William Blanchet Lafrenière  
Groupe 0002

**JEU « QUI SE JOUE TOUT SEUL »**Travail présenté à monsieur François Paradis

PROGRAMMATION DE JEUX VIDÉO II  
420-W13-SF

Département de l’informatique  
Programme programmation web, mobile et jeux vidéo  
Cégep de Sainte-Foy  
15 novembre 2022

Le concept de Wizards War a été repris avec les mêmes états que proposés dans l’énoncé.

Liste des états tel que proposés dans l’énoncé :

* Normal. Le magicien avance vers une tour adverse, et engageant les ennemis s’il en rencontre. Si un ennemi entre dans sa portée, il le choisit comme cible et le garde comme cible jusqu’à ce qu’il soit éliminé ou qu’il fuit hors de portée.
* Intrépide. Si le magicien a éliminé trois cibles, il devient confiant. Il se dirige toujours vers les tours. Il régénère même en combat. Il ne s’occupe plus de choisir une cible humaine, se concentrant sur les tours. Par contre, un ennemi qui vient de l’attaquer devient sa cible. À partir de ce moment-là, il garde toujours cette cible jusqu’à ce qu’elle soit désactivée ou en fuite. Sa vitesse augmente légèrement.
* Fuite. S’il lui reste moins de 25% de ses points de vie, un magicien va fuir vers ses lignes et recherche le point de couvert le plus prêt (tour ou forêt), et s’il en trouve un, devient planqué ou en sureté, selon qu’il entre dans une forêt ou dans sa tour. En fuite, un magicien va toujours reculer vers ses lignes et n’entre jamais en combat. Sa vitesse augmente légèrement (un peu plus rapide qu’intrépide).
* Planqué. Un magicien qui reste planqué en forêt reste immobile peu importe les circonstances jusqu’à ce qu’il ait remonté à 100% de ses points de vie OU s’il remonte à 65% de ses points de vie ET si un ennemi arrive à sa portée. (Si une de ces conditions se produit, il revient à normal). Il engage tout de même les ennemis qui vont passer à sa portée mais ne va jamais bouger. Il choisit toujours la cible potentielle la plus près, que celle-ci l’attaque en retour ou non, et s’il ne se bat pas, il régénère 2 fois plus vite que la normale. Un magicien planqué qui subit une attaque et retombe sous les 25% de point de vie se remet à fuir vers une planque plus proche de sa tour ou vers la tour elle-même.
* En sureté. Un magicien dans la tour ne peut plus attaquer ni se faire attaquer et régénère 3 fois plus vite. Il ressort à 100% de ses points de vie, en état normal. Si pas de last stand et sa tour de planque est détruite, passe en état normal.

Non implémenté

**Voici les métriques à déterminer**

* + Points de vie des tours : 500
  + Points de vie d’un magicien (fixe ou variable) : 200
  + Cadence de tir d’un magicien : 1.5 (normal)
  + Dommages du magicien (pourrait varier selon les états) : 50 (normal)
  + Rythme de spawning des magiciens : 2 secondes + 0.05 secondes par spawn
  + Nombre maximal de magiciens actifs par camp (recyclage) : 40
  + Portée d'engagement des magiciens (pourrait varier selon les états) : 1.6
  + Vitesse de déplacement des magiciens : 2.5 (normal)
  + Forêt : effet sur la vitesse et les dommages subis : même équipe vitesse augmentée de 1, équipe adverse vitesse réduite de 0.5
  + Regénération : 10pv/s pour tous les états

**De plus certains éléments pourraient varier selon l’état (à votre choix)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **État** | **Temps entre chaque régénération\*** | **Cadence de tir** | **Dommage** | **Portée**  **(unités)** | **Vitesse** |
| Normal | 1s / pas en combat | 1.5 | 50 | 1.6 | 2.5 |
| Intrépide | 1s | 1.5 | 55 | 1.6 | 3.25 |
| Fuite | 1s | - | aucun | - | 4 |
| Planqué | 0.6s / pas en combat | 1.25 | 55 | 1.6 | Toujours immobile |
| En sûreté | 0.3s | - | aucun | - | Toujours immobile |
| Last stand | 1s | - | - | - | Toujours immobile |

\* Valeurs présentées dans l’exemple. Vous pouvez modifier ces valeurs selon votre concept.

j’ai tout fait le projet de A à Z par moi-même à part les assets fournis.

**Conseils techniques**

**Listes et collections**

Puisqu’en C# les listes et autres collections ne contiennent de que des références d’objets, vous pouvez vous en servir allègrement pour construire dynamiquement des listes d’objets

* Listes de cibles pour un magicien.
* Listes de tours actives (on retire les tours désactivées).
* Etc.

**Projectiles : communication entre le tireur et la cible.**

Si jamais dans votre projet vous employez des projectiles standards, sachez que vous pouvez de manière simple, si vous en avez besoin communiquer des informations entre le tireur et sa cible peu importe quelle sera sa cible.

* Le projectile garde une référence sur le GameObject de son tireur d’origine.
* À l’impact, il peut connaitre la cible par collision.
* À ce moment là si vous en avez besoin, les deux objets peuvent s’échanger de l’information.
* ATTENTION : il n’est pas garanti que vous ayez besoin de faire ceci, mais si jamais vous en avez besoin…

**Les lignes de couleur dans l’exemple.**

GameObject avec un composant LineRenderer.

* Attention : n’est pas le plus aisé des composants à manipuler pour avoir un bon résultat.
* Peut-être portée comme GameObject attribut par le magicien qui l’utilise (activé et désactivé au besoin).
* Attention : ne pas les créer et détruire au besoin. Activation et désactivation.
* Aura besoin du material et du sortingLayerName du magicien qui l’utilise.