if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_VehicleCombat());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_RangedCombat());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_MeleeCombat());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_Engage());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_VehicleBoarded());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_VehicleExited());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_StopCombat());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_StopEngage());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_MeleeWeaponEquipped());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_RangedWeaponEquipped());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_EnemyInMeleeRange());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_OutOfAmmo());
}

if (in deciding.activity_switch.engaging) {
  broadcastEvent(new ev_NoEnemyInMeleeRange());
}
idle

```
if (npc.canWander()) {
  broadcastEvent(new ev_Wander());
}
```

```
if (npc.canTalk()) {
  broadcastEvent(new ev_PlaySound(Sounds.IDLE));
}
```

```
if (random==1) {
  startIdleTimer();
}
```
not_searching

startSearchTimer();

characterMemorizer.getEnemiesOutOfView();

[wandering.size==0 & & npc.canWander()]
broadcastEvent(new ev_StartWander());

[wandering.size>0]
broadcastEvent(new ev_StartExecuteSearch());
broadcastEvent(new ev_StopExecuteSearch());

[wandering.size<=0]
if ev_StopSearch

ev_SearchTimerUp; broadcastEvent(new ev_EnemyNotFound());
normal

ev_AreaThreat

attacker_nearby

ev_AreaThreatAbated

flee_all

broadcastEvent(new ev_TakeCover())
broadcastEvent(new ev_StopTakeCover())

ev_AreaThreat

ev_Safe

flee_nearby

broadcastEvent(new ev_EvacuateArea())
broadcastEvent(new ev_StopEvacuateArea())

ev_AreaThreatAbated

ev_Safe

ev_AreaThreat

ev_AreaThreatAbated

ev_Safe

ev_Safe

ev_AreaThreat

ev_AreaThreatAbated

ev_Safe