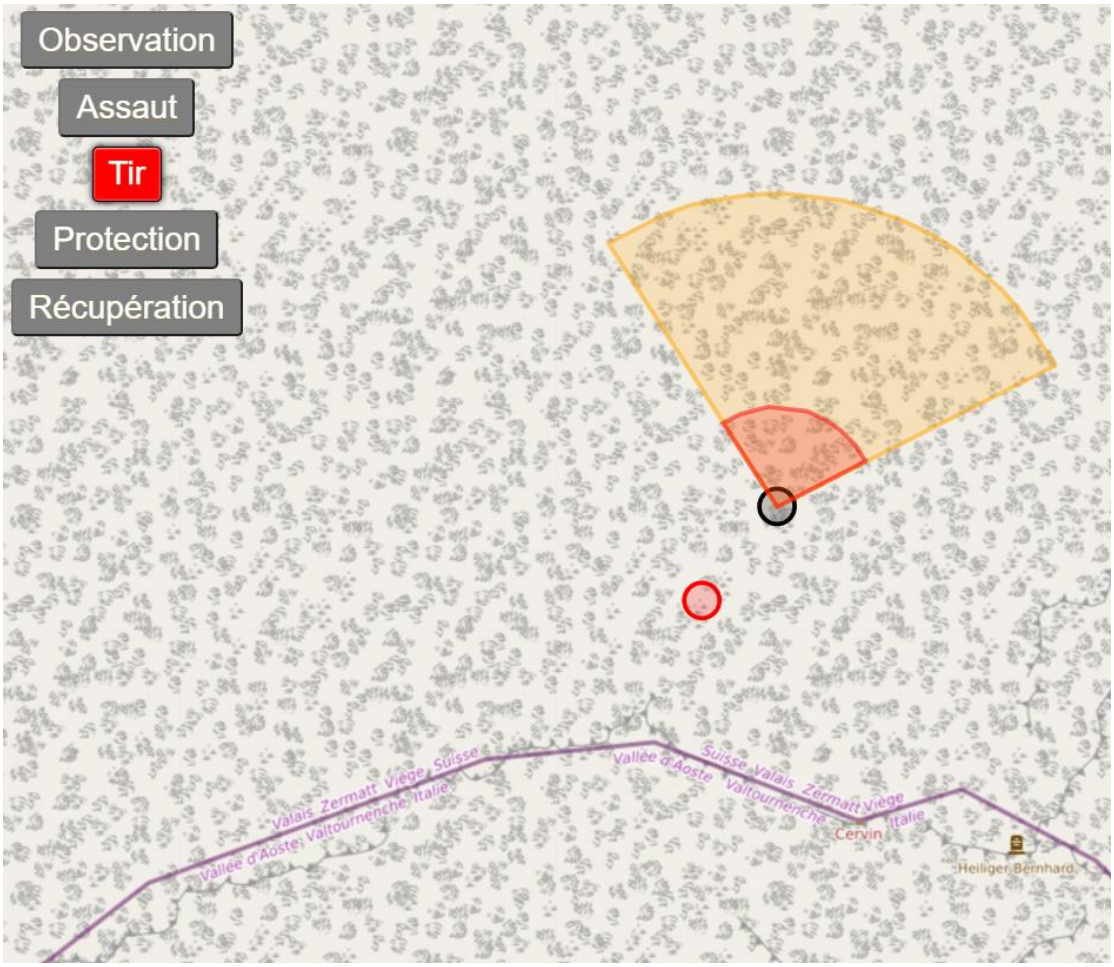


Documentation programmeur

PROJET CAPA SUJET 2

HEAU Vincent | Cours de WebMapping | 28 février 2022

Pour tout questionnement sur le fonctionnement du jeu, lire au préalable la documentation utilisateur

1- Compte Always Data et mise en ligne du site sur la plateforme Always Data

La mise en ligne du site sur Always Data est relativement complexe. En effet, il s'agit d'un site fait avec Node JS. Nous avons suivi cette vidéo assez complète:

How to host NodeJS Express server on Alwaysdata.net

Adresse du site en ligne :

https://vincentheau-capa.alwaysdata.net/capa.html

Code d'accès à la base de données

Accès à Always Data : heauvincent@gmail.com / Capa2021

Accès à la base de donnée : Accéder à l'onglet PostgreSQL/PhpMyAdmin et entrer :

vincentheau-capa / Capa2021

2- Fonctionnement, fichier Capa.js

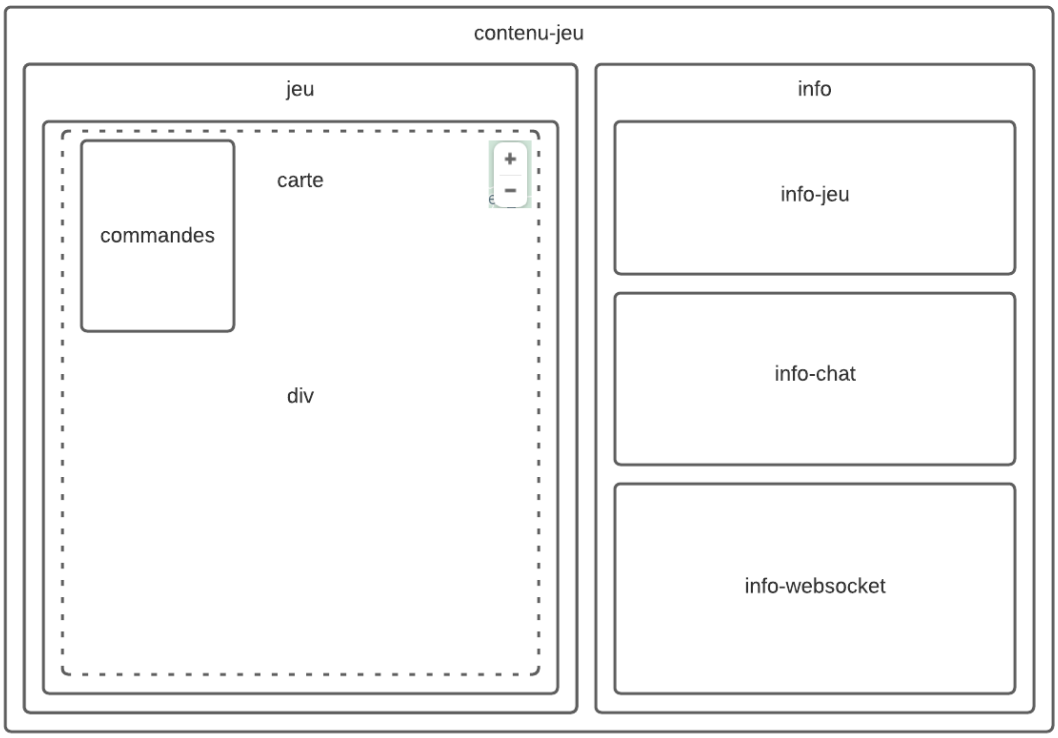
Le code fonctionne avec Node JS. Il est donc nécessaire d'avoir un fichier javascript serveur qui permet le lancement du jeu. C'est le rôle de Capa.js. Il permet le fonctionnement en Websockets. En effet, la première étape est la création d'un serveur via la variable serveur qui écoute ensuite le port 80 pour un fonctionnement en local.

Ce code gère aussi la connexion à la base de données en ligne, ainsi que la mise à jour via les Websockets de cette base de données.

L'exécution de ce code se fait dans un terminal, via la commande **node capa.js** (qui requiert d'abord au préalable installé Node JS sur la machine) et non dans le navigateur

3- Style Css de la page et code html

Pour comprendre le style CSS, voici un schéma des div du fichier capa.html qui permet de coder sur cette architecture. La partie développement (tableau de la base de donnée et boutons) existe au départ mais est masquée. Elle n'apparaît que lorsque le mode développement est activé.



Comme il s'agit d'un site qui est normalement dédié aux téléphones, ajout du responsive web design , en dessous de 500px de large (valeur pour Iphone, Android, etc), la div info passe en dessous de la div jeu

4- Fichier jeu_architecture.js

Le fichier JS d'architecture permet de faire fonctionner l'ensemble des fonctionnalités du jeu. Il permet dans un premier temps d'afficher la carte avec les joueurs. Les commentaires sur le code sont organisés en partie et permettent de se repérer aisément.

Trois points essentiels :

- L'évènement navigator.geolocation.watchposition

  Cet évènement permet d'avoir accès en temps direct à la position du joueur. L'évènement est activé dès lors que la position du joueur change.

- L'évènement DeviceOrientation

  Cet évènement permet d'avoir accès à l'orientation. Sur un ordinateur portable HP ENVY, l'orientation peut être également récupérée.

- Le fonctionnement du bouton tir

Dans le code, deux évènements s'emboîtent. Lorsqu'on appuie sur le bouton tir , le cône apparaît distinguant les deux zones (tir proche en rouge, tir lointain en orange). C'est l'évènement « mousedown ». L'évènement « mouseup » est déclenché dans l'évènement « mousedown » et permet de déclencher un tir. Il se produit une boucle sur l'ensemble des joueurs pour récupérer leur position et vérifier s'ils sont dans le cône via des « if » sur l'angle et la distance.

REFERENCES ET AUTRES

Pour comprendre le fonctionnement des Websockets, le site de Victor Coindet et Vincent Da Olivera est d'une grande aide.

https://ensg_dei.gitlab.io/web-az

Un exercice de chat est détaillé permettant de mieux comprendre les différents appels entre un fichier JS serveur et un fichier JS purement lié au jeu.

Ce chat est conservé dans l'interface.