420-5W5

Projet

Super Cartes Infinies (aka Super HeartStone Infini)

Date de remise : Semaine 6

# L’application

## **Objectifs** :

Créer un application web à l’aide d’Angular, Web API et MVC pour faire un super jeu de carte.

Le site sera utilisé par des membres inscrits et un administrateur.

## Les règles :

* Deux joueurs s’affrontent tour à tour et joue une carte chacun par tour.
* Chaque carte a une certaine quantité de points d’attaque et une certaine quantité de points de défense.
* Après avoir joué une carte, toutes les cartes présente du jouer attaque une carte de l’adversaire.
* Chaque attaque diminue les points défense de la carte ciblée jusqu’à ce que la carte n’ait plus de points de défense.
* Le premier joueur à ne plus avoir de carte en jeu et à ne plus avoir de carte à jouer a perdu la partie.

# Le détail des sections

## Carte

* Une carte doit avoir une valeur d’attaque et une valeur en défense.
* ~~Une carte peut avoir un élément (feu, terre, eau, etc).~~
* ~~Les cartes peuvent être triés pour être plus faciles à repérer.~~
* Les cartes doivent pouvoir être achetés ~~et vendues~~.
* ~~Une carte peut avoir une catégorie en fonction de sa rareté et de sa force. Une carte plus rare doit être vendue plus cher.~~

## Utilisateur

* Un nouvel utilisateur doit obtenir un paquet de carte de départ.
* ~~Un utilisateur doit avoir des points qu’il gagnera en remportant des parties ou en vendant des cartes.~~
* ~~Un utilisateur peut échanger des points pour acheter des cartes.~~
* Un utilisateur doit avoir un jeu de cartes (c’est-à-dire une quantité restreinte de cartes qu’il pourra amener dans une partie).

## Partie

* Un utilisateur peut jouer contre un autre utilisateur présent.
* ~~Une victoire doit donner des points à l’utilisateur victorieux.~~
* Chaque utilisateur doit jouer une carte par tour.
* Chaque carte en jeu doit attaquer une autre carte à chaque tour et ainsi enlever des points de points de défense à l’autre carte.
* Une carte n’ayant plus de points de défense doit être retirée du jeu.
* Chaque action des utilisateurs doit être envoyée au serveur pour que l’autre joueur puisse récupérer l’action effectuée.

# Contraintes

* Le travail doit être effectué en équipes de 4.
* Vous devrez utiliser Git.
* Vous devrez faire une application cliente en Angular.
* Vous devrez faire une application serveur en ASP.NET Web API et MVC.
* Vous devrez utiliser un login utilisant des Tokens.
* Vous devrez créer la section d’administration en MVC.

Critères de correction

Total / 25

# Fonctionnalités / 20

## Gestion des cartes /10

|  |  |  |  |
| --- | --- | --- | --- |
| **La gestion des cartes est excellente.**  Les points énumérés dans l’énoncé ont pratiquement tous été complétés.  Les fonctionnalités ont été réalisés avec une grande qualité. | **La gestion des cartes est bien.**  Les points énumérés dans l’énoncé ont été complétés en majorité et les fonctionnalités ont été réalisés avec une grande qualité.  Ou les points énumérés dans l’énoncé ont pratiquement tous été complétés, mais le fonctionnement pourrait être mieux. | **La gestion des cartes est passable.**  Les points énumérés dans l’énoncé essentiels ont été complétés et les fonctionnalités ont été réalisés avec une grande qualité.  Ou les points énumérés dans l’énoncé ont été complétés en majorité, mais le fonctionnement pourrait être mieux. | **La gestion des cartes est très faible.**  Les points énumérés dans l’énoncé essentiels n’ont pas été complétés.  Ou la qualité des fonctionnalités était insatisfaisante. |
| 9 à 10 | 8 à 9 | 6 à 7 | 0 à 5 |

## Déroulement d’une partie /10

|  |  |  |  |
| --- | --- | --- | --- |
| **Le déroulement d’une partie est excellent.**  Les points énumérés dans l’énoncé ont pratiquement tous été complétés.  Les fonctionnalités ont été réalisés avec une grande qualité. | **Le déroulement d’une partie est bien.**  Les points énumérés dans l’énoncé ont été complétés en majorité et les fonctionnalités ont été réalisés avec une grande qualité.  Ou les points énumérés dans l’énoncé ont pratiquement tous été complétés, mais le fonctionnement pourrait être mieux. | **Le déroulement d’une partie est passable.**  Les points énumérés dans l’énoncé essentiels ont été complétés et les fonctionnalités ont été réalisés avec une grande qualité.  Ou les points énumérés dans l’énoncé ont été complétés en majorité, mais le fonctionnement pourrait être mieux. | **Le déroulement d’une partie est très faible.**  Les points énumérés dans l’énoncé essentiels n’ont pas été complétés.  Ou la qualité des fonctionnalités était insatisfaisante. |
| 9 à 10 | 8 à 9 | 6 à 7 | 0 à 5 |

# Qualité du code / 10

|  |  |  |  |
| --- | --- | --- | --- |
| **La qualité du code est excellente.**  Les points énumérés si dessous sont très bien respectés dans le code. | **La qualité du code est bien.**  Très peu des points énumérés si dessous n’ont pas été respectés dans le code. | **La qualité du code est problématique.**  Plusieurs des points énumérés si dessous n’ont pas été respectés dans le code. | **La qualité du code est très faible.**  La plupart des points énumérés si dessous n’ont pas été respectés dans le code. |
| 9 à 10 | 8 à 9 | 6 à 7 | 0 à 5 |

# Choses à vérifier pour la qualité du code

## Accès aux données

Façon de faire les requêtes et de populer les objets

## Client

Stratégie pour créer les pages

Utilisation d’Angular

Utilisation des API

## Structure du projet

Les classes sont bien organisées le et/ou les projets

L'utilisation de plusieurs projets est utilisée lorsque nécessaire

Réutilisation des classes et des objets

## Web API

Les principes objet sont respectés

Réutilisation du code

Bon découpage en couches

Sécurité de l’API

Utilisation adéquate des classes de transfert

## Maintenance

Facilité de maintenance du code

Utilisation adéquate du gestionnaire de code source

Respect des standards d'équipe

Le code est lisible facilement, si ce n'est pas possible, un commentaire qui explique

## Extensibilité

Peut supporter un grand nombre d'utilisateurs

Peut facilement ajouter des modules

Peut facilement ajouter des informations aux modules existants

Le site est pensé pour une utilisation soutenue à long terme