

## Règles :

- Tout ce qui n'est pas interdit est autorisé
- Si le sujet est ambigu, posez la question, ça ne coûte rien
- Interdit de parler, mais vous avez le droit de chuchoter, si vous chuchotez trop fort vous perdrez des points mais je ne vous préviendrai pas
- Langage de programmation : C++, **sinon : 0**
- Dans ce sujet cela peut être écrit en français, mais dans votre code tout doit être écrit en anglais, **sinon : 0**
- Le rendu se fait à l'adresse : [lois.duplain@ac-lyon.fr](mailto:lois.duplain@ac-lyon.fr)
- Pour avoir tous les points il faut rendre avant la fin du cours, sinon doit être rendu avant minuit (mais il y aura une pénalité de retard), **sinon : 0**
- Ce qui est rendu doit compiler et pouvoir s'exécuter, **sinon : 0**
- Rien n'empêche de me rendre qqch qui respecte toutes les règles à la fin du cours et me rendre qqch de plus poussé avant minuit
- Si les conditions des propriétés des objets ne sont pas respectées (genre ça  $0 \leq x \leq \text{INT32\_MAX}$ ) : **0**

## Sujet :

- 1- Crée un objet Entity qui contient :

- Propriétés de l'objet:

- Name: chaîne de caractères
    - MaxHealth: nombre entier,  $0 \leq x \leq \text{UINT16\_MAX}^*$
    - Health: nombre entier,  $0 \leq x \leq \text{MaxHealth}^*$
    - IsDead: boolean
    - -> Toutes ces propriétés doivent être initialisées par les paramètres du constructeur sauf IsDead
    - Elles doivent aussi être private mais doivent avoir un getter
    - Name doit en plus avoir un setter

- Les méthodes :

- public takeDamage(damager, damages):

- damager -> Entity
      - damages -> nombre entier,  $0 \leq x \leq \text{UINT16\_MAX}^*$
      - -> Doit retirer la bonne quantité de points de vie à l'entity (attention aux contraintes) + afficher un message du style : <Name>: Aïe j'ai pris X points de dégâts de la part de <Damager Name> !
      - Mettre un message différent quand l'entity meurt

- public attack(target):

- target -> Entity
      - -> Doit attaquer l'entity target en lui faisant 10 points de dégâts, soit logique par rapport à ce qu'on a fait avant, pensez à mettre un message

- public heal(healer, hp):

- -> Soyez logiques et pensez à mettre un message

- 2- Faites combattre entre elles deux Entity initialisées avec des paramètres différents, imprimer le résultat dans la console

- 3- Crée un objet Soldat qui hérite d'Entity :

- Params du constructeur:

- Name: chaîne de caractères

- Automatiquement initialisé:

- MaxHealth = 100
    - Health = 100

- Override la méthode attaque pour que le soldat fasse 15 points de dégâts

- 4- Faites en sorte que 2 soldats attaquent une Entity et imprimer le résultat

- 5- Crée un objet Tank qui hérite d'Entity :

- Le tank a énormément de points de vie mais fait peu de dégâts

- Egalement, quand il se fait attaquer, il y a une chance sur 5 pour qu'il se prenne dégâts double

- 6- Faites un combat entre des entités, des soldats et des tanks

- 7- Si tu es arrivé jusqu'ici bravo, appelle moi, je viens te corriger en direct et éventuellement je te complèterais le sujet pour que tu découvres des concepts plus complexes de programmation

\* (n'utilisez pas un type de nombre trop grand inutilement)

