DOCUMENT DE CONCEPTION

Travail présenté à :

Jean-Christophe Demers

Dans le cadre du cours :

Projet synthèse

420-C61-IN, 00001

Réalisé par :

William Caron : 1569047

David Demers: 1942878

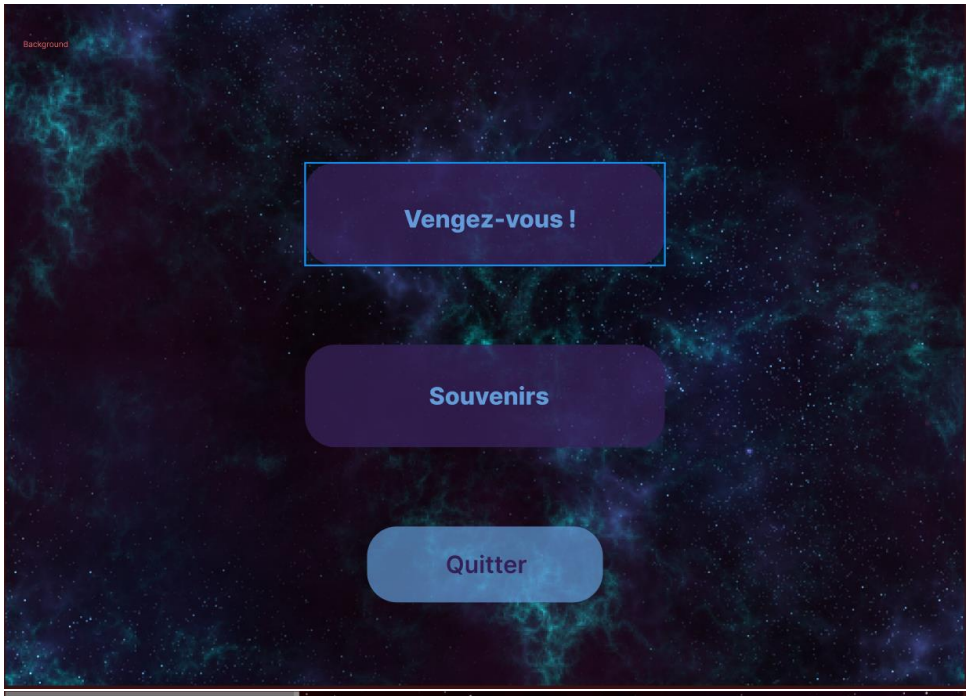
Technique de l'informatique

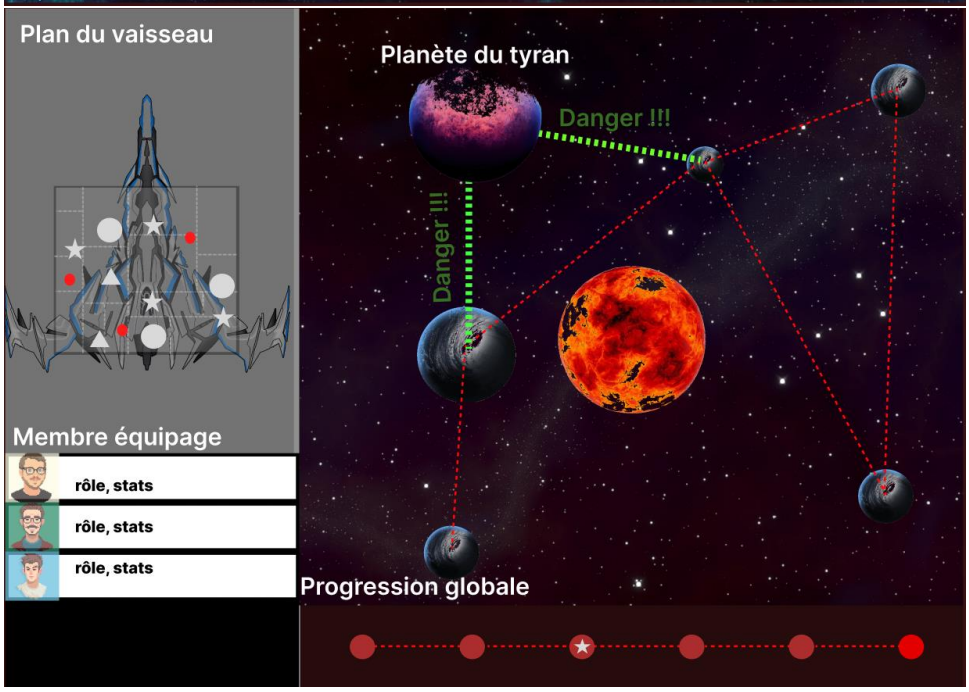
Cégep du Vieux Montréal

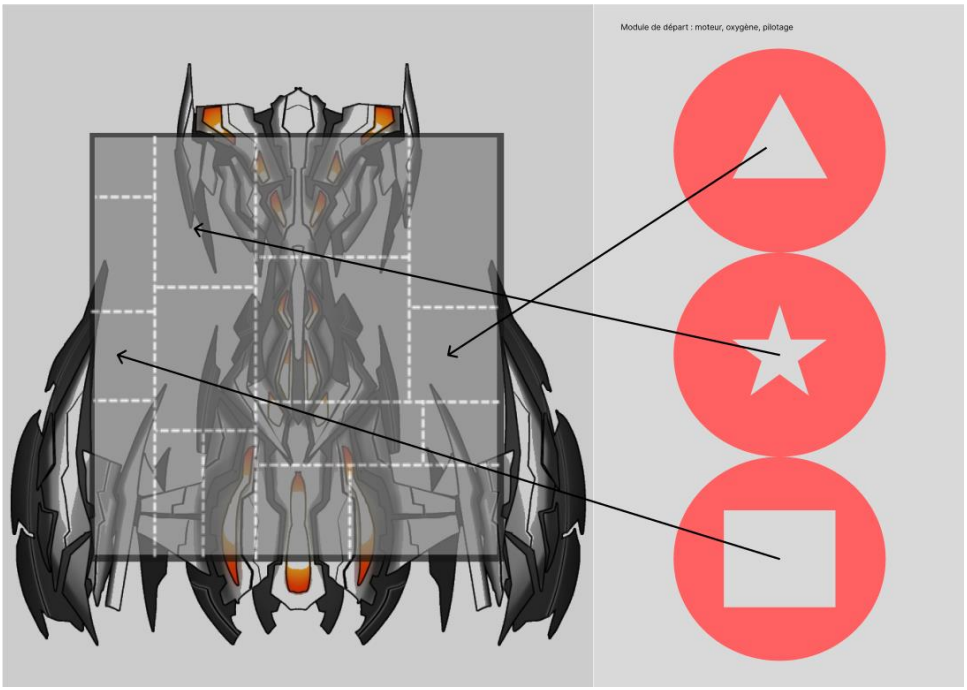
Date de remise :

Mercredi le 28 septembre 2022

Maquettes

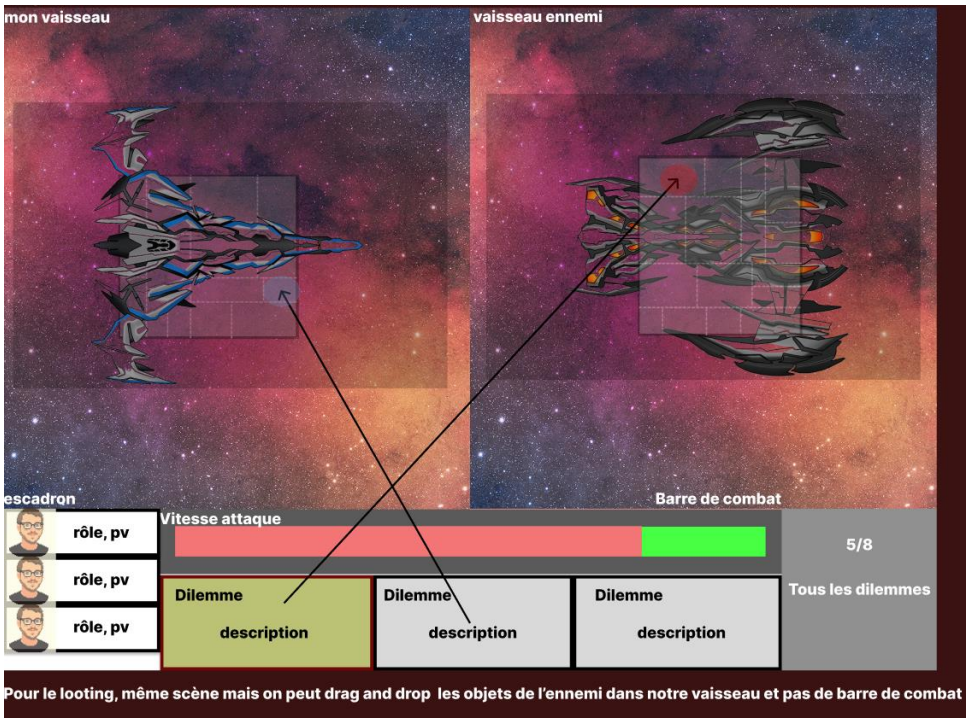

Background
Vengez-vous !
Souvenirs
Quitter


Plan du vaisseau
Membre équipage
rôle, stats
rôle, stats
rôle, stats
Planète du tyran
Danger !!!
Danger !!!
Progression globale


Module de départ : moteur, oxygène, pilotage


Retour
Essayer de vous rappeller...
De vos objets...
image
description
image
description
image
description
image
description
image
description
image
description
Des vaisseaux conquérants..
Nom du vaisseau
Nom du vaisseau


apparait si click sur planète, ceci est un évènement généré aléatoirement
texte
Oui
Non
Quitter


mon vaisseau
vaisseau ennemi
escadron
rôle, pv
rôle, pv
rôle, pv
Vitesse attaque
Barre de combat
5/8
Dilemme
description
Dilemme
description
Dilemme
description
Tous les dilemmes
Pour le looting, même scène mais on peut drag and drop les objets de l'ennemi dans notre vaisseau et pas de barre de combat

Conception UML

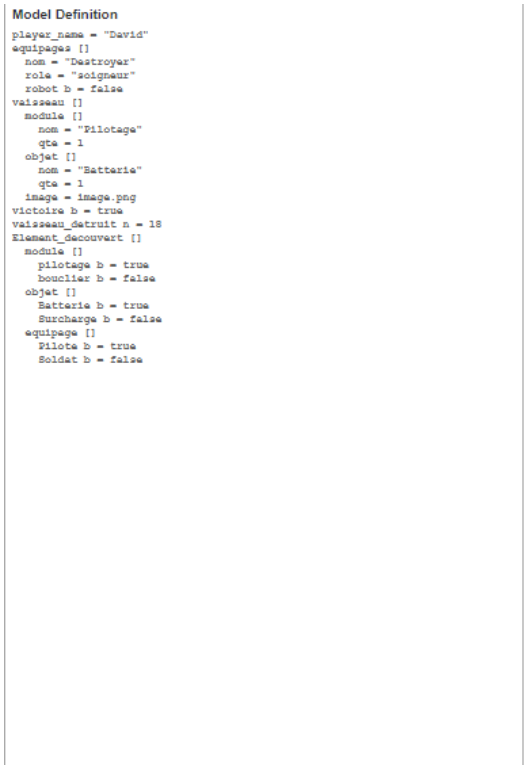
Diagramme des cas d'usage

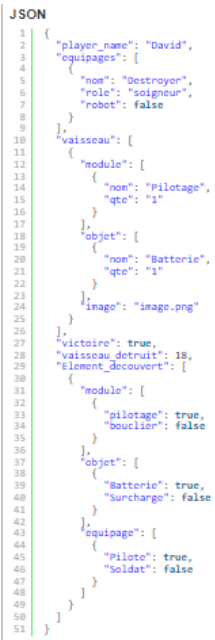
Page 2 : https://app.diagrams.net/#G1mqIU2U10wk2HH_WTVd4IGeJ71Gk4xS81

Diagramme des classes

Page 1 : https://app.diagrams.net/#G1mqIU2U10wk2HH_WTVd4IGeJ71Gk4xS81

Structure de données





http://www.objgen.com/json/models/2s33

Éléments de conception

- Structures de données
    - Choix : liste, Vector2D, arbre binaire
    - Motivations :
        - Liste : nous voulons avoir une structure de donnée mutable pour pouvoir changer les éléments et ordonnés pour pouvoir itérer chaque élément facilement. On veut pouvoir ajouter des éléments au fil de la partie, il nous faut donc une structure non-immuable. De plus, chaque élément n'est pas unique, c'est pourquoi nous avons choisi la liste.
        - Vector2D : nous avions besoin d'une structure pour identifier des coordonnées en X et en Y dans un plan cartésien, nous avons donc choisi un Vector qui est une structure qu'offre Unity pour représenter des coordonnées
        - Arbre Binaire : nous avons choisi un arbre binaire pour faire une partition binaire de l'espace en divisant nos salles de manière récursive. En représentant ces divisions dans un arbre il est facile de représenter en objet les salles, puisque ceux-ci se trouvent à être les feuilles ("leafs") de l'arbre binaire.
    - Éléments Techniques :

        - Arbre Binaire : nous devons faire un mécanisme de balancement sur l'arbre binaire (un pourcentage plus bas de division par niveau dans l'arbre) pour assurer une homogénéité de salle à gauche et à droite de la racine.
        - Vector2D : le vecteur ne représentera pas une direction mais bien une coordonnée dans un plan cartésien
        - Liste : il y aura une liste de planètes, de membre d'équipage, de planètes rencontrées par le joueur, de module, d'objets.
- Patrons de conception
    - Choix : State (État)
    - Motivations : Il y aura une IA qui aura un Finite State Machine, il faut donc implémenter plusieurs états qui changerons au fil du déroulement d'un combat.
    - Éléments Techniques : Il existe des déclencheurs en Unity qui nous permettra de passer d'un état a un autre selon le changement de variable, par exemple, si une pièce de vaisseau est à moins de 20% de sa vie maximale, l'état se changera en réparation. L'état initial sera l'attaque.
- Expression régulière
    - Choix : classe Regex dans Unity
    - Motivations : Lors de la saisie du nom du joueur, il nous faut nous assurer que le joueur n'entre que des lettres, il faut donc refuser les caractères spéciaux, les chiffres, etc.
    - Éléments Techniques : Nous utiliserons le Regex de Unity et sa méthode "IsMatch" qui vérifie si le String saisie ne contient ni caractère, ni chiffre.
- Algorithme
    - Choix : Binary Space Partitioning
    - Motivations : Permet de subdiviser des objets salle en plusieurs sous objets qui sont des enfants de l'objet parent.
    - Éléments Techniques : Avec un nombre de génération et avec l'utilisation de la récursion tout en y associant une ou des variables de contrainte. Il est possible d'obtenir le nombre d'objet final désiré.
- Mathématique
    - Choix : Modèle de récursion mathématique
    - Motivations : Implique l'implémentation de variables contrôlés pour gérer des comportements aléatoires.
    - Éléments Techniques : Ce type de comportement aléatoire va permettre de bien partitionner la création des salles de vaisseau comme les variables seront manipuler de sorte à respecter un type de modèle pour empêcher une série de division dans le même sens.

Lien avec le cours de Veille Technologique

Quel(s) technologie(s) comptez-vous explorer dans le cours de C66 ?

Dans le cours du programme Veille Technologique, nous avons décidé d'utiliser le Binary Space Partitioning. Ce principe qui possède un système mathématique de niveau complexe, comme il utilise la récursivité, peut permettre de réduire le temps d'exécution de plusieurs types de programme selon leur cas d'utilisation. Il peut aussi permettre plusieurs autres comportements comme il implique le partitionnement. C'est cette approche de niveau technologique qui nous fait penché vers l'utilisation de cette technologie.

Pourquoi avoir choisi cette technologie en lien avec votre projet ?

Nous avons choisi le BSP car il nous fallait une méthode de génération procédurale de vaisseau, nous avons donc choisi un BSP puisque celui-ci créer des formes rectangulaires facilement représentable dans un plan cartésien, ainsi il nous est facile de connaitre les coordonnées adjacentes, diagonales, de chaque point et ainsi faire des synergies entre pièces, objets, etc. De plus, le résultat du BSP ressemble vraiment à ce que pourrait ressembler un vaisseau, c'est à dire qu'il y a des pièces de différentes grandeur, mais tous de même de taille semblable (il y a une certaine uniformité dans la partition aléatoire).

Qu'apportera cette technologie ?

Le BSP apportera une dimension de rejouabilité presque infinie puisque chaque vaisseau est unique et il faudra sans cesse s'adapter aux différentes configurations du vaisseau. De plus, certaines partitions sont avantageuses pour le joueur, d'autre le désavantage, ce qui peut rendre une partie plus ou moins difficile. Ce sentiment de nouveauté donne envie au joueur de démarrer une nouvelle partie et découvrir un vaisseau complètement différent du précédent , et donc de changer sa manière de combattre en conséquence.