**Essayez de compléter ces devoirs pour être à jour sur le TP. Ces devoirs ne sont pas additionnels au TP2, ils en font partie!**

**Cette partie devrait être complétée pour le cours de mardi 10** **octobre**.

1. ***Hitbox* system**
   1. Créer un système de hitbox. Un gameobject contenant un *collider* de type *trigger* doit être activé lorsque le joueur effectue un coup, **et désactivé à la fin de l’animation**. \*Plusieurs façons de faire cette partie.
   2. Créez un script pour gérer les *hitboxes* dans votre jeu. Ce script devrait, en gros, permettre de détecter les *triggers* avec des personnages ennemis.  
        
      Notez que le script *hitbox* devrait également fonctionner lorsque les ennemis infligent des attaques au joueur. On ne fait pas un script pour les hitboxes ennemies et un autre pour les alliés.
   3. Ajoutez un ennemi dans votre salle de test. Je vous conseille [cet ennemi](https://assetstore.unity.com/packages/3d/characters/robots/robot-kyle-urp-4696#releases). Faites jouer une animation de style “*On* *Hit*” lorsque l’ennemi est frappé par le joueur.
2. **GameManagerStateMachine** -> GameManagerSM
   1. Dans votre jeu, ajoutez une classe qui représentera l’état actuel de notre jeu. Il s'agit de notre GameManager. Assurez-vous également que celui-ci est accessible de partout (Singleton).
   2. Ajoutez ensuite 3 états, que vous devez ajouter à votre SM
      1. GameplayState(get CharacterStateMachine update and fixedUpdate)
      2. CinematicState
      3. SceneTransitionState
   3. \*\* Cette partie sera peut-être faite ensemble à un cours ultérieur, mais si vous avez le temps, faites-la! \*\* Assurez-vous que les parties communes des deux **StateMachines** sont mises en commun, et utilisez l’héritage pour distinguer notre **PlayerCharacterSM** de notre **GameManagerSM**.
   4. Finalement, ajoutez une autre caméra, que vous nommerez “CinematicCamera”, à votre scène. Cette caméra ne suit pas votre joueur, elle peut être fixe. Assurez-vous qu’à l’aide d’une touche, **nous passons de l’état gameplay à cinematic** et que nous passons **d’une caméra à l’autre**. Également, les **inputs de notre joueur devraient être coupés** dans l’état cinématique. (Le personnage ne bouge pas)