

// Actions in Viv are defined as activities an NPC can take in certain circumstances, with strong use of metadata to allow other NPCs to observe, remember, comment on, and react to each others' actions. Here are a few baseline actions that characters in the simulation can perform.

action hunting-replicants:

gloss: "hunts renegade replicants"

report: "{@runner}, a Blade Runner, is hunting for renegade replicants to retire in the grimy streets of Los Angeles."

roles:

runner: *initiator*

associations:

runner: *productive, dangerous*

hearer: *interesting*

preconditions:

@runner.job?.profession == "Blade Runner"

@runner fits atWork

action walk-in-rain from **street-action:**

gloss: "walks through the rain"

report: "{@person} walk through the rainy streets."

action dodge-vehicle from **street-action:**

gloss: "dodges a vehicle"

report: "{@person} dodges a reckless vehicle careening through the streets."

salience:

person: #NOTEWORTHY

associations:

person: *stressful*

action watch-animal from **street-action:**

gloss: "stops in awe to watch an animal in the street"

report: "{@person} stops in awe to watch a rare animal being led through the street."

tags: *conspicuous, cool, interesting*

join preconditions:

5%

salience:

person: #MEMORABLE

witness: #MEMORABLE

hearer: #NOTEWORTHY

action watch-blimp from **street-action:**

gloss: "watches an overhead blimp"

report: "{@person} watches an advertizing blimp circling overhead."

```
// Where Viv gets more interesting is in allowing for emergent chains of cause and effect. Here, a series of more specialized actions allows a character to experience trauma that they later wish to escape by stealing happy memories from a memory lab. The scenario defines one example of trauma (gang-assault) but any event tagged "traumatic" could trigger the next step in this sequence (relieve-trauma). A robust library of these templates could easily be mixed and matched by the system to generate bespoke narrative arcs, driving NPC behavior and procedural crimes to investigate.
```

action **gang-assault**:

```
gloss: "leads a gang attack on a lone victim"
```

```
report: "At {@this.where}, gang leader {@leader} leads an assault (assisted by fellow gang members {*gang_member}) on an innocent victim, {@victim}."
```

```
roles:
```

```
  leader: initiator
```

```
  1-5 gang_member: partner
```

```
  victim: recipient
```

```
  0-9 witness: bystander
```

```
preconditions:
```

```
  @leader.job?.profession == "Gang Member"
```

```
  @victim.job?.profession != "Gang Member"
```

```
  @victim.age >= 18
```

```
  @leader.charge(@victim) < #HIGH
```

```
  // Ensure all NPCs cast as "gang members" have the right profession
```

```
  loop *gang_member as @g:
```

```
    @g.job?.profession == "Gang Member"
```

```
  end
```

```
  // Ensure that impartial witnesses are not gang members
```

```
  loop *witness as @w:
```

```
    @w.job?.profession != "Gang Member"
```

```
  end
```

```
salience:
```

```
  leader: #NOTEWORTHY
```

```
  gang_member: #NOTEWORTHY
```

```
  victim: #MEMORABLE
```

```
tags: harmful, criminal, conspicuous
```

```
associations:
```

```
  leader: fun, productive, interesting
```

```
  gang_member: fun, productive, interesting
```

```
  victim: stressful, traumatic
```

```
embargoes:
```

```
  victim for 2 weeks
```

```
effects:
```

```
  // Victim loathes the lead attacker.
```

```
  @victim.charge(@leader, -#DEFINING)
```

```

@victim.spark(@leader, -#DEFINING)
loop *gang_member as @g:
  // Victim loathes each gang member.
  @victim.charge(@g, -#DEFINING)
  @victim.spark(@g, -#DEFINING)
  // Gang members bond with gang leader.
  @g.charge(@leader, #EVERYDAY)
  @leader.charge(@g, #EVERYDAY)
end
// Witnesses dislike the gang leader.
loop *witness as @w:
  @w.charge(@leader, -#SMALL)
  @w.spark(@leader, -#SMALL)
end
reactions:
  // Victim gets the heck out of there
  if @victim.residence:
    queue goto:
      bindings:
        traveler: @victim
        destination: @victim.residence.dwelling
      urgent: true
      when:
        between 1 minute and 8 hours
      priority: 1000
    end
  end

```

action **relive-trauma**:

```

gloss: "relives past trauma"
report: "{@traumatized}, haunted by the memory of past trauma, is unable to
stop thinking about the disturbing events."
roles:
  traumatized: initiator
  trauma from @traumatized.search("traumatic", {recipient: @traumatized}):
action, subject
preconditions:
  // Being in the same place as the place where trauma occurred can
  re-trigger it.
  {!@traumatized.traits.unflappable &&
  !@traumatized.traits.stable}
  @traumatized.status.location->status.characters.length == 1
  @traumatized.knowledge.actions[@trauma.id].salience >= #NOTEWORTHY
saliences:
  traumatized: @traumatized.knowledge.actions[@trauma.id].salience
tags: traumatic, stressful, embarrassing

```

```
effects:
  @traumatized.knowledge.actions[@trauma.id].salience += #MEMORABLE
reactions:
  if {@traumatized.knowledge.actions[@trauma.id].salience >= #DEFINING}:
    queue plan-memory-theft:
      bindings:
        thief: @traumatized
        trauma: @trauma
      end
end
```

```
special action plan-memory-theft:
gloss: "plans to steal a happy memory"
report: "{@thief} hatches a plan to steal some happy memories from a memory
lab where they are engineered."
roles:
  thief: initiator
  trauma: precast, action
  venue from @thief.world->businesses: location
preconditions:
  @venue.businessType == "Memory Lab"
saliences:
  thief: #NOTEWORTHY
reactions:
  queue sneaky-goto:
    bindings:
      traveler: @thief
      origin: @thief.status.location
      destination: @venue
    priority: 1001
  queue steal-memory:
    bindings:
      thief: @thief
      venue: @venue
    urgent: true
    priority: 1000
    where: @venue
```

```
special action steal-memory:
gloss: "steals a happy memory"
report: "{@thief} breaks into {@venue} and steals a recording of some happy
memories."
roles:
  thief: initiator
  venue: precast, location
  0-2 witness: bystander
```

```
// Because this is tagged as criminal, it is an action that might be reported by witnesses and which NPC or player detectives can investigate (by interviewing suspects, tracing movements, etc.)
```

```
tags: criminal
```

```
associations:
```

```
  thief: stressful, shameful, dangerous
```

```
  witness: interesting
```

```
  hearer: interesting
```

```
saliences:
```

```
  thief: #MEMORABLE
```

```
  witness: #NOTEWORTHY
```

```
  hearer: #NOTEWORTHY
```

```
reactions:
```

```
  queue goto:
```

```
    bindings:
```

```
      traveler: @thief
```

```
      destination: @thief.residence.dwelling
```

```
    urgent: true
```

```
    priority: 1000
```

```
  queue replay-memory:
```

```
    bindings:
```

```
      viewer: @thief
```

```
special action replay-memory:
```

```
  gloss: "plays an external memory to assuage past trauma"
```

```
  report: "To assuage past trauma, {@viewer} plays back recordings of happy memories, watching them over and over until starting to feel a little happier."
```

```
  roles:
```

```
    viewer: initiator
```

```
    trauma from @viewer.search("traumatic", {recipient: @viewer}): action
```

```
associations:
```

```
  viewer: relaxing, shameful, embarrassing
```

```
saliences:
```

```
  viewer: #NOTEWORTHY
```

```
effects:
```

```
  @viewer.knowledge.actions[@trauma.id].salience -= #MEMORABLE
```