**Rendu**

**1. Hiérarchie des classes dans le module Mobile :**

**1.1. Superclasse Mobile :**

La classe **Mobile** est la classe de base qui contient les attributs généraux et les méthodes communes à toutes les entités mobiles.

**Attributs :** x0, y0 : Coordonnées de la position de l'entité.

a0 : Angle de déplacement de l'entité.

d0 : Distance de déplacement (vitesse).

**Méthodes :**

* Mobile(double x0, double y0, double a0, double d0) : Constructeur de la classe, initialisant les coordonnées et les paramètres de déplacement.
* get\_x0(), get\_y0(), get\_a0(), get\_d0() : Méthodes « getters »  pour récupérer les valeurs des attributs.

Cette classe sert de base à la gestion de la position, direction et distance de déplacement pour toutes les entités mobiles.

**1.2. Classe dérivée Particule :**

**Attributs spécifiques :** c0 : Compteur de cycle de vie de la particule.

**Méthodes spécifiques :**

* Particule(double x0, double y0, double a0, double d0, unsigned c0) : Constructeur pour initialiser une particule avec des valeurs spécifiques.
* lecture(std::istringstream &data) : Méthode permettant de lire et valider les attributs d'une particule.
* set\_c0(unsigned x) : Permet de mettre à jour le compteur c0 de la particule.
* draw() : Méthode pour dessiner la particule à l'écran.
* move() : Gère le déplacement de la particule, prenant en compte la réflexion si la particule sort des limites.

**1.3. Classe dérivée Faiseur :**

**Attributs spécifiques :** r0 : Rayon du faiseur.

nbe0 : Nombre de cercles à construire autour du faiseur.

**Méthodes spécifiques :**

* Faiseur(double x0, double y0, double a0, double d0, unsigned r0, unsigned nbe0) : Constructeur initialisant les attributs spécifiques du faiseur.
* lecture(std::istringstream &data, const std::vector<Faiseur> &V) : Permet de lire et valider les attributs d'un faiseur, y compris la vérification de collisions avec d'autres faiseurs.
* constructionFaiseur() : Génère la liste des cercles du faiseurs.
* draw() : Dessine tous les cercles associés au faiseur.
* move() : Déplace le faiseur selon les règles définies, en tenant compte de la réflexion si nécessaire.

**2. Structuration des données des entités du Modèle:**

**2.1.Quels sont les attributs de la classe Jeu ?**

Dans le fichier jeu.h, la classe Jeu possède les attributs suivants :

| **Type** | **Nom** | **Rôle** |
| --- | --- | --- |
| Chaine | chaine | La chaîne d'articulations à guider. |
| vFaiseurs | vfaiseurs | Vecteur de faiseurs . |
| vParticules | vparticules | Vecteur de particules . |
| Etat | etat\_lecture | État actuel de la lecture du fichier (SCORE, PART, FAIS, etc.). |
| unsigned | score | Le score du jeu. |
| unsigned | nbPart | Nombre actuel de particules. |
| unsigned | nbFais | Nombre actuel de faiseurs. |
| unsigned | nbArtic | Nombre actuel d’articulations dans la chaîne. |
| Mode | mode | Le mode du jeu (CONSTRUCTION ou GUIDAGE). |
| bool | lecture\_ok\_ | Indique si la lecture du fichier s’est bien passée. |
| Status | status | Statut du jeu (ONGOING, WON ou LOST). |

### **2.2. Où et comment sont mémorisés les différents ensembles que doit gérer le jeu ?**

|  |  |  |
| --- | --- | --- |
| **Ensemble à gérer** | **Où c’est mémorisé** | **Comment c’est organisé** |
| Particules | vparticules | std::vector<Particule> contenant tous les objets Particule. |
| Faiseurs (ensemble de cercles) | vfaiseurs | std::vector<Faiseur> où chaque Faiseur gère ses propres cercles. |
| Articulations de la chaîne | chaine.articulations() | Stockés dans l’objet Chaine,sous forme de vecteur de Cart. |

**Les ensembles** sont mémorisés **dans des vecteurs** appartenant à la classe Jeu

**3. Types utilisés dans tools: state :** direction (FORWARD, BACKWARD).

**S2d :** point 2D (x, y).

**Cart :** coordonnées cartésiennes .

**Pol :** coordonnées polaires.

**Cercle :** centre (Cart) + rayon (R) .

**Usage d’IA :** Deepseek pour la compréhension de quelques erreurs et des méthode de gui.cc .

**Activité :**

Module tools :

S2d / Pol / Cart / angleNormalise/ reflect : ensemble

Cercle/ intersection : Adam

Distance / operator+= / operator+/ opp/ drawCircle/ drawLine : Yassine

Module Graphic : Yassine

Module mobile : Adam

Module chaine : Yassine

Module jeu : ensemble

Module Gui : Yassine