**Limite du terrain et nombre de vies :**

Pour cette partie, nous avons décidé de fixer les limites du terrain aux limites de l'écran. Nous prévoyons une refonte du terrain (avec des attributs de limites) pour plus tard.

Les tests pour savoir si le joueur à dépassé les limites de l'écran se feront dans la méthode update() de GameEngine. Si le joueur quitte l'écran (par le haut, le bas, la gauche ou la droite) toutes les forces qui lui sont appliquées sont annulées, il est également repositionné à un endroit prédéfinit et une vie lui est enlevé. Si le nombre de vie d'un joueur tombe à 0 , la partie est terminée et l'écran du menu principal s'affiche (création d'un écran GameOver par la suite).

**Partie sons :**

Pour les sons deux choix s’offre à nous, utiliser un Clip ou une SourceDataLine. Clip semble être clairement la solution la plus simple et adaptée à notre type de besoin (SourceDataLine est beaucoup plus adapté aux fichiers volumineux qui ne peuvent pas êtres pré-chargés en entier dans la mémoire, il faut donc écrire un buffer lorsque l’on joue le son). Nos fichiers sont de faibles tailles et sont donc pré-chargés dans la mémoire et êtres joués instantanément ou bouclés.

Une nouvelle classe sera créée, “Sound”, elle prendra le fichier son (wav) en paramètre et comportera toutes les méthodes utiles à son exploitation, c’est à dire :

- le lire une fois.

- le lire en boucle.

- le mettre en pause

- reprendre la lecture.

- baisser le volume.

- augmenter le volume.

Pour ne pas surcharger les classes d’attributs, les sons seront des variables locales (sauf pour la musique de fond qui devra être mise en pause lorsque l’on n’est pas dans le jeu).

La classe Sound possède trois attributs :

- un flux audio de lecture (AudioInputStream)

- un clip qui ouvre le flux audio

- un booléen qui indique si le son est en train d’être joué.