**Event :**

Toutes classes évènement hérite de la classe mère « Event ».

Event a 2 méthodes virtual qui lui permette de définir si ce sera un évènement spontané ou un évènement qui dur dans le temps, grâce aux méthodes :

* applyEvent() : évènement spontané
* applyEvent(int time) : évènement avec une durée (time)

La méthode launchEvent(int time) permet de lancer le bon appel d’évènement selon le temps donné en paramètre :

void Event::lauchEvent(int time)

{

if (m\_runtime == -1) applyEvent();

else applyEvent(time);

}

Ici int time correspond au temps de la classe : time().

S’il est précisé dans le constructeur, il sera différent de -1, sinon un constructeur qui ne demande que le nom définira m\_time par défaut a -1 :

* évènement spontané :

Event::Event(QString name): m\_name(name), m\_runtime(-1)

{

}

* évènement avec une durée :

Event::Event(QString name, int time): m\_name(name), m\_runtime(time)

{

}

La méthode reverse() sert a inversé un évènement temporel. Exemple :

Evènement :

Nom : Fête national

Action : Augmente la popularité du bar

Durée : 1 jour ( reverse() sera appelé a la fin de la journée )

**Classe métier :**

Tous les attributs des classes métiers qui ont besoin d’apparaitre à l’écran sont des Q\_PROPERTY, ce qui permettra leurs affichages avec Qt.