

The Call Of The Goose

Projet de S2

Rapport de soutenance

Yvon Morice

Souleymane Sentici

Amin Salmi

Marie Legay

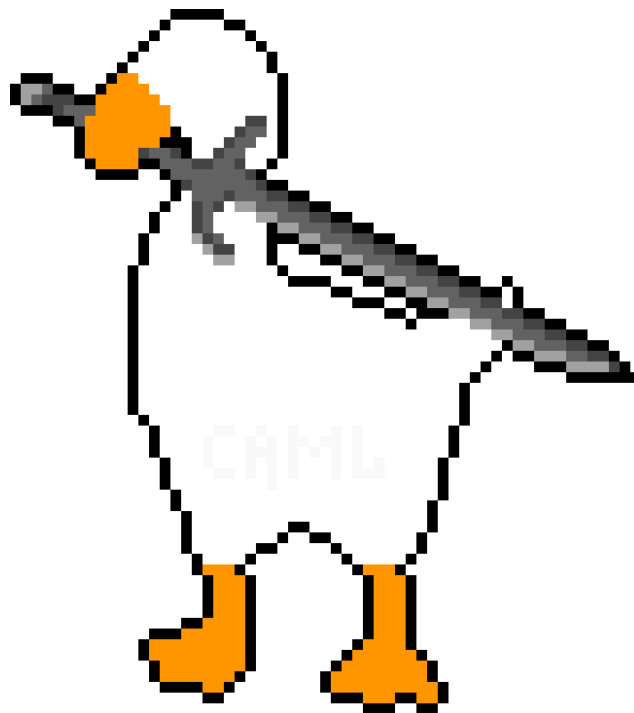


Table des matières

1	Introduction	2
2	Bibliographie	3
3	Le cahier des charges	4
3.1	Le projet	4
3.1.1	Présentation du jeu	4
3.1.2	Principe du jeu	4
3.2	Le groupe	4
3.3	Travail	5
3.3.1	Répartition des tâches	5
3.3.2	Prévisions	6
4	Avancement	7
4.1	Classes (Programmation orientée objet)	7
4.2	Niveaux	8
4.2.1	Génération de niveaux	8
4.2.2	Déplacements dans les niveau	10
4.3	Menus	11
4.4	Sprites	12
4.5	Combats	14
4.6	Multijoueur	17
4.7	Installateur	17
4.8	Attentes pour les prochaines soutenances	20
4.8.1	Design de contenu	20
4.8.2	Ennemis	20
4.8.3	Système de Combats	21
5	Conclusion	21

1 Introduction

Après la validation du cahier des charges, le groupe a commencé à prendre les mains les outils nécessaires à la réalisation de ce projet. Ceux-ci étant : Unity, le moteur utilisé pour créer le jeu, GitHub, la plateforme hébergeant le projet et nous permettant au travers de Git, un logiciel de versionnage, de travailler ensemble et d'avoir un historique des différentes versions du projet. Nous avons également commencé à utiliser les logiciels Piskel, permettant la création de certains assets, notamment les différents sprites et tiles affichés à l'écran, ainsi que Bootstrap Studio, un logiciel WYSIWYG (What You See Is What You Get) nous permettant de réaliser un site web et enfin Inno Setup Compiler, un outil nous permettant de générer un script et de le modifier afin de créer un outil d'installation pour notre projet.

Une fois ces bases acquises, nous avons pu commencer la réalisation du projet. Nous avons ainsi pu commencer à développer de nombreux éléments : la génération de niveau, un menu, un joueur contrôlable au clavier, un niveau tutoriel et le début du système de combat.

Enfin, le site web présentant le projet sera sûrement fini dans les semaines à venir.

2 Bibliographie

Contrôles :

- lien

Génération de niveaux :

- <https://www.youtube.com/watch?v=WtDXk6uuZO4>
- <https://www.youtube.com/playlist?list=PLX2vGYjWbI0SKsNH5Rkpxvvr1dPE0Lw8F>
- <https://www.youtube.com/watch?v=hk6cUanSfXQ>
- https://www.youtube.com/playlist?list=PLBIb_auVtBwA-qr2-WnWX0LjZXkqKu5Aj
- <http://weblog.jamisbuck.org/2011/2/7/maze-generation-algorithm-recap>

Menu / Interfaces :

- https://www.youtube.com/watch?v=zc8ac_qUXQY

Graphismes :

- <https://blog.studiominiboss.com/pixelart>
- <https://www.piskelapp.com>

Site web :

- bootstrapstudio.io/

Installateur :

- <https://jrsoftware.org/isdl.php>

3 Le cahier des charges

3.1 Le projet

3.1.1 Présentation du jeu

- **Genre** : Rogue-like
- **Style** : Action, RPG
- **Plateforme** : PC (Windows)
- **Moteur** : Unity 2019.3.0f5

3.1.2 Principe du jeu

Au lancement de la partie :

- Choix du type de partie (Solo ou Multijoueur)
- Sélection de la classe du personnage
- Lancement du jeu

Lors de la partie :

- Génération du niveau en cours de manière aléatoire
- Le jeu commence à l'étage 0 (sur 5 étages)
- Il y a un boss par étage ainsi qu'un changement d'ambiance aussi bien visuelle que sonore
- Objectif du joueur : atteindre le 5ème étage et vaincre le boss

3.2 Le groupe

Legay Marie, passionnée de jeux vidéos et de musique.

Morice Yvon, passionné de jeux vidéos qui a pour vocation de devenir développeur, si possible dans le domaine du jeu vidéo.

Salmi Amin, passionné de littérature Lovecraftienne, de cuisine et de jeux vidéos, qui a pour vocation de devenir développeur.

Sentici Souleymane est un passionné de jeux vidéos qui a pour vocation de devenir développeur.

3.3 Travail

Après cette première phase de travail, nous avons apporté quelques modifications au planning ainsi qu'au tableau de répartition des tâches proposés dans le cahier des charges. Cette révision tient compte du retard et de l'avance pris dans chaque partie.

3.3.1 Répartition des tâches

Partage des tâches		
Tâche	Responsable	Suppléant
Histoire	Amin	Marie
Ambiance	Amin	Marie
Bande son	Marie	Amin
Character design	Amin	Marie
Animations	Marie	Amin
Dessin 2D	Marie	Amin
Level design	Souleymane	Yvon
Gameplay	Yvon	Souleymane
Tutoriel	Souleymane	Yvon
Système de combat	Marie	Amin
Interactions avec le marchand	Marie	Souleymane
Génération des niveaux	Yvon	Souleymane
Items	Souleymane	Amin
Ennemis	Marie	Yvon
IA des ennemis	Marie	Yvon
Personnages	Souleymane	Amin
Pathfinding	Amin	Souleymane

Menu/Menu de pause	Amin	Yvon
Multijoueur	Marie	Yvon
Site web	Yvon	Souleymane

3.3.2 Prévisions

Tâche	Soutenance		
	1ère	2nde	3ème
Bande son	0	25	100
Character design	50	70	100
Animations	10	50	100
Dessin 2D	10	50	100
Level design	70	90	100
Gameplay	40	800	100
Tutoriel	10	60	100
Système de combat	30	60	100
Interactions avec le marchand	0	50	100
Génération des niveaux	40	70	100
Items	10	50	100
Ennemis	30	50	100
IA des ennemis	10	60	100
Personnages	20	60	100
Pathfinding	0	50	100
Menu/Menu de pause	50	75	100
Multijoueur	20	60	100
Site web	30	50	100

4 Avancement

4.1 Classes (Programmation orientée objet)

Réalisation (Souleymane) Celles-ci sont primordiales pour le bon fonctionnement du système de combat que nous prévoyons de concevoir.

1. La classe Player sert à attribuer les différentes statistiques que possédera le joueur en combat (telles que les points de vie, la mana ou l'endurance par exemple). Elle sert également à créer les méthodes qui permettront au joueur d'utiliser les différents éléments des autres classes présentées ici.

```
public class Player
{
    protected int maxHp;
    protected int maxMana;
    protected int maxEndurance;
    protected int hp;
    protected int lvl = 1;
    protected int exp = 0;
    protected int mana;
    protected int endurance;
    protected int armor;
    protected int gold = 0;
    public string name;
    protected int[] expThreshold = {100, 164, 268, 441, 723, 1186, 1945, 3190, 5233}; // exp nécessaire pour lvl up
    //public Item[] consumablesInventory;
    //public List<Relic> relicInventory;

    Souleymane Sentici
    public int Hp
    {
        get { return hp; }
        set { hp = value; }
    }

    1 usage Souleymane Sentici
    public Player(string name)
    {
        this.name = name;
        //consumablesInventory = new Item[10];
        //relicInventory = new List<Relic>();
    }
}
```

FIGURE 1 – Les mentions en commentaires correspondent aux éléments des classes pas encore implémentées

2. La classe `TestPlayer` qui hérite de `Player` est prévue pour un personnage de test. Celui-ci sera le personnage que l'on incarne dans le tutoriel du jeu, et nous permettra de tester les différentes fonctionnalités existantes lorsque celui-ci sera terminé.
3. La classe `Consumables` sert à implémenter les différents objets à utilisation unique que le joueur pourra rencontrer et utiliser durant ou entre les combats. Les méthodes les concernant sont déjà implémentées dans la classe `Player`, celle-ci sert donc à créer les objets en eux-même, cependant nous ne les avons pas encore conceptualisés.
4. La classe `Enemy` a pour but de donner leur caractéristiques aux monstres que le joueur rencontrera. Elle hérite également de `Player`, puisque les monstres partagent la majorité des caractéristiques d'un joueur, possédant une barre de vie, de mana etc... Nous n'avons cependant pas encore créé d'ennemis particuliers pour le moment.

4.2 Niveaux

4.2.1 Génération de niveaux

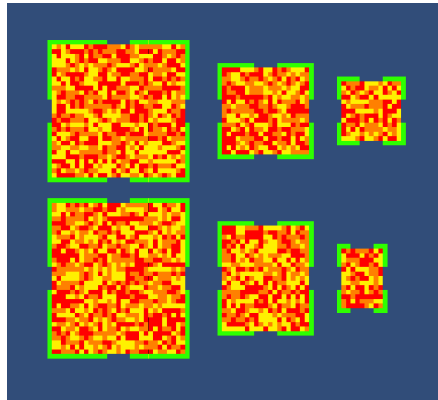
Réalisation (Yvon) Cette section parle de la génération des différents niveaux que le joueur va avoir à traverser, ceux-ci étant au nombre de cinq. La génération de ces niveaux n'étant pas complètement finie, nous aborderons également les algorithmes que nous comptons utiliser par la suite afin de la compléter. Chaque niveau est généré de manière aléatoire de façon à former un labyrinthe ayant toujours la même taille (4 salles de larges pour 5 de hauteur).

 Pictures/Map.png

Un exemple de map pouvant être généré dans l'état actuel du projet

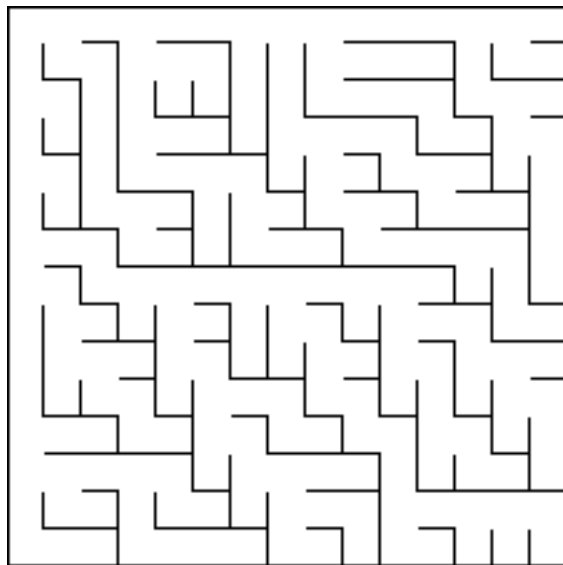
Il y a six modèles de salles avec chacune une taille différente. Chacune

de ces salles va prendre des tiles à afficher et les répartir de manière aléatoires.



Les 6 modèles de salles, de la gauche vers la droite sur la première ligne : boss, spawn, shop. Sur la seconde ligne : grande salle, salle moyenne, petite salle

Pour la prochaine soutenance les salles auront des murs placés de manière aléatoires à l'intérieur ainsi que des ennemis dans certaines salles. Il y aura également des couloirs reliant chaque salle générés via l'algorithme des arbres binaires.



Un exemple de labyrinthe généré avec cet algorithme

Cet algorithme est souvent assimilé aux arbres binaires car pour chaque case, on choisit entre deux cases et non quatre comme pour la plupart des algorithmes de générations de labyrinthe. Cet algorithme n'est pas idéal car il crée un biais mais reste intéressant car il n'a besoin que d'une case à la fois pour faire son choix de direction et n'a pas besoin de garder en mémoire quelle case fait déjà partie du labyrinthe final.

4.2.2 Déplacements dans les niveau

Réalisation (Souleymane) Cette section concerne les déplacements du joueur à travers la carte du jeu. Ceci à été réalisé à l'aide d'Unity, en créant un GameObject représentant le joueur. On lui a ensuite assigné des caractéristiques physiques, comme sa variable de gravité, et un corps rigide pour que celui-ci puisse se déplacer en prenant en compte le décor,

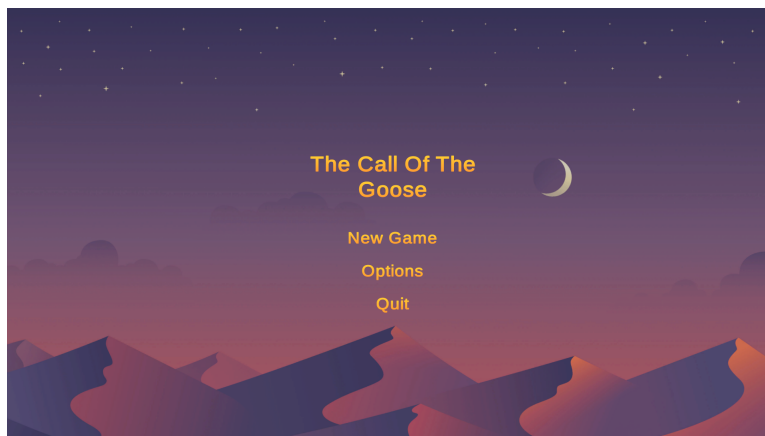
et ainsi ne pas traverser les murs. Enfin, nous avons fait en sorte que le vecteur mouvement du joueur soit actualisé à un rythme constant pour éviter les problèmes de taux de rafraichissement et pour qu'il prenne en compte les commandes données par le joueur.

4.3 Menus

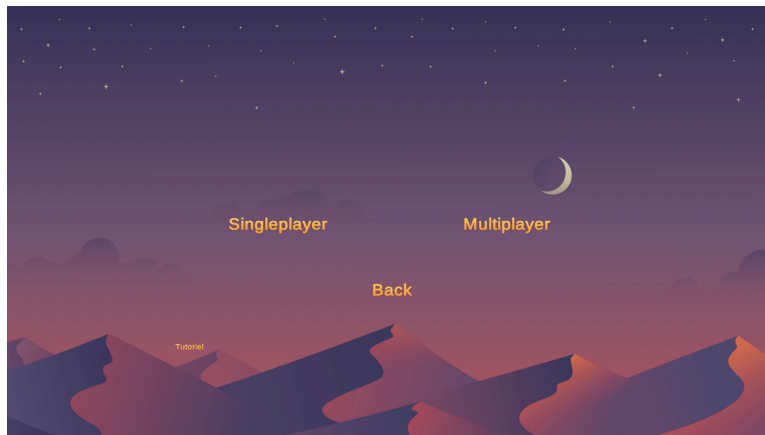
Réalisation (Amin)

Cette section concerne, comme son nom l'indique, le menu de notre jeu. Réalisé sur Unity et avec Rider pour les scripts, le menu principale permet de :

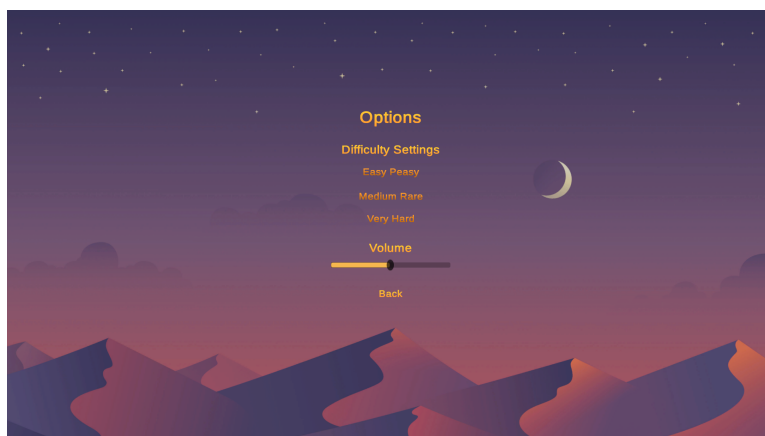
1. Quitter le jeu à l'aide du bouton "Quit".
2. Accéder au sous-menu des options à l'aide du bouton "options", ce sous menu contient un slider qui modifie le volume du son, un bouton "Back" pour revenir au menu précédent, et 3 boutons de difficultés "easy", "medium", et "very hard", chaque bouton assignera à une variable le chiffre 0, 1 ou 2 qui permettront de modifier certaines valeurs du jeu, tel que l'attaque, les point de vie et le nombre de points d'expérience que donneront les ennemies.
3. Accéder au sous-menu de lancement de jeux, avec les trois boutons "Singleplayer", "Multiplayer" et "Tutorial" qui n'ont pas besoin d'être expliqués, et le bouton "Back" pour revenir au dernier menu.



Le menu principal



L'écran de sélection entre les modes de jeu solo, multijoueur et le tutoriel



Le menu d'options, pour changer la difficulté du jeu et modifier le volume du son

4.4 Sprites

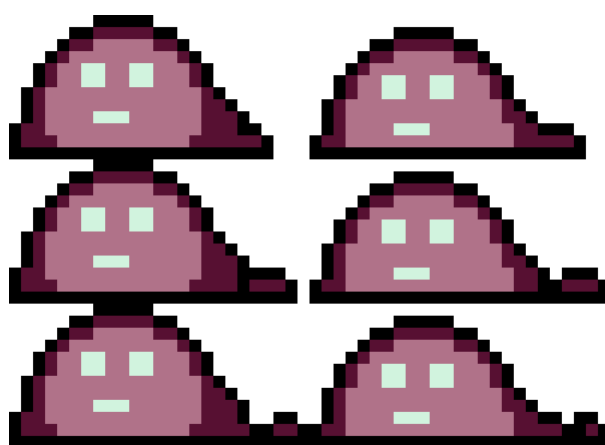
Réalisation (Amin et Marie)

Cette section regroupe les Sprites réalisés jusqu'alors grâce au logiciel Piskel, certains ne sont pas encore inclus dans le jeu. Nous ne montrerons

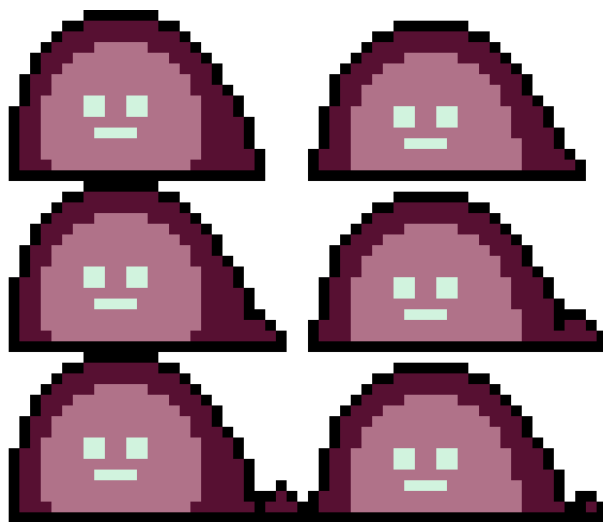
que la Sprite Sheet car il est compliqué d'imprimer un gif sur une feuille en papier.



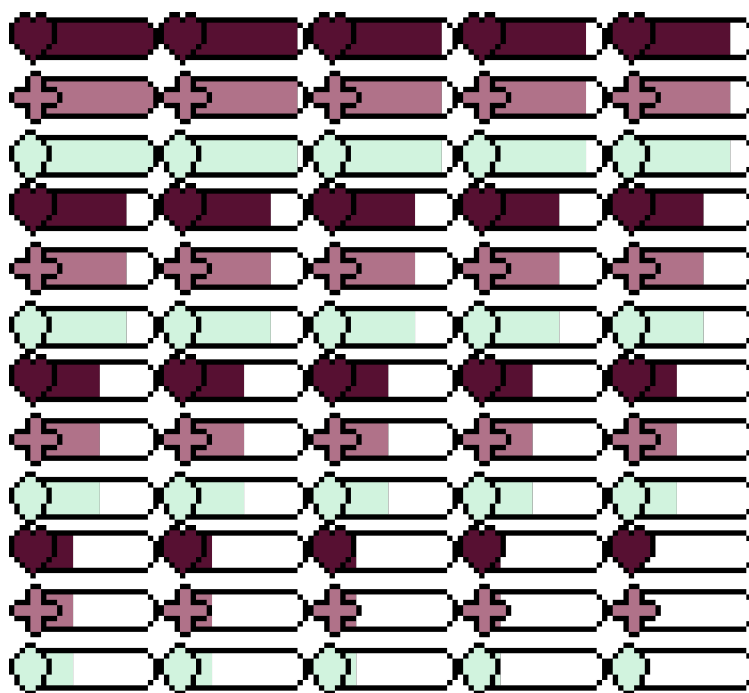
Le personnage joueur.



Un ennemi de base.



Un ennemi d'élite.



Les barres de points de vie, d'endurance et de mana.

4.5 Combats

Réalisation (Marie)

Cette section abordera la mise en place du système de combat. il se base

sur les combat au tour par tour ; plus particulièrement celui de Final Fantasy VII.

```
Frequently called 2 usages
IEnumerator EnemyTurn()
{
    Pl.TakeDamage(Op.Attaque1);

    yield return new WaitForSeconds(1f);

    if (Pl.Hp <= 0)
    {
        State = Battle.Lost;
        End();
    }
    else
    {
        State = Battle.PlayerTurn;
        PlayerTurn();
    }
}
```

Script correspondant au tour de l'ennemi

```
if (Pl.Endurance > 0 && Input.GetKeyDown(KeyCode.Space))
{
    Op.TakeDamage(Pl.Attaque);
    Pl.Endurance -= 5;
    Pl.EndurancePlayer.SetHp(Pl.Endurance);
    yield return new WaitForSeconds(2f);

    if (Op.HP <= 0)
    {
        State = Battle.Won;
        End();
    }
    else
    {
        State = Battle.EnemyTurn;
        StartCoroutine( routine: EnemyTurn());
    }
}
```


Script correspondant au cas où le joueur fait une attaque Physique

A la fin de chaque tour il faut vérifier si le joueur est mort ou si il a tué chaque ennemis sur le terrain. Les dégâts infligés dépendent de la statistique d'attaque du personnage qui inflige les dégâts.

On y trouve une barre de Point de Vie pour chaque entité. Chaque joueur sera en plus munie d'une barre Magie, utilisée lors des attaques à distances, et une barre d'endurance, utile pour les attaques physiques. Ces barres permettront de diversifier la façon de jouer chacun des personnages.

```
4 asset usages 8 usages 4 exposing APIs
public class HealthPoint : MonoBehaviour
{
    public Slider slider;  Changed in 1 asset

    public void MaxHealth(int hp)
    {
        slider.maxValue = hp;
        slider.value = hp;
    }

    Frequently called 8 usages
    public void SetHp(int hp)
    {
        slider.value = hp;
    }
}
```

Code permettant de gérer l'affichage des barres de PV, Mana et Endurance

Le joueur pourra restaurer ces barres grâce à des consommables qui seront ajoutés ultérieurement.

4.6 Multijoueur

Réalisation (Marie)

Cette section parle du début de la création du mode multijoueur. Il a été créé grâce à Unity et Rider pour les scripts. Un des ordinateurs jouera le rôle de serveur hôte, géré par le script "Host". Il permet de créer une partie que d'autres joueurs pourront rejoindre. et de la lancer une fois chaque joueur présent.

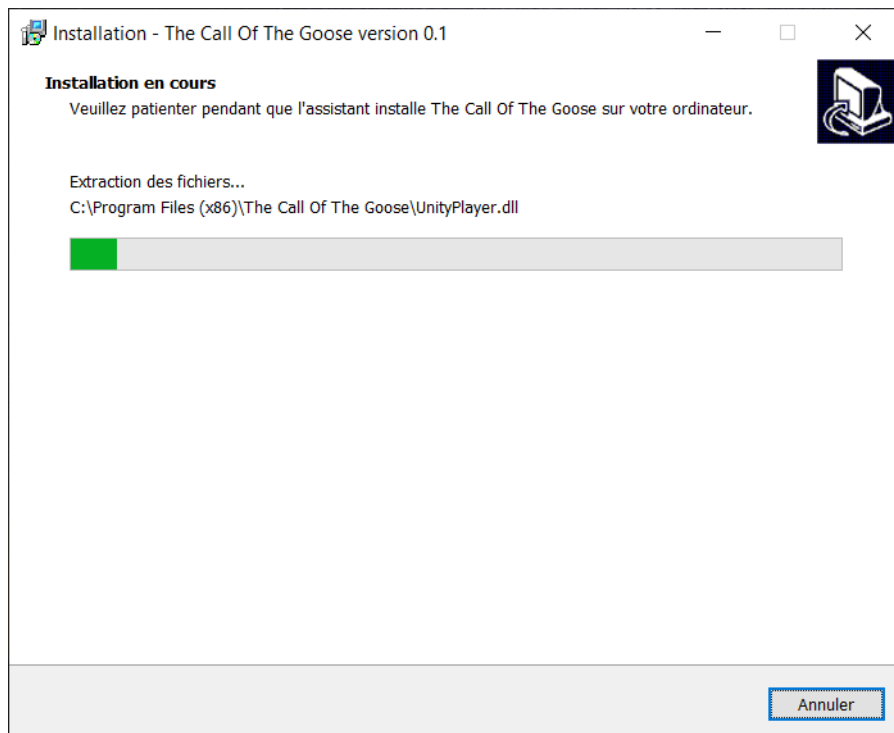
Les autres joueurs s'y connecteront grâce au script "JoinGame". Il permet de détecter les parties en attente de joueurs et de les rejoindre.

Son avancement actuel ne permet cependant pas de jouer au mode multijoueur.

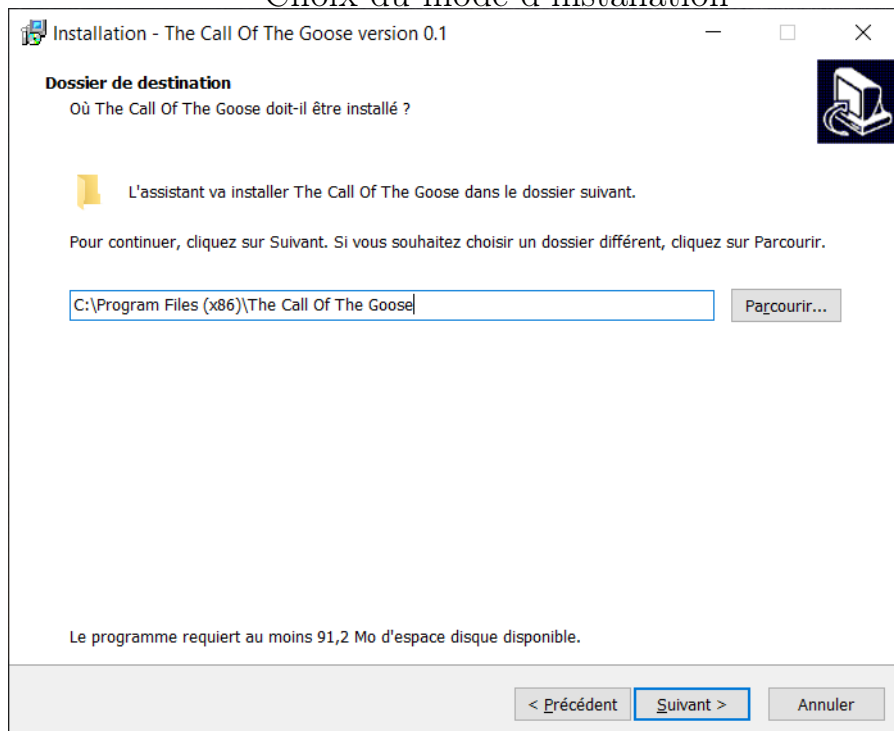
4.7 Installateur

Réalisation (Yvon)

Cette section parle de l'installateur qui a été créé pour ce projet. Le script a été généré via l'outil Inno Setup Compiler puis modifié pour qu'il corresponde à nos besoins. Celui-ci donne la possibilité au joueur de choisir le mode d'installation (en tant qu'administrateur, c'est-à-dire pour tout le monde, ou uniquement pour son compte utilisateur), l'endroit auquel va être installé le jeu, par défaut le jeu sera installé dans le dossier "Program Files" si il a été installé en mode administrateur et dans le dossier "User/AppData/Local/Programs" sinon.

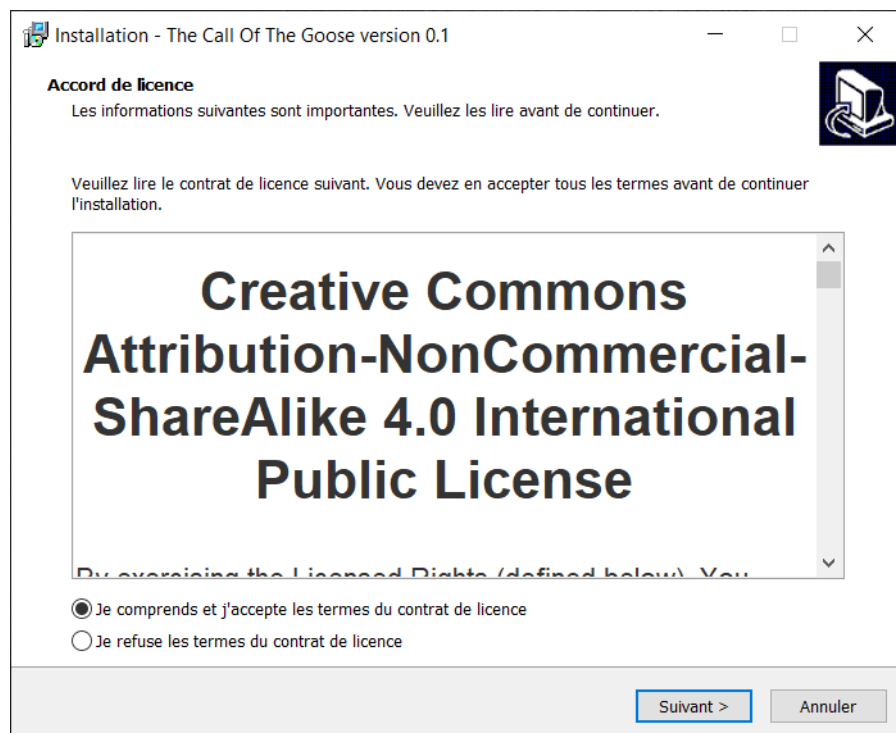


Choix du mode d'installation

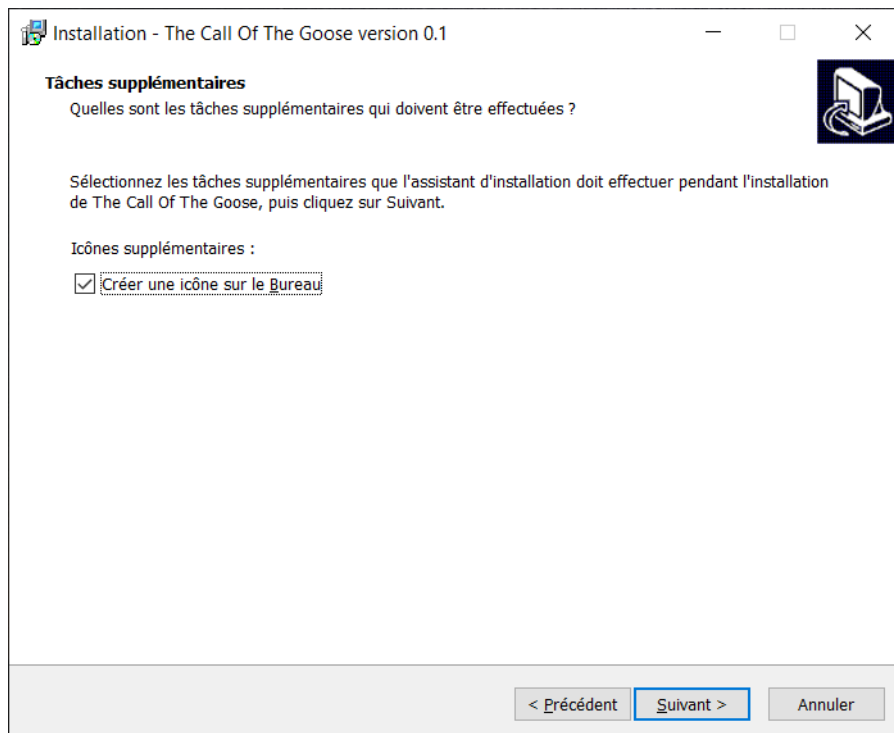


Choix du dossier d'installation

Le joueur devra accepter une licence CC BY-NC-SA-4.0 qui stipule que les utilisateurs ont le droit de réutiliser les ressources du projet de manière non-commerciale si les auteurs sont crédités et que le contenu dérivé utilise une licence du même type. Enfin, l'installateur proposera à l'utilisateur de créer un raccourci sur le bureau après l'installation.



Affichage de la licence



Proposition à l'utilisateur d'ajouter un raccourci sur son bureau

4.8 Attentes pour les prochaines soutenances

4.8.1 Design de contenu

Nous devons créer et implémenter au moins une partie des classes jouables que nous avons prévu. Nous devons également créer les différents objets consommables que trouvera le joueur durant les parties, ainsi que certaines des reliques qui augmenteront ses statistiques pour la partie en cours.

4.8.2 Ennemis

De plus, une partie des ennemis trouvables en jeu devraient être terminés. Ces ennemis seront rencontrables sur la carte générée aléatoire-

ment et tenteront de se rapprocher du joueur. Nous allons donc leur créer une IA, accompagnée d'un algorithme de pathfinding. Ceci leur permettra de poursuivre le joueur qui pourra choisir de les éviter pour aller plus vite, ou de les combattre pour obtenir des récompenses.

4.8.3 Système de Combats

Le système de combat de notre jeu est presque terminé, mais il requiert encore quelques modifications. Une fois celles-ci apportées, les combats devraient être fonctionnels.

5 Conclusion

Pour le moment, le prototype du projet correspond à l'idée que l'on avait au départ du projet. Nous allons continuer à le développer et à améliorer à la fois les graphismes, l'interface et l'ergonomie du jeu. Nous avons été un peu surpris face à la charge de travail et à sa difficulté mais nous avons peu à peu appris à nous organiser et à utiliser les outils à notre disposition. Ainsi même si nous sommes un peu en retard concernant certains points, nous pourrons donc travailler plus efficacement et rapidement grâce à nos acquis.