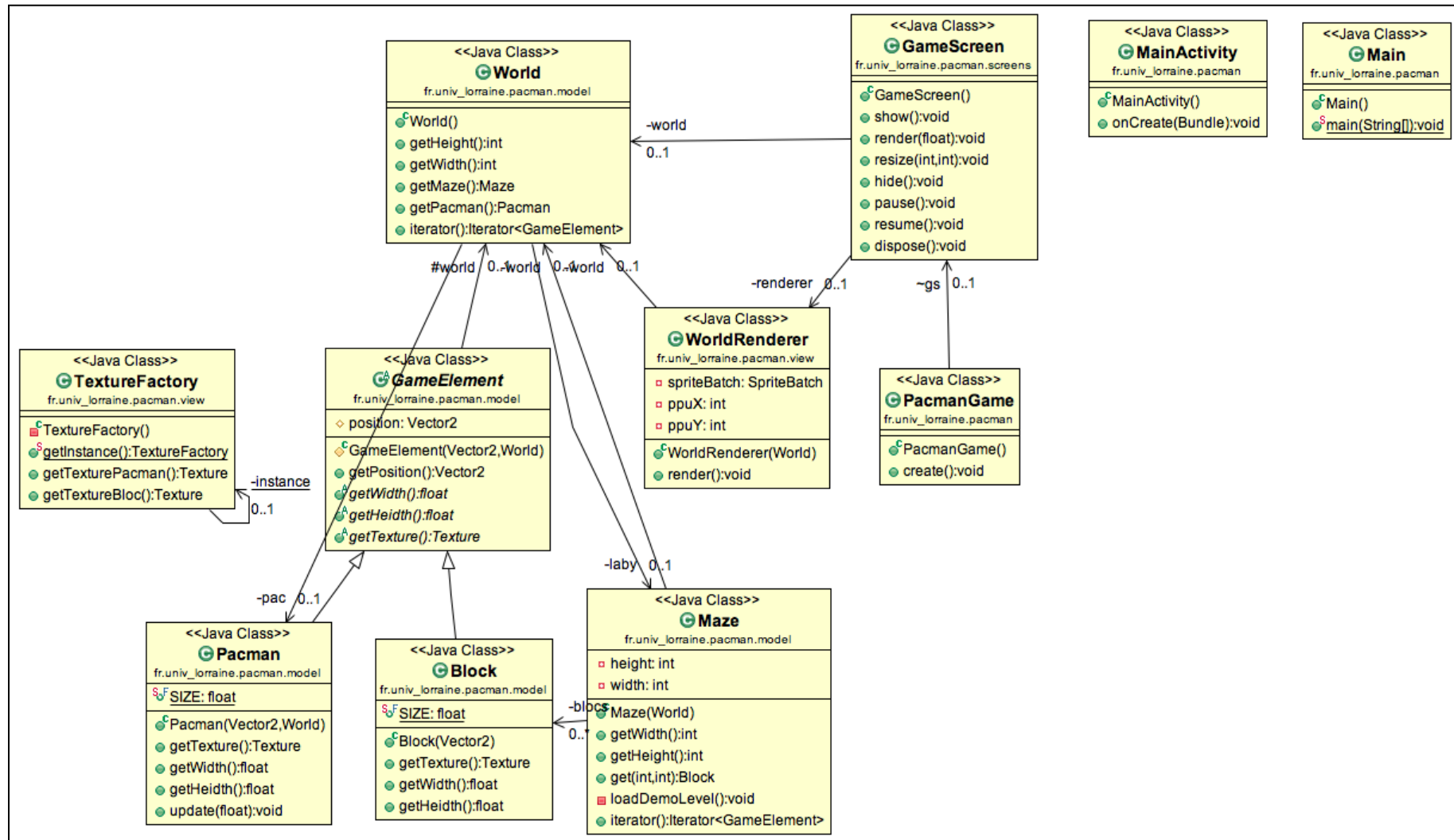


Implémenter l'ensemble des classes du diagramme de classes.



Les classes Maze et World doivent être itérables pour permettre l'utilisation de :

```
for (GameElement ge : Maze ou World) { ... }
```

Cela permet de rajouter des GameElement dans Maze ou dans World sans pour autant modifier le code qui les parcourt. Cela peut s'implémenter de deux façons différentes.

La première se fait par l'intermédiaire d'une structure itérable. Exemple :

```
public class A implements Iterable<B> {
    private B b1, b2, b3;
    ...
    @Override
    public Iterator<B> iterator() {
        ArrayList<B> l = new ArrayList<B>();
        l.add(b1);
        l.add(b2);
        l.add(b3);
        return l.iterator();
    }
}
```

Il est également possible de définir son propre itérateur sans passer par une structure intermédiaire. Exemple :

```
public class A implements Iterable<B> {
    private B b1, b2, b3;
    ...
    @Override
    public Iterator<B> iterator() {
        return new Iterator<B>() {
            private int i = 0 ;

            @Override
            public boolean hasNext() {
                return i < 2;
            }

            @Override
            public GameElement next() {
                if (!hasNext())
                    throw new NoSuchElementException("plus de B");

                i++ ;
                switch (i)
                {
                    case 0 : return b1;
                    case 1 : return b2;
                    default : return b3;
                }
            }
        };
    }
}
```