir d'en trouver un autre dans ces dédales de débris.

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 !

## 3 Découpage du projet

### 3.1 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.

### 3.2 Mode un joueur

Le mode solo consistera à parcourir les ruines de vieilles usines abandonnées à la recherche de tout ce qui pourrait être utile à votre survie. Il faudra donc essayer de ne pas se faire tuer par les défenses automatiques encore actives et de se frayer un chemin parmi les décombres.

### 3.3 Multijoueur

Le multijoueur est un point important de ce projet. En effet, même s'il disposera d'un mode solo, l'intérêt principal d'Issidi résidera dans les affrontements épiques entre joueurs. Le réseau ne permettra donc pas seulement d'allonger la durée de vie du jeu mais aussi d'enrichir l'expérience de jeu. Il sera capable de regrouper jusqu'à "42" joueurs dans des modes de jeux divers et variés (death-match, capture de drapeau..).

### 3.4 IA

Nous aurons besoin de deux types d'Intelligence Artificielle(IA) dans Issidi. Premièrement, il faudra un système de visée pour les tourelles qui défendront les usines désaffectées. Deuxièmement, la création de pathfinding pour les systèmes de défense mobile sera aussi nécessaire (de gentilles araignées mécaniques dont le seul but sera de se faire sauter à votre proximité).

### 3.5 Particules

Afin d'améliorer la beauté de ce projet, l'ajout de particules est indispensable! On pourra par exemple faire apparaître la traînée des différents projectiles, afin de montrer leurs trajectoires. Ou encore de dynamiser l'action et d'égayer les décors.

### 3.6 Personnage

Comment jouer sans personnage? Cela semble en effet plutôt complexe, pour cela nous donnerons vie à de mignons petits robots qui pourront répandre la démocratie américaine. Les joueurs auront non seulement la possibilité de se déplacer sur le sol, mais aussi sur les murs ou le plafond du terrain. Une bonne gestion des déplacements et de la caméra est donc indispensable, dans un monde où l'on peut se retrouver face à un ennemi la tête à l'envers!

### 3.7 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.

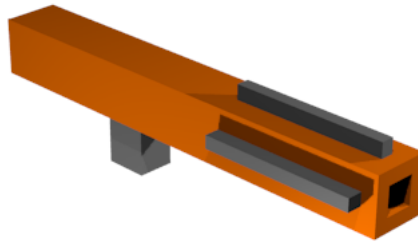


FIGURE 1 – Un lance roquette

### 3.8 Son

Que serait un jeu sans son ? Le but ici sera de créer une bande son simple mais originale.

### 3.9 Créations des modèles

Tout jeu digne de ce nom doit aussi avoir des graphismes à la hauteur ! C'est pourquoi cette partie consistera à la réalisation des modèles nécessaires afin de réaliser notre projet, et lui donner une ambiance qui le différenciera des autres jeux post apocalyptiques. Seront inclus la création des personnages ainsi que les environnements.

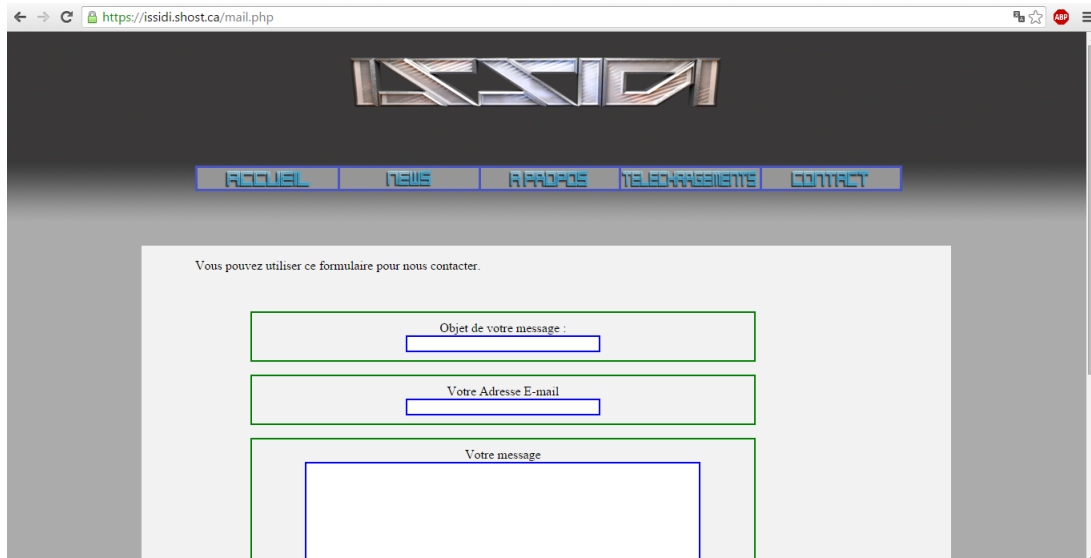
### 3.10 Interface

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.



### 3.11 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.



## 4 Répartition des tâches

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

Légende :

⊗ : s'occupe de

○ : aide

## 5 Planning

### 5.1 1<sup>re</sup> soutenance

	Premiere Soutenance
Site Web	XX
Physique	X
Multijoueur	
Animation	
IA	
Particules	X
Personnage	XX
Arme	X
Son	
Modèles	XX
interface	X

Légende :

X : partie commencée

XX : partie à moitié faite

XXX : partie finie

### 5.2 Éléments effectués

#### 5.2.1 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 (sera expliqué plus loin)

### 5.2.2 Mode un joueur

### 5.2.3 Particules

### 5.2.4 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<sup>e</sup> personne

### 5.2.5 Arme

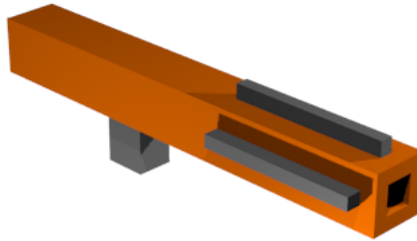


FIGURE 2 – Un lance roquette

### 5.2.6 Créations des modèles

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

### 5.2.7 Interface

### 5.2.8 Site Web

## 6 Présentation des éléments effectués individuellement

### 6.1 Personnage

#### 6.1.1 Déplacement

De part la nature de notre tps, où nous pouvons nous déplacer sur le sol, mais aussi les murs et le plafond, le déplacement du personnage est plus compliqué, car l'on ne peut associer une touche à l'un des axes (x, y ou z), mais chaque touche peut être associée à plusieurs axes au fil du temps, lorsque nous décidons de marcher sur un mur au lieu du sol par exemple. De plus, au lieu d'avoir un énorme enchevêtrement de condition pour gérer chaque cas possible de déplacement, la documentation d'Unity nous est venue en aide et nous a montré l'existence de :

```
Vector3 transform.forward  
Vector3 transform.right
```

qui désigne respectivement le vecteur directionnel frontal du transform et le vecteur directionnel droit du transform, et qui sont soumis aux transformations subites par notre personnage, donc ici les translations liées aux déplacements et les rotations liés aux changements d'axes et à la caméra (tourner la caméra de gauche à droite fait tourner le personnage). Ainsi, ce qui aurait été avant un gros code spaghetti c'est transformé en code simple. Il ne reste plus qu'à récupérer les événements utilisateur (l'appui des touches) grâce à :

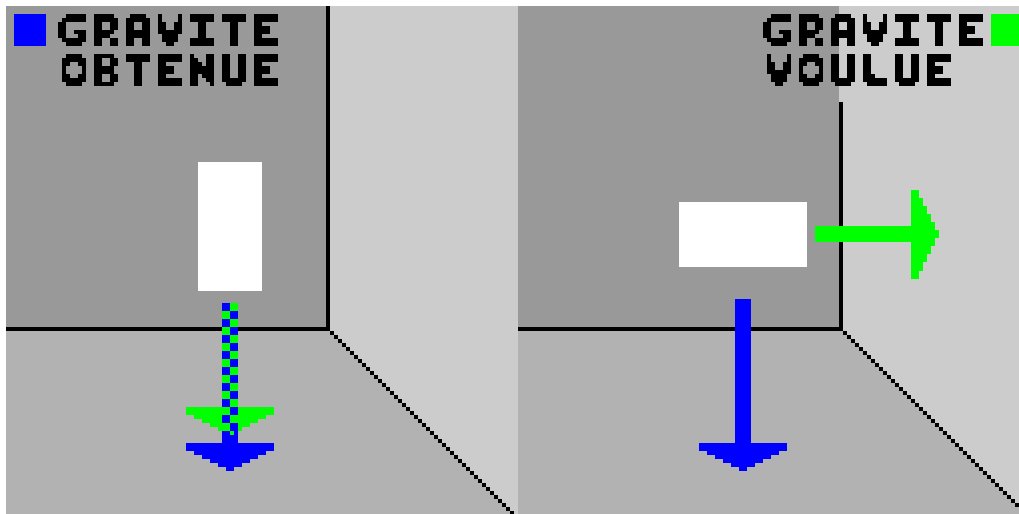
```
Input.GetAxis("Horizontal");//Déplacement gauche droite  
Input.GetAxis("Vertical");//Mouvement avancer/reculer
```

afin de pouvoir déplacer facilement notre personnage, quelque soit l'axe sur lequel il se trouve. ce qui donne donc

```
Vector3 vitesse_deplacement = Input.GetAxis("Horizontal") * transform.right +  
    Input.GetAxis("Vertical") * transform.forward;
```

#### 6.1.2 Physique du personnage

Afin d'ajouter la présence d'une gravité pour notre personnage, qui change si l'on change d'axe, il a fallu désactiver la gravité implémentée de base dans Unity. En effet, comme le montre le schéma ci-dessous :



lors d'une rotation du personnage (quand l'on marche sur le mur par exemple), la gravité reste toujours la même, c'est-à-dire sur l'axe des y, or nous voudrions que celle-ci s'adapte en fonction du joueur. C'est pourquoi nous avons décidé de refaire une gravité simple, qui est exprimé comme cela :

```
Vector3 vitesse_gravite_personnage = gravity * direction * time;
```

où gravity est un flottant, représentant la force de la gravité, direction un Vector3 représentant le transform.up du personnage (contient un 1 ou -1 sur l'un des axes et 0 sur les autres) (il s'agit d'une autre variable pour des raisons qui seront expliquées plus tard), et time représente le temps passé depuis la dernière frame. Le fait que nous calculons uniquement la vitesse, en non la nouvelle position du joueur s'explique par le fait que lors de l'ajout d'une boîte de collision :

CollisionBox

le moteur physique d'Unity ne fonctionne pas si l'on déplace directement le personnage d'une position à une autre. Il à donc été choisi de non plus modifier la position du joueur, mais sa vitesse, en réinitialisant la vitesse du personnage sur son axe de déplacement, et en ajoutant sa vitesse-gravite-personnage et sa vitesse-déplacement, ce qui donne donc :

```
//réinitialisation de la vitesse sur les axes de déplacement
collision_box.velocity = new Vector3(collision_box.velocity.x * Mathf.Abs(direction.x),
                                     collision_box.velocity.y * Mathf.Abs(direction.y),
                                     collision_box.velocity.z * Mathf.Abs(direction.z),)
collision_box.velocity += vitesse_gravite_personnage + vitesse_deplacement;
```

Ainsi, nous pouvons avoir un personnage qui se déplace tout en étant soumis à de la physique, et donc à de la gravité.

### 6.1.3 Double saut

Une des mécanique additionnelle du projet Issidi est le double saut, qui amène donc à la question du saut pour les personnage. En effet, vous avez sans doute remarqué la possibilité de se déplacer, ainsi que la gravité, mais rien ne nous permettait pour l'instant d'aller vers le *haut*, qui est une chose relative, mais passons. Plusieurs problème ont apparus lors de la création du double saut : La volonté de ne pas pouvoir changer de direction lors d'un saut (toute la vitesse au moment du saut est conservé), ainsi que la possibilité du double saut qui se produit à chaque appui sur une touche (lors de la première frame, on saute, on se retrouve alors dans l'air, et lors de la deuxième frame, la touche de saut étant toujours enfoncée, on double saute alors). Afin de résoudre ses problèmes, nous avons dû ajouter un booléen, afin de savoir si le personnage est en l'air ou non, s'il ne l'est pas on peut alors le déplacer d'après la formule précédente ce trouvant dans 6.1.2, sinon nous conservons la vitesse actuelle de la CollisionBox, avec la gravité adéquate. De plus afin d'avoir un double saut qui marche correctement, nous avons dû ne lancer l'action de saut que si la touche de saut a été réappuyée (relâchée puis renfoncée).

### 6.1.4 Dash

Le Dash, une des mécaniques de ce jeu les plus simples à implémenter : Il ne suffit que d'augmenter la vitesse du personnage temporairement, tout en empêchant le relancement de cette action si elle est en cours, ainsi que de bloquer la caméra durant le dash, afin d'empêcher le changement de direction lors de celui-ci, qui est la contrepartie à la vitesse du personnage soudainement augmentés. Julien ayant déjà de vagues connaissances en HTML5 et CSS3 il s'est donc chargé de cette tâche. Ainsi il a pu approfondir sa connaissance des balises et usages de ces 2 langages se complètent. Le HTML5 qui d'une part permet de gérer l'organisation et le contenu de la page (texte, liens, images...) tandis que le CSS3 permet de définir l'apparence (positionnement, décorations,...). En ce qui concerne les interactions avec le serveur (principalement le formulaire de contact) Nous avons utilisé du PHP.

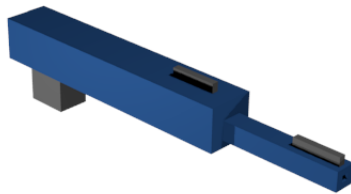
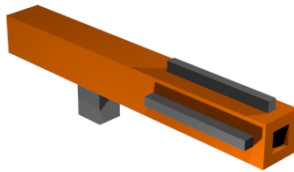
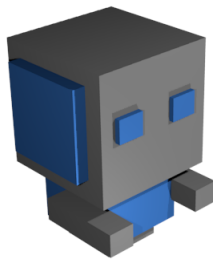
## 6.2 Modèles

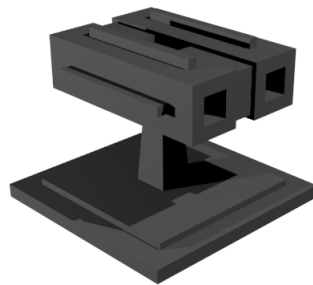
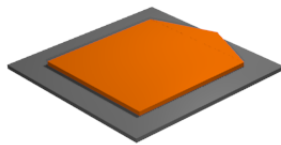
### 6.2.1 Description

Les modèles sont une partie importante d'un projet, c'est pour cela qu'ils ne doivent pas être négligés. Ainsi, nous avons choisi tout d'abord de les faire au tout début de projet, afin de ne pas devoir attendre un modèle afin d'implémenter une fonctionnalité du projet, et de remplacer ce modèle par une simple cube en attendant. De plus, cela nous permettait de nous concentrer dans les prochains mois uniquement sur la partie code du projet, et plus sur les autres parties. Enfin, nous avons décidé d'avoir des modèles cubiques, pour plusieurs raisons : Tout d'abord, il est très simple de le faire, et ne demande pas de grandes compétences en modélisation. Ensuite, cela nous évite de devoir prendre

dans les assets standard de Unity, et donc d'avoir un projet personnalisé. Dernièrement, cela nous permet d'avoir des modèles qui seront plus simples à animer.

### 6.2.2 Exemples





## **6.3 Site Web**

### **6.3.1 Intérêt du site**

Le site internet doit à terme compléter le projet. Il permettra entre autre de la progression d'Issidi via la rubrique des news mais aussi de regrouper plein d'information relative au background et au différent membres du groupe. Enfin pour notre groupe il servira de sauvegarde pour les membres du groupe qui pourront y trouver les rapports de soutenance, des captures d'écrans et même à terme un lien pour télécharger le jeu.

### **6.3.2 Structure du site**

Le site se compose en 5 grandes sections : 1) L'accueil : Cette rubrique représente simplement la page d'accueil du site qui contient une description d'Issidi et



de son univers. 2) Les news : Cette rubrique recense les avancées réalisées dans le cadre du projet (les soutenances, l'apparition du réseau...). 3) A propos : cette rubrique est divisée en quatre parties. Chaque partie correspond à la présentation d'un des membres du groupe. 4) Téléchargements : cette rubrique contiendra les sources du jeu, le cahier des charges et les rapports de soutenance et autres liens utiles. 5) Lien contacts : dans cette rubrique, il est possible de nous envoyer un mail ce qui sera sans doute très utile pour les supports de l'année prochaine qui se poseront des questions sur le projet.

### 6.3.3 Réalisation du site

Avancement : A ce jour le site web est en ligne à l'adresse : [Issidi.shost.ca](http://Issidi.shost.ca), la structure générale est finie et seul le contenu et le design pourront être soumis à des changements.

Choix de l'hébergeur : Nous avons choisi shost pour plusieurs raisons. La première étant qu'il est gratuit bien entendu, néanmoins la gratuité n'en a pas fait notre premier choix pour autant. En effet il existe une myriade d'autres hébergeurs gratuits, seulement souvent l'hébergeur impose des pubs sur votre site pour se rémunérer, shost Non ! De plus avec shost nous avons une adresse simple facile à retenir. Il faut bien reconnaître que [Issidi.shost.ca](http://Issidi.shost.ca) est bien mieux que [Issidi.embergeurtropbiensuperpraticable.com](http://Issidi.embergeurtropbiensuperpraticable.com). Et enfin le petit plus repose dans le service technique qui répond très rapidement (moins de 12h) et en français.

## 7 Planning de la 2<sup>e</sup> 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

## 8 Récit de la réalisation

Ses joies, ses peines...

## 8.1 Issidi.shost.ca

L'une des premières difficultés, mais aussi joie de ce projet a été rencontrer lors de la réalisation du site à la rentrée de janvier.

En effet si la confection de la structure du site (HTML, PHP) a été déroulée sans soucis particulier. La vraie mise en forme en CSS a elle en revanche présenté plusieurs challenges.

Le premier étant plutôt courant lorsque qu'on fait du design : que faire ? quelle couleur ? Quelle forme ? etc... Après plusieurs concertations avec les membres du groupe nous sommes arrivés au design actuel qui nous convient à tous.

Le second fut lié à l'impression de savoir faire du CSS. En effet le fait de penser savoir faire alors que la plupart des méthodes connues datait un peu a conduit à passer beaucoup de temps à essayer de faire le bandeau par exemple avec les propriétés des tableaux. Bien heureusement la découverte de l'inline-block qui a permis d'avoir des boîtes contenant du texte dont la taille pouvait être modifiée. La deuxième partie de la solution est venue du fait d'utiliser des classes sur les classes qui a permis une utilisation intelligente des div classes et des Id qui a conduit à une factorisation de la page de style, la rendant ainsi plus claire et plus simple à modifier.

Conclusion :

Ainsi nous avons réappris l'importance de définir les spécifications de ce que l'on veut faire avant de se lancer tête baissée dans la réalisation. Et aussi qu'il ne faut pas se surestimer et constamment chercher si des nouvelles méthodes n'existeraient pas, et le cas échéant ne pas se contenter de copier-coller la partie de code qui fait ce que l'on veut mais prendre le temps de bien comprendre comment ça marche pour gagner du temps de le futur.

## 9 Conclusion

Ce projet nous permettra donc d'apprendre à construire un TPS à l'aide d'un moteur de jeu, Unity3d, et de pratiquer le travail en groupe pour réaliser un projet d'envergure.