# Deceit_in_Detail_Tactic (marshaled)
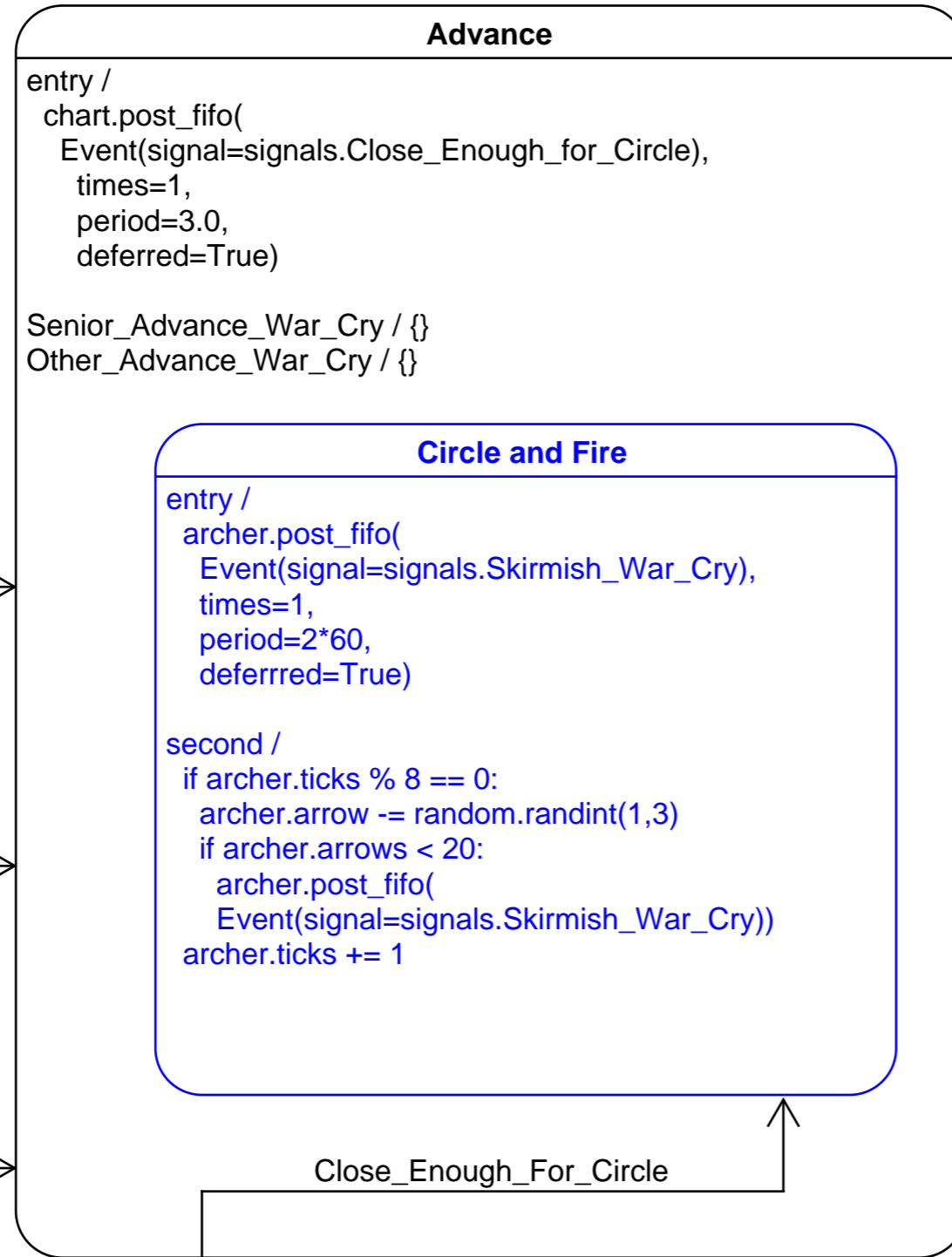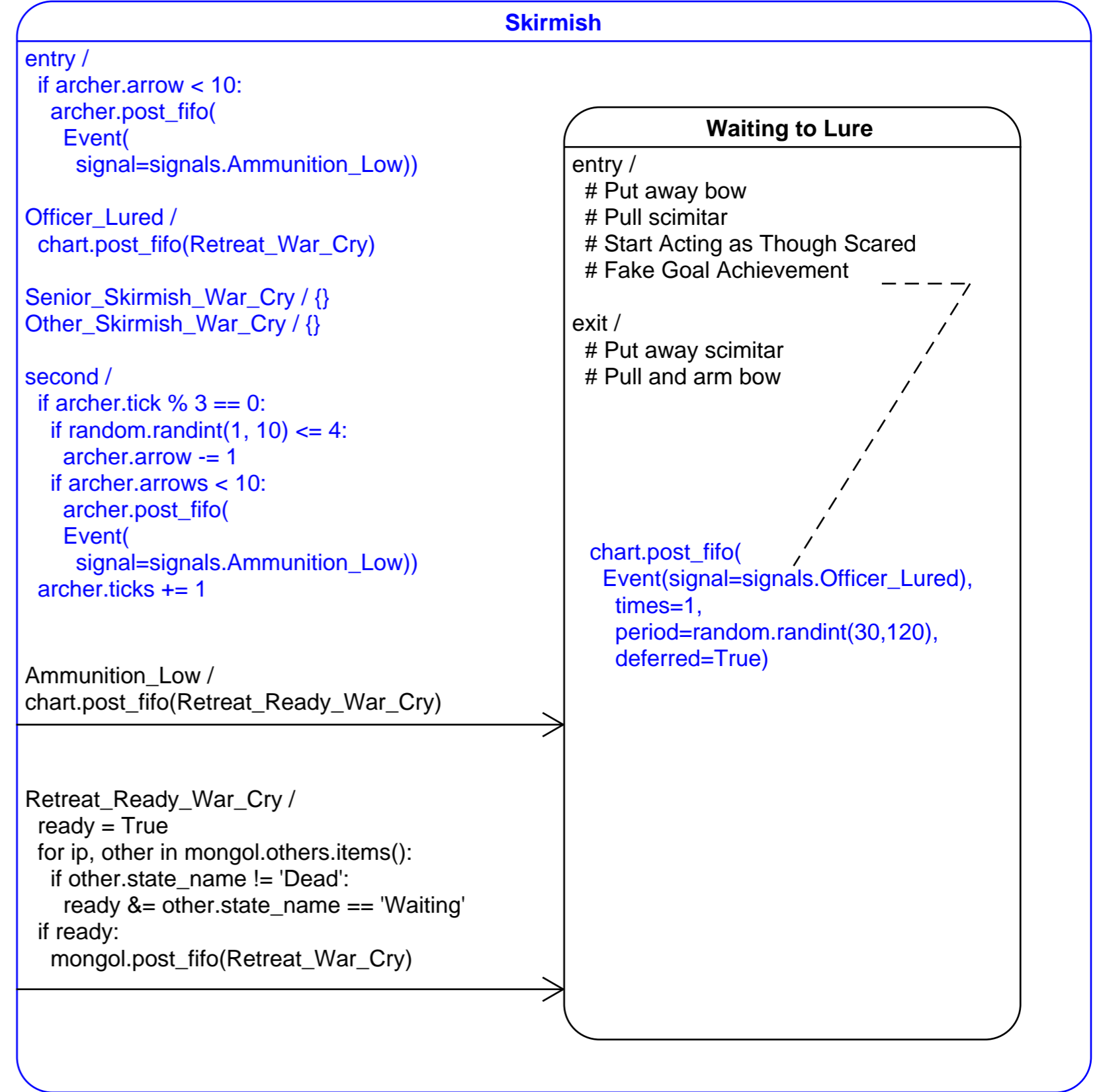
entry /
  # Identify next marshal point
  # Load horse with arrows
  mongol.arrows = 60
  archer.ticks = 0
second /
  archer.ticks += 1

Senior_Advance_War_Cry \ chart.post_fifo(Event(signal=signals.Advance_War_Cry))
Senior_Skirmish_War_Cry \ chart.post_fifo(Event(signal=signals.Skirmish_War_Cry))
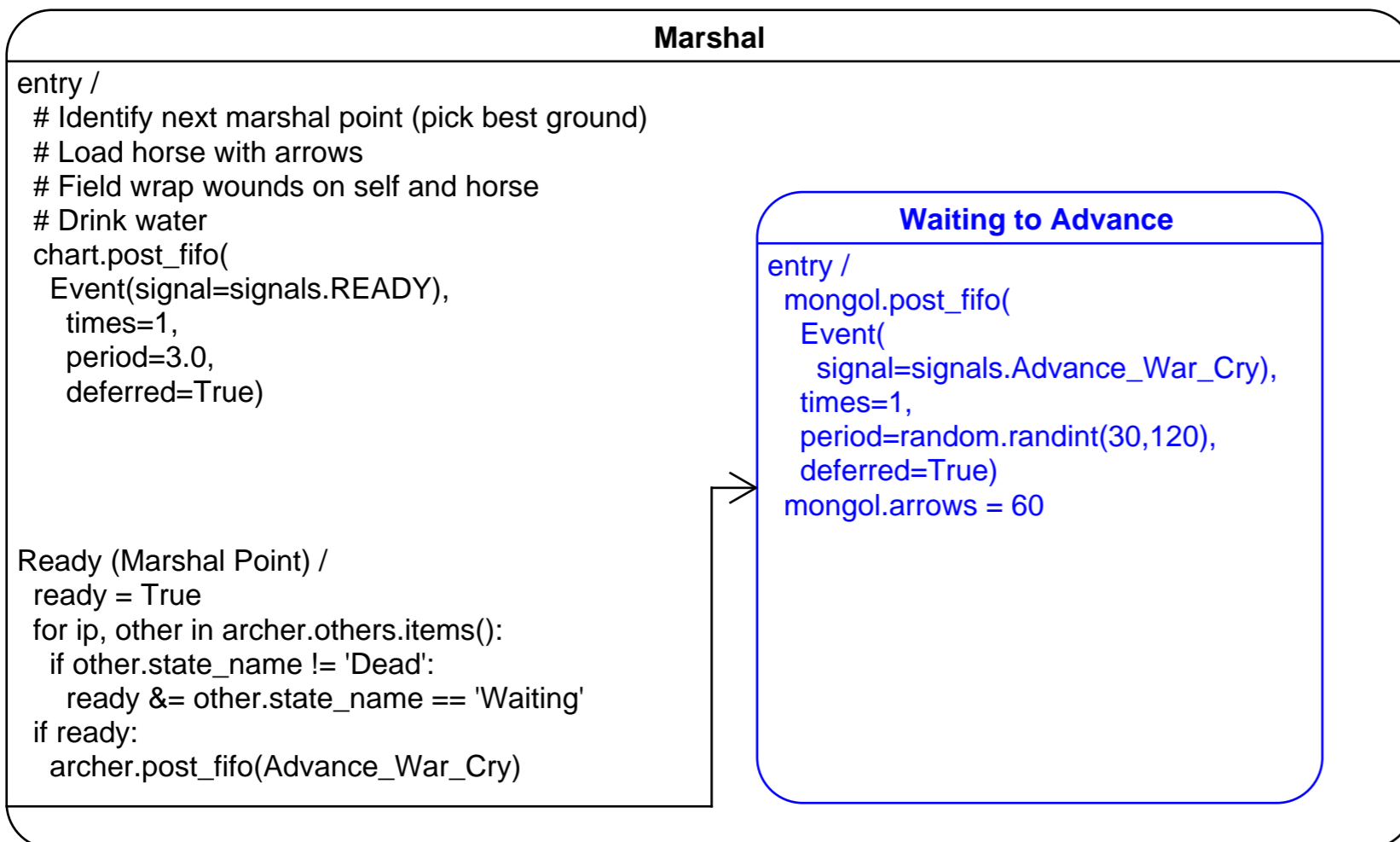Senior_Retreat_War_Cry \ chart.post_fifo(Event(signal=signals.Retreat_War_Cry))

## Advance

entry /
  chart.post_fifo(
    Event(signal=signals.Close_Enough_for_Circle),
      times=1,
      period=3.0,
      deferred=True)

Senior_Advance_War_Cry / {}
Other_Advance_War_Cry / {}

### Circle and Fire

entry /
  archer.post_fifo(
    Event(signal=signals.Skirmish_War_Cry),
      times=1,
      period=2*60,
      deferrred=True)

second /
  if archer.ticks % 8 == 0:
    archer.arrow -= random.randint(1,3)
    if archer.arrows < 20:
      archer.post_fifo(
      Event(signal=signals.Skirmish_War_Cry))
  archer.ticks += 1

Advance_War_Cry as e /
  mongol.yell(e)
  for ip, other in mongol.others.items():
    other.dispatch(e)

Other_Advance_War_Cry as e, ip: /
  mongol.other[ip].dispatch(e)

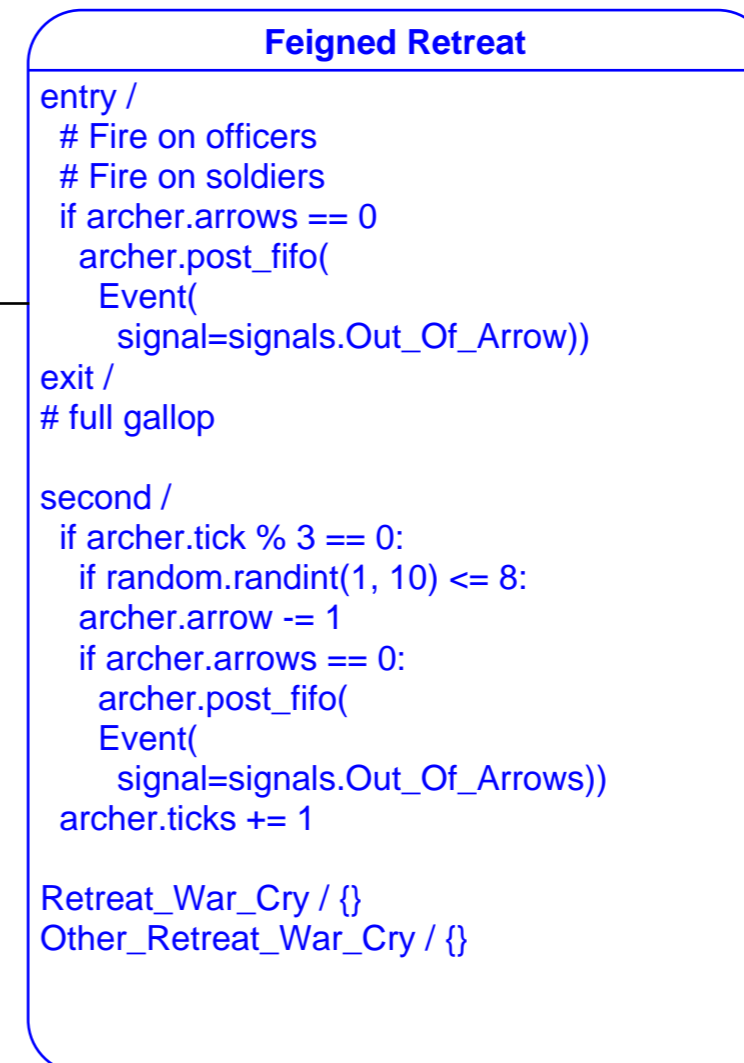Close_Enough_For_Circle

## Skirmish

entry /
  if archer.arrow < 10:
    archer.post_fifo(
      Event(
        signal=signals.Ammunition_Low))

Officer_Lured /
  chart.post_fifo(Retreat_War_Cry)

Senior_Skirmish_War_Cry / {}
Other_Skirmish_War_Cry / {}

second /
  if archer.tick % 3 == 0:
    if random.randint(1, 10) <= 4:
      archer.arrow -= 1
    if archer.arrows < 10:
      archer.post_fifo(
      Event(
        signal=signals.Ammunition_Low))
  archer.ticks += 1

### Waiting to Lure

entry /
  # Put away bow
  # Pull scimitar
  # Start Acting as Though Scared
  # Fake Goal Achievement

exit /
  # Put away scimitar
  # Pull and arm bow

  chart.post_fifo(
    Event(signal=signals.Officer_Lured),
      times=1,
      period=random.randint(30,120),
      deferred=True)

Ammunition_Low /
  chart.post_fifo(Retreat_Ready_War_Cry)

Retreat_Ready_War_Cry /
  ready = True
  for ip, other in mongol.others.items():
    if other.state_name != 'Dead':
      ready &= other.state_name == 'Waiting'
  if ready:
    mongol.post_fifo(Retreat_War_Cry)

Skirmish_War_Cry /
  mongol.yell(Skirmish_War_Cry)

Other_Skirmish_War_Cry as e, ip \
  mongol.other[ip].dispatch(e)

## Marshal

entry /
  # Identify next marshal point (pick best ground)
  # Load horse with arrows
  # Field wrap wounds on self and horse
  # Drink water
  chart.post_fifo(
    Event(signal=signals.READY),
      times=1,
      period=3.0,
      deferred=True)

Ready (Marshal Point) /
  ready = True
  for ip, other in archer.others.items():
    if other.state_name != 'Dead':
      ready &= other.state_name == 'Waiting'
  if ready:
    archer.post_fifo(Advance_War_Cry)

### Waiting to Advance

entry /
  mongol.post_fifo(
    Event(
      signal=signals.Advance_War_Cry),
      times=1,
      period=random.randint(30,120),
      deferred=True)
  mongol.arrows = 60

Out_Of_Arrows

## Feigned Retreat

entry /
  # Fire on officers
  # Fire on soldiers
  if archer.arrows == 0
    archer.post_fifo(
      Event(
        signal=signals.Out_Of_Arrow))
exit /
  # full gallop

second /
  if archer.tick % 3 == 0:
    if random.randint(1, 10) <= 8:
    archer.arrow -= 1
    if archer.arrows == 0:
      archer.post_fifo(
      Event(
        signal=signals.Out_Of_Arrows))
  archer.ticks += 1

Retreat_War_Cry / {}
Other_Retreat_War_Cry / {}

Retreat_War_Cry as e /
  archer.yell(e)
  for ip, other in mongol.others.items()
    other.dispatch(e)

Other_Retreat_War_Cry as e, ip:
  mongol.other[ip].dispatch(e)