

Rapport latex

Vargpacket

March 2020

Timothée, Anthony, Matthieu, Nathan

Table des matières

I	Introduction	2
II	Design et Modélisation (responsable : Nathan)	3
1	Ce qui a été fait	3
a	Les bases du corps	3
b	Les mains	3
c	Partie inférieure du corps	4
d	Les cheveux	5
e	Les yeux	7
2	Ce que je compte faire pour la prochaine soutenance	10
III	Unity et Réseau (responsable : Matthieu)	11
1	Ce qui a été fait	11
a	Scène de base	11
b	Réseau	11
2	Ce que je compte faire pour la prochaine soutenance	12
IV	(responsable : Timothée)	13
1	Ce qui a été fait	13
2	Ce que je compte faire pour la prochaine soutenance	14
V	Site web (responsable : Anthony)	15
1	Ce qui a été fait	15
a	Langage de programmation	15
b	Navigation	15
c	Le site	15
d	Hébergement	17
2	Ce que je compte faire pour la prochaine soutenance	17
VI	Conclusion	18

I Introduction

Kingdom Siege est un projet de jeu multijoueur (a deux joueurs) asymétrique, avec un joueur en défense d'un royaume et un joueur qui attaque le royaume en faisant apparaître des sbires. Le jeu s'apparente à un tower defense à la troisième personne dont la caméra affiche un point de vue en plongée. Un mode solo correspondra à un mode "entraînement" où le joueur se défend contre des vagues d'ennemis de différents niveaux pré définis. Le projet sera fait avec Unity et distribué sur Windows. Il utilisera également FMOD pour la partie sonore. Les modèles 3D sont réalisés avec Blender et le multijoueur fonctionnera grâce à Photon. Les différentes parties de ce rapport traiteront de l'avancée du projet sur les différentes parties prévues sur le cahier des charges.

II Design et Modélisation (responsable : Nathan)

1 Ce qui a été fait

a Les bases du corps

Pour la partie graphique, je me suis d'abord attardé sur la sélection du logiciel que j'allais utiliser. C'est sur Blender que j'ai voulu commencer la création d'un personnage jouable.

Avant d'entreprendre quoi que ce soit, je me suis renseigné sur ce logiciel (ce qu'il permet de faire, comment le faire et comment l'utiliser). Je me suis donc tout d'abord familiarisé avec les outils de sculpture que proposait ce logiciel.

Après m'être adapté un minimum avec ces derniers, j'entrepris de sculpter mon premier visage pour commencer. Pris dans mon élan, j'ai aussi sculpté son buste et ses bras dont voici le résultat :



FIGURE 1 – illustration prototype

Comme on peut le remarquer, le bras sur la droite est le résultat intermédiaire.

C'est-à-dire que j'ai réalisé le bras différemment par rapport au reste de la photo. En effet, je me suis basé sur une sphère que j'ai étendue pour lui donner une forme tubulaire et, seulement après, j'ai sculpté pour y ajouter des détails.

A l'inverse, pour la tête et le buste, j'ai directement sculpté dans la sphère.

b Les mains

Continuant dans ma lancée, je me suis mis à la réalisation des mains. J'ai encore une fois découvert une nouvelle façon de fabriquer une base de sculpture avec les "metaballs". Ce sont des objets qui ont les mêmes caractéristiques que les sphères utilisées précédemment sauf quand on en rapproche plusieurs. Elles se mettent alors à se rapprocher comme si on approchait un aimant d'une pâte magnétique. Cette propriété s'avère très utile

pour les doigts avec les “metaballs” qui servent de phalanges. J’ai donc pris 3 “metaballs” pour les 4 doigts et 2 “metaballs” pour le pouce pour commencer (comme montré en haut sur la photo ci-dessous) :

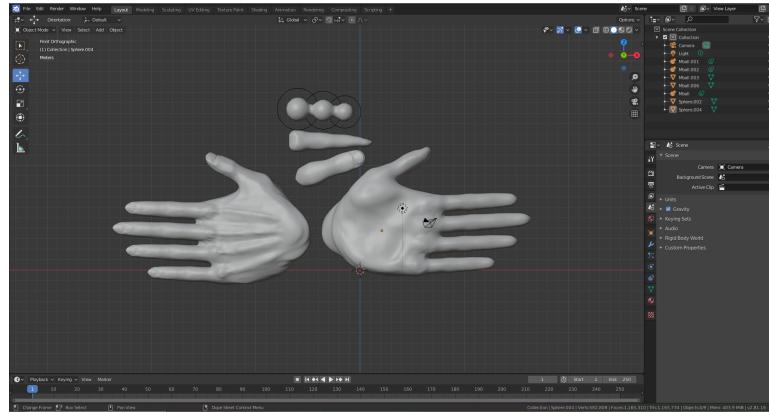


FIGURE 2 – illustration mains

Avec le différents doigts réalisés, je les ai donc assemblé et en utilisant une dernière “metaball”, j’ai assemblé les doigts au creux de la mains et aux éminences thénar et hypothénar de la main.

c Partie inférieure du corps

Après avoir terminé des mains, j’ai rajouté les jambes du personnage, puis je me suis attelé aux pieds. Ou plus précisément aux bottes. Car après réflexion sur sa tenue vestimentaire hypothétique, je me suis rendu compte que faire les bottes directement sur une base de pied triangulaire serait plus judicieux. De ce fait, je me suis mis à faire des bottes montantes que j’ai assemblées avec le reste du corps. Cependant, les différentes parties étaient seulement assemblées et non jointes ce qui est problématique si tout le corps doit bouger. C’est alors, après plusieurs recherches, j’ai découvert le modificateur (traduction littérale de “modifier” en anglais). C’est un outil très polyvalent permettant de multiples actions, mais celle qui me fallait était le modificateur boolean qui permet la sélection de deux objets et de choisir soit leur intersection, différences ou, ce qui est approprié au problème, leur union. Grâce à cet outil, le personnage a enfin une forme humanoïde complète :

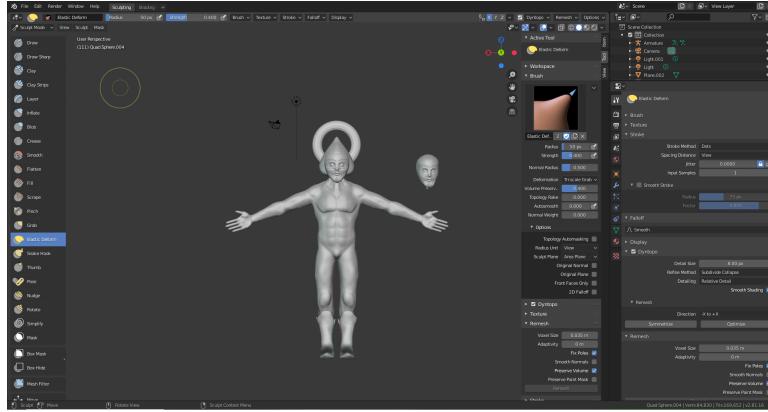


FIGURE 3 – illustration prototype entier

d Les cheveux

Trouvant que le tore situé au-dessus de la tête du personnage était plus encombrant qu'autre chose, j'ai décidé avec les membres de mon groupe de lui mettre des cheveux. Pour ce faire, je suis d'abord partis sur une coiffure inspirée des moines avec un trou dans les cheveux sur le dessus de la tête en y ajoutant des mèches sculptées. Mais le résultat n'était pas satisfaisant. j'ai donc décidé de modéliser les cheveux en manipulant des formes déjà existantes. En aplatissant un cône j'ai pu obtenir ce résultat :



FIGURE 4 – illustration cheveux (1er essai)

Toujours insatisfait, je suis reparti sur l'idée de l'auréole (comme avec le tore) mais cette fois-ci avec ses cheveux. En plus de cette façon de faire, j'ai aussi appris que l'on pouvait associer un cercle et un disque (le premier donnant la

forme du disque et le second créant des tubes finis) ce qui m'a permis de plus modular la forme des cheveux (rayon, longueur), donnant le résultat suivant :



FIGURE 5 – illustration cheveux de derrière (2ème essaie)



FIGURE 6 – illustration de devant (2ème essaie)

Enfin, pour en finir avec les cheveux, j'ai découvert par hasard que le cercle donnant la forme générale aux cheveux pouvait se diviser en plus de 4 segments (comme on peut le voir sur la photo ci-dessus). Cette découverte m'a permis de créer plusieurs mèches avec différentes formes et structures qui allaient encore mieux au personnage :



FIGURE 7 – illustration personnage complet

e Les yeux

A présent, il reste à concevoir la dernière partie du corps visible : l'oeil. Avant d'entreprendre quoi que ce soit avec cet organe, j'ai fait de nombreuses recherches sur les façons de créer un oeil. Même si ça peut paraître un peu trop détaillé, j'ai opté pour la création séparée de l'iris et de la cornée :



FIGURE 8 – illustration la forme de l'oeil

Et la forme des yeux est déjà finie. Tout se joue sur la couleur pour ceux-ci. J'ai alors appris à utiliser l'éditeur de noeuds qui m'a permis de faire les différentes colorations de l'oeil en passant par la cornée, l'iris, la pupille et le globe oculaire montrés sur les photos suivantes :



FIGURE 9 – illustration des yeux avec blueprint 1



FIGURE 10 – illustration avec blueprint 2

Cet éditeur de noeuds, m'a aussi permis de donner une texture non seulement à la peau du personnage mais aussi aux cheveux, sourcils et bottes.

Le résultat final du personnage était terminé car tout le groupe en était satisfait. J'ai ensuite créé une armature servant de squelette au personnage. Cette dernière est indispensable pour pouvoir animer le personnage. Une fois créé, il ne restait plus qu'à la relier au corps et ensuite animer le tout. Pourtant, même en reliant ces deux parties, il m'était impossible de bouger le corps :

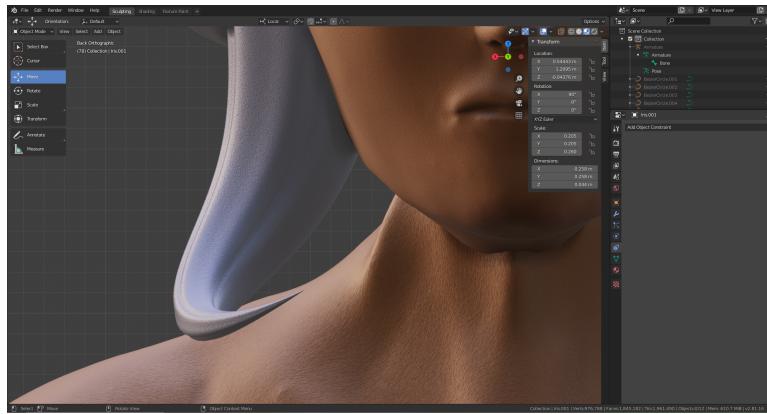


FIGURE 11 – illustration détails des cheveux et de la peau



FIGURE 12 – illustration personnage en couleur

Je ne comprenais pas pourquoi seulement le squelette bougeait et pas le reste. J'étais devant un problème qui allait me prendre plusieurs semaines. En effet au début, j'essaie avec le peu que je connaisse de Blender pour le résoudre mais rien à faire, le problème persistait. C'est donc après des semaines de recherches que j'ai compris qu'il fallait que je modélise le personnage pour l'animer. Et à ce moment-là, je comprenais enfin l'importance de la modélisation et sa différence avec la sculpture. Il faut d'abord sculpter ce que l'on veut, puis sur cette base, il faut modéliser le personnage et seulement ensuite l'animer. J'ai donc dû modéliser notre personnage :



FIGURE 13 – illustration du problème de mouvement



FIGURE 14 – illustration personnage habillé

Pour ce qui est des habits, j'ai d'abord voulu utiliser le logiciel Marvelous Designer qui est spécialisé dans la conception de vêtements mais des conflits et erreurs sont apparus lors de l'importation dans blender. De ce fait j'ai opté pour une combinaison sobre comme on peut le voir ci-dessus.

2 Ce que je compte faire pour la prochaine soutenance

Pour la prochaine soutenance, je compte pour ma part, animer le personnage et commencer la création du personnage attaquant. Ainsi on pourra avoir une idée des personnages présents dans le jeu. Je compte aussi commencer à concevoir la carte avec son décor que le joueur percevra. Sans oublier d'aider à la conception aux sons et musique que l'on compte intégrer au jeu.

III Unity et Réseau (responsable : Matthieu)

1 Ce qui a été fait

a Scène de base

J'ai créé le projet Unity à partir d'un projet 3D vide, auquel j'ai ajouté une scène avec un plan horizontal, et un cube. Les deux ont des colliders et le cube a un rigidbody. J'ai ajouté à cette scène le joueur 3e personne par défaut d'Unity, avec son modèle, son animation, ses scripts, en particulier le script de déplacement. J'ai également ajouté des textures de base à l'environnement.

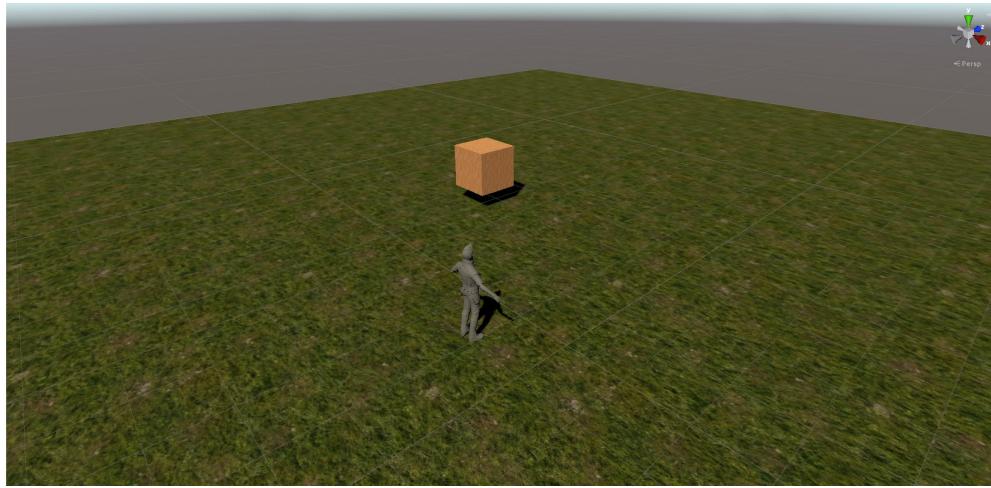


FIGURE 15 – scène de base du jeu

Afin d'avoir une caméra du dessus à la 3e personne, j'ai créé un script pour la caméra, qui suit à une certaine distance (angle de vue et distance) le modèle du joueur (une cible). A ce stade, le joueur en Défense (celui qui contrôle le joueur) pouvait alors se déplacer comme il faut.

b Réseau

Afin d'implémenter le multijoueur, j'ai utilisé Photon Unity Networking (PUN 2). Ce framework permet de facilement implémenter un mode multijoueur en synchronisant des éléments des scènes Unity. J'ai ainsi ajouté un Photon View et Photon Transform View au joueur et à l'objet, afin qu'ils se synchronisent entre les deux joueurs.

J'ai à ce moment ajouté une 2e scène : l'écran titre, avec un bouton qui permet d'ouvrir la connexion au serveur Photon, et qui rejoint un lobby multijoueur automatiquement. En effet, quand un joueur clique sur "Jouer", un script le place automatiquement dans une partie qui attend un joueur si possible, ou sinon il crée une nouvelle partie. Le premier joueur à se connecter

est considéré comme le Défenseur (il prend un tag conservé par Photon), et le deuxième est l'attaquant.

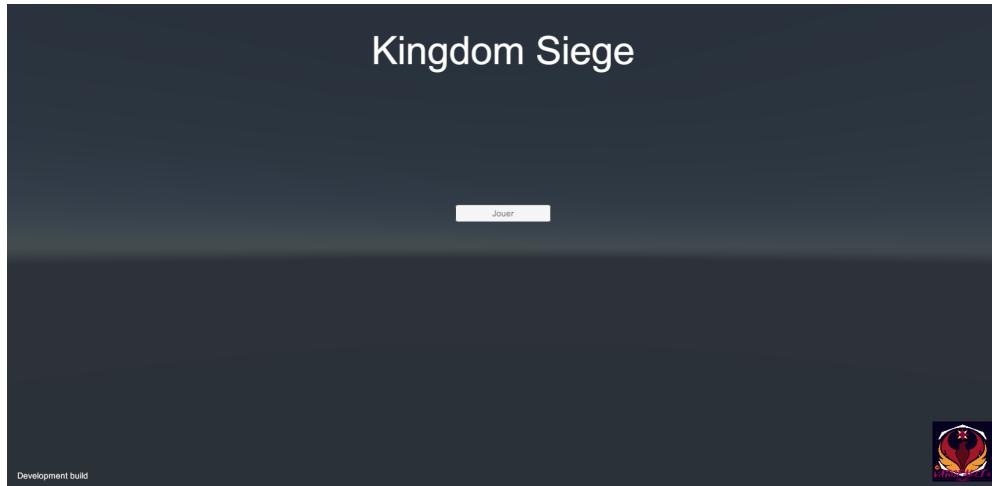


FIGURE 16 – écran titre

Le défenseur peut se déplacer et interagir avec la scène (pousser le cube) et l'attaquant voit la scène sans pouvoir interagir. La caméra de l'attaquant reste pour le moment fixée sur celle du défenseur, mais elle sera libre à l'avenir, pour permettre à l'attaquant de faire apparaître les monstres où il le souhaite.

Le cube a été ensuite transformé par Timothée en ennemi de test (voir la section ci-dessous).

2 Ce que je compte faire pour la prochaine soutenance

Pour la prochaine soutenance, il faut que je règle l'animation du joueur qui ne se synchronise pas (donc qui n'apparaît pas chez l'attaquant). Il faudra ensuite remplacer le modèle par défaut avec celui de Nathan, et commencer le gameplay (attaque du défenseur, contrôle des monstres pour l'attaquant). Je commencerai aussi le son, avec l'API FMOD pour Unity, les premiers effets sonores et la musique.

IV (responsable : Timothée)

1 Ce qui a été fait

Pour que le personnage se déplace j'ai du utiliser les vecteurs Mais selon la méthode utilisé les déplacements sont différents ,par exemple dans les déplacements que l'on a choisi de prendre, le demi-tour prend du temps à s'exécuter en parti dû à l'animation. Ce demi-tour long doit faire prendre conscience au joueur du risque de foncé tête baissée . Pour la caméra on utilise la position du joueur comme référence , référence a laquelle on ajoute une distance en X et une hauteur en y et on définit le joueur comme cible. Par conséquent la caméra suit le joueur à une certaine distance. C'est la méthode décrit ci-dessus utilisant du code que l'on a implémenter. Mais on aurait pu aussi définir le joueur et une caméra comme enfant d'un GameObject vide. Ainsi il aurait fallu que ce soit le GameObject vide qui se déplace, ainsi la caméra et le joueur auraient bougé ensemble.

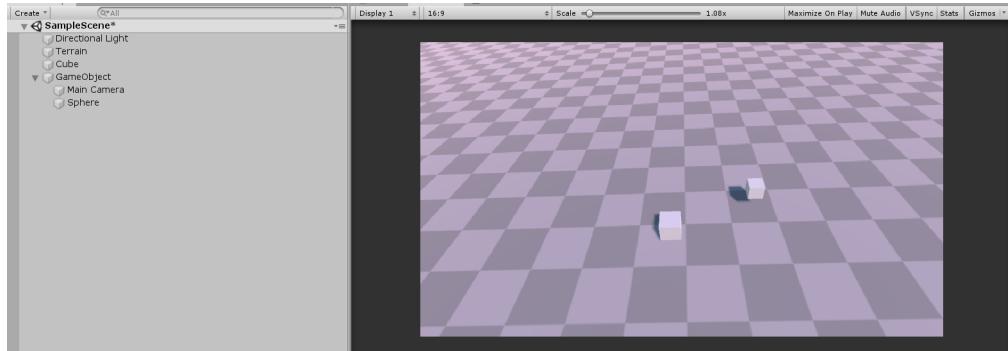


FIGURE 17 – Vue à la troisième personne

On peut faire de diverse manière un objet qui suit le joueur. On peut utiliser le NavMesh pour produire cet effet. Le NavMesh est un component qui permet de définir une zone sur laquelle les NavMesh agent vont pouvoir se déplacer. Le NavMeshAgent possède une destination, si on défini cette dernière comme la position du joueur alors L'Agent va suivre le joueur. Or j'ai eu des problèmes pour le déplacement de l'Agent. J'ai trouvé alors une autre manière sans utiliser le NavMesh. Une autre solution consiste a assigné un script dans l'objet qui suit le joueur, dans ce dernier on va définir 3 méthodes. Déjà si le joueur est à une certaine distance l'objet va le regarder. Pour cela on modifie la rotation de l'objet avec la position du joueur et la position de l'objet. Puis si le joueur se rapproche l'objet va le poursuivre en utilisant la position du joueur comme destination de son déplacement. Pour ça on va utiliser le character controller, cet outil va nous permettre de mieux maîtriser les directions que vont prendre les ennemis, ainsi que leur rotation. De base cet outil permet aux objets d'ignorer les interactions physiques. Effectivement le

character controller va permettre de diriger l'objet sans utiliser de vecteurs.

2 Ce que je compte faire pour la prochaine soutenance

Il faudra implémenter la vie et les attaques pour le joueur et pour les ennemis
Pour cela il faudra créer des variables vie et des méthodes prend des dégâts
qu'il faudra invoqué dans l'objet qui attaque Les attaques retireront une
certaine quantité de Point de Vie avant que l'on joue l'animation de mort et
que l'on détruise le GameObject Ainsi que l'ennemi doit poursuivre sa route
vers un point donné si le joueur est hors de portée.

V Site web (responsable : Anthony)

1 Ce qui a été fait

a Langage de programmation

Le projet ayant pour but l'apprentissage de l'utilisation des divers logiciels nécessaires à la création d'un jeu, j'ai donc commencer la programmation du site en HTML avec Notepad++ qui est un éditeur de texte. Je possédais des bases du langage HTML grâce à des cours de terminale mais je n'avais jamais créé de site auparavant. L'avancement dans la programmation du site fut lente à cause de mes fréquentes recherches de lexique. J'ai également utiliser du CSS(Cascading Style Sheet) pour définir la présentation du site, au lieu de la définir dans la "head" du code HTML afin de pouvoir l'utiliser dans d'autres pages internet si besoin, ce "langage" étant basique, la plus grande difficulté fut l'esthétique du site.

b Navigation

Je comptais implémenter d'autres pages au site, j'ai donc dans un premier temps créer une barre de navigation entre les pages se situant en haut de la page, sur cette barre j'ai mis des liens vers les autres pages, pour l'instant il n'y a que la page d'accueil ainsi que le cahier des charges, lorsque l'on passe d'une page à l'autre, le lien qui mène vers la page sur laquelle nous nous trouvons est écrit en gras afin de mieux se situer.

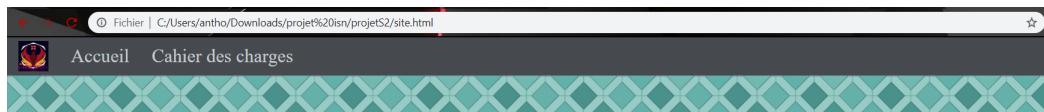


FIGURE 18 – état actuelle de la barre

c Le site

Pour la page d'accueil j'ai mis une brève description du jeu ainsi qu'un logo temporaire pour le jeu.



FIGURE 19 – accueil actuel

J'ai ensuite créé une seconde page simplement constituée du cahier des charges, j'ai donc rajouté un lien sur la barre de navigation qui mène vers la page du cahier des charges.

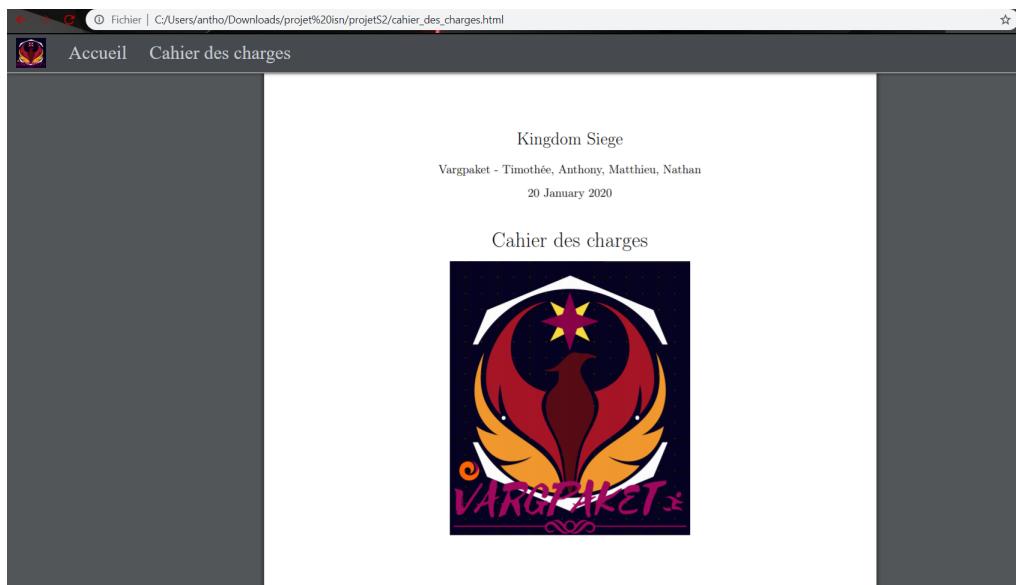


FIGURE 20 – cahier des charges

d Hébergement

Ensuite afin que le site soit accessible depuis Internet il faut que le site soit héberger, pour cela j'ai utiliser GitHub. j'ai d'abord télécharger GitKraken pour pouvoir commit et push le programme du site ainsi que les documents nécessaire a son fonctionnement sur Github,le site est donc maintenant disponible en ligne : <https://anthonycheng06.github.io/VargpaketSite/> .

2 Ce que je compte faire pour la prochaine soutenance

Pour la prochaine soutenance je compte améliorer le site ,avec différentes fonctionnalité et textes,ainsi que l'améliorer au niveau esthétique et ergonomique.

VI Conclusion

Nous avons donc détaillé nos avancements respectifs sur le projet. Si nous regardons le planning présent dans le cahier des charges, nous en sommes globalement au stade souhaité. Et pour certaines parties, nous avons même dépassé ce qui était prévu. On ne compte pas pour autant être moins productif pour le prochain rendu.

Table des figures

1	illustration prototype	3
2	illustration mains	4
3	illustration prototype entier	5
4	illustration cheveux (1er essai)	5
5	illustration cheveux de derrière (2ème essaie)	6
6	illustration de devant (2ème essaie)	6
7	illustration personnage complet	7
8	illustration la forme de l'oeil	7
9	illustration des yeux avec blueprint 1	8
10	illustration avec blueprint 2	8
11	illustration détails des cheveux et de la peau	9
12	illustration personnage en couleur	9
13	illustration du problème de mouvement	10
14	illustration personnage habillé	10
15	scène de base du jeu	11
16	écran titre	12
17	Vue à la troisième personne	13
18	état actuelle de la barre	15
19	accueil actuel	16
20	cahier des charges	16