# Hub project description

*Warden Engine*

## Context and purpose of the project

*La piscine de C++ 2020 a fait émerger parmi notre groupe une idée, celle de créer un moteur de jeu, inspiré de Unity, et nous permettant de créer des jeux vidéo en 2D.  
Notre groupe n’étant composé que de redoublants, nos connaissances du programme de tek2 et la charge de travail moindre (compte tenu des modules ne nécessitant pas une remise en jeu) nous ont encouragé à dédier plus de temps à ce projet. De plus, nous savions qu’il pourrait nous permettre d’avancer plus vite sur certains projets de l’année, tels que l’Arcade ou le Zappy.*

*C’est ainsi que le Warden Engine fut créé, durant le rush 3 de la piscine de C++.*

*Ce rush 3 avait une saveur particulière, étant donné que l’école avait autorisé les tek2 à participer à la Global Game Jam 2020, qui se tenait au même moment, sous certaines conditions. Il était demandé aux teks 2 de ne pas chercher d’aide extérieure sur la partie développement de leur projet, ainsi que de garder en tête que ce rush avait pour but de démontrer leurs capacités sur le paradigme objet.  
Le projet du Warden Engine avait déjà fleuri dans nos esprits, et cette opportunité était l’occasion rêvée de développer un premier prototype. C’est ainsi qu’après 2 jours de rush, une première version du système émergea. Cette première version était un semblant de système ECS (Entity Component System), qui permettait au développeur d’un jeu de créer ses objets et de les afficher dans une scène.*

*Au bout de ces deux jours, cette première version a révélé de nombreuses erreurs de conception, et il a été décidé, si le projet venait à être continué, de reprendre le système de zéro, en tenant compte des erreurs réalisées sur cette première version.*

*C’est ainsi que, un mois après cette première version, l’arcade débuta. Quand est venue la question de s’il fallait reprendre le Warden pour l’améliorer au travers de ce projet, la réponse fut unanime.  
Le Warden Engine devait vivre, et il vivrait.  
C’est ainsi que le Warden Engine 2.0, Arcade Edition vit le jour. Nouveau système ECS, fondé sur des composants génériques et indépendants, gestion de scène et de librairies plus poussées, le projet devint fonctionnel, et créer un jeu 2D fut bien plus simple.*

*Fort de ce succès, une étape manquait encore au Warden Engine : la transition vers un logiciel pratique et fonctionnel. Durant l’arcade, les scènes étaient créées en lignes de code, en ajoutant un à un les éléments qui la composaient. Pour la version 3.0, cette limite doit être dépassée, le Warden Engine doit devenir un réel moteur de jeu, et non plus une librairie graphique.*

*C’est avec cette envie et cette passion que nous venons vous proposer notre projet Hub,  
le Warden Engine 3.0, Software Edition.*

## Project leader

*Alexandre Seignol : Partie Software  
Joseph Vidal : Module Musique / sons   
Bastien Joubert : Module Réseau  
Marc-Antoine Thénard : Partie Software*

## Functional objective

*- En tant que développeur de jeux-vidéos, je veux pouvoir créer un projet qui contiendra mes ressources et mes scènes.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir créer des scènes (des niveaux) dans mon projet.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir créer des objets et les ajouter à mes scènes, ainsi qu’avoir la possibilité de les sauvegarder sous forme de préfabs pour les dupliquer et les réutiliser.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir créer des composants et les ajouter à mes objets, ainsi qu’avoir la possibilité de les sauvegarder.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir compiler mon jeu et en obtenir un exécutable.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir sauvegarder mon travail sur un projet et le reprendre plus tard.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir ajouter des textures, des animations, des sons et des musiques à mon projet.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir tester mon projet sans avoir à l’exporter.*

*- En tant que développeur de jeux-vidéos, je veux pouvoir créer un installeur pour mon jeu.*

## Technical / Technological environment

*Langage utilisé : C++  
Librairie graphique des jeux : SFML  
OS : Linux (cross-platform envisageable)  
Librairie graphique utilisée pour créer le logiciel : Qt*

## Delivery description

*Must :*

* *Interface graphique*
  + *Création de scènes*
  + *Ajouter des objets à une scène à la volée (menus déroulants)*
  + *Ajouter des composants aux objets (menus déroulants)*
  + *Quelques objets et composants créés  
    Objets : Basique (vide), rectangle, bouton, caméra  
    Composants : Transform, Collider, MeshRenderer, TextRenderer, RigidBody*
  + *Sauvegarde de projet*
  + *Compilation de projet*
* *Module Musique*
  + *Fonctionnel via ligne de code (aucun lien avec l’interface graphique)  
    Gestion de musiques et sons à travers des fonctions dédiées dans le moteur*
* *Module Réseau*
  + *Fonctionnel via ligne de code (aucun lien avec l’interface graphique)  
    Gestion de connections à travers des fonctions dédiées dans le moteur*
* *Documentation Doxygen Complète pour les Must*

*Should :*

* *Interface graphique*
  + *Ajouter des objets à une scène à la volée (glisser-déposer)*
  + *Ajouter des composants aux objets (glisser-déposer)*
  + *Explorateur de fichiers*
  + *Création de fichiers (Objets, composants, scripts)*
* *Module Musique*
  + *Intégration dans l’interface graphique (menus déroulants)*
* *Module Réseau*
  + *Intégration dans l’interface graphique (menus déroulants)*
* *Documentation Doxygen Complète pour les Should*

*Could :*

* *Interface graphique*
  + *Éditeur visuel*
* *Module Musique*
  + *Intégration dans l’interface graphique (section dédiée)*
* *Module Réseau*
  + *Intégration dans l’interface graphique (section dédiée)*
* *Documentation Doxygen Complète pour les Could*

## Organization and temporality

Chaque partie de cette roadmap est dépendante des autres parties d’une même colonne.  
Chaque personne doit avoir terminé sa partie pour que le groupe puisse passer à la phase suivante, et les personnes ayant terminé leurs parties seront chargées d’aider les personnes n’ayant pas terminé la leur.

|  |  |  |  |
| --- | --- | --- | --- |
| *///* | *Must - 1/06/20* | *Should - 12/06/20* | *Could - 26/06/20* |
| *Alexandre* | *UI + gestion de projet* | *Explorateur de fichiers + création de fichiers* | *Éditeur visuel* |
| *Joseph* | *Module Musique* | *Intégration graphique musique* | *Intégration dans l’interface graphique (section dédiée)* |
| *Marc-Antoine* | *Logique logicielle* | *Glisser - déposer + améliorations* | *Éditeur visuel* |
| *Bastien* | *Module réseau* | *Intégration graphique réseau* | *Intégration dans l’interface graphique (section dédiée)* |
| *Tout le monde* | *Doxygen* | *Doxygen* | *Doxygen* |