Journal de bord : Phase 1 (Groupe7 - Pac Man)

Introduction

Nous avons développé la première phase d'un jeu de tir. Au terme de cette phase, notre code prend la forme d'un jeu de lancer de particule qui fonctionne comme un lance pierre. Les fonctionnalités sont décrites ci-après.

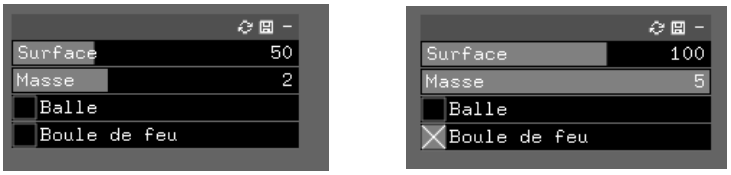
Le joueur peut :

- Définir la particule en modifiant sa surface et sa masse
- Cliquer sur des boutons de particule prédéfinis (balles ou boule de feu)
- Cliquer sur la particule à l'écran pour la contrôler
- Orienter le lance pierre pour contrôler la puissance et la direction du lancer

Démonstration

Menu

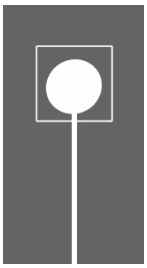
Le menu possède 2 slidebar qui permettent de modifier en temps réel la surface (la taille) de la particule et sa masse. Il est aussi possible de cliquer sur les boutons « Balle » et « Boule de feu » pour préremplir des valeurs de surface et masse correspondantes.



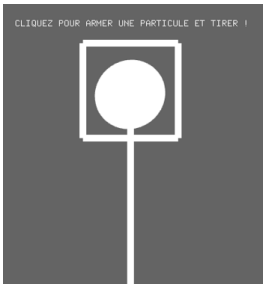
Captures du menu

Lance particule

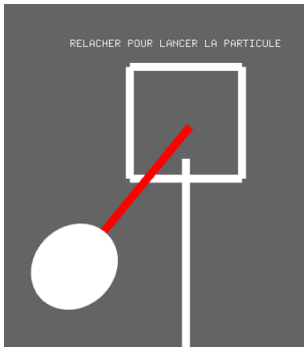
Le lance particule est composé d'un pied, d'une boite et d'une particule ronde prête à être lancée.



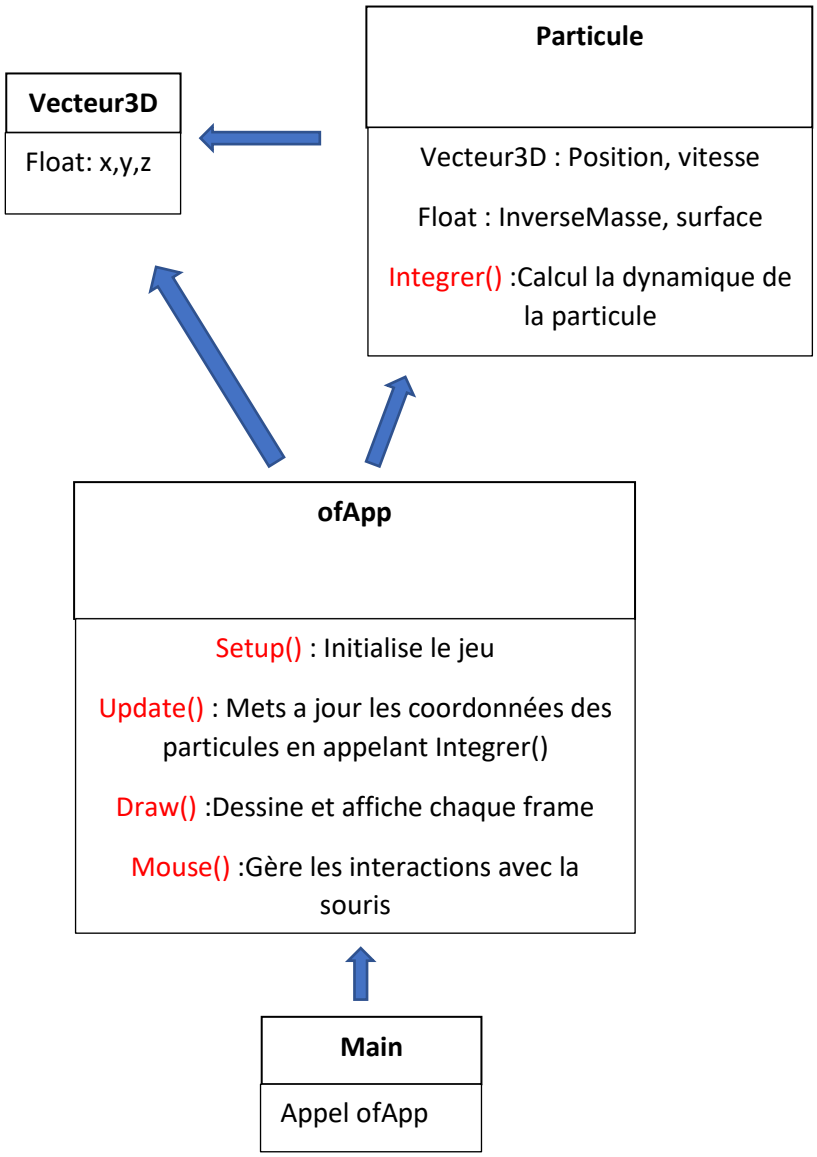
Lorsque la souris passe dessus la particule, il est indiqué de cliquer sur la particule pour la contrôler.



Puis en maintenant la souris cliquée, le joueur peut déplacer la particule et orienter son lancer. Le joueur peut relâcher la souris pour lancer la particule.



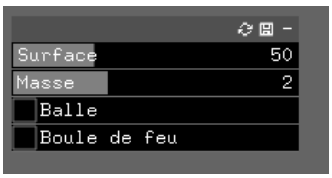
Architecture



Explication du code

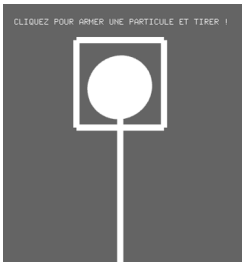
Menu

On utilise la bibliothèque (`#include "ofxGui.h"`) et on affecte les valeurs à des variables globales (`ofxFloatSlider surface;ofxFloatSlider masse`).



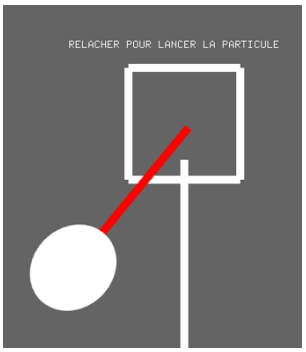
Animation quand la particule est à l'arrêt

Pour les affichages des aides contextuelles, on a utilisé la commande `(mouseX, mouseY)` qui permet de récupérer les coordonnées de la souris et vérifier ci celle-ci se trouve sur la particule ou non. Cette information est stockée dans la variable globale `bool mouseInSquare`. L'affichage gérée par la fonction `Draw()` qui se comporte en conséquence.



Affichage de la particule lorsque le joueur la sélectionne

Dans la fonction `Draw()` la particule ainsi que le trait indicateur rouge sont dessinés si la condition que le joueur clique sur la particule `if(ofGetMousePressed(OF_MOUSE_BUTTON_LEFT) && slingshotActive == true)`.



Création d'une particule

Une particule est créée lorsque le joueur relâche la souris. Cela est géré par la fonction mouseReleased(). La particule est créée avec les paramètres indiqués par le menu et est ensuite stockée dans un dictionnaire de particules.

On créer une particule :

```
Particule nouvelleParticule(trainee, couleur, surface, 1 / masse, position, velocity);
```

On stock la particule :

```
particules.push_back(nouvelleParticule);
```

Gestion de la dynamique des particules

Pour chaque particule, la postion est mise à jour par un appel à la fonction integrer(). Cela est fait a chaque frame par la fonction update() :
`particule.integrer(ofGetLastFrameTime());`

Commentaire

Nous avons fait le choix de mettre en place un système de lance pierre, car cela rend l'expérience utilisateur plus agréable. La qualité du lancer n'est pas parfaite mais les contrôles semblent assez intuitifs. La trajectoire de la particule est orientée selon le trait de départ rouge et plus ce trait est grand plus la vélocité initiale est élevée.

Nous avons aussi implémenté une force de frottement qui ralentit le déplacement de la particule. Pour l'instant cette force dépend uniquement de la surface de l'objet (et de sa vitesse bien sûr). Donc plus une particule est grande plus elle est freinée.

Le choix des menus et du reste de l'affichage est venu naturellement au cours de la rédaction de notre code.