Deceit_in_Detail_Tactic (marshaled)

entry /
 # Identify next marshal point
 # Load horse with arrows
 archer.arrows = HorseArcher.MAXIMUM_ARROW_CAPACITY
 archer.ticks = 0
 archer.post_fifo(Event(signal=signals.Second, times=0, period=archer.to_time(1.0), deferred=True)
second /
 archer.ticks += 1
exit / archer.cancel_event(Event(signal=signals.Seconed))
Senior_Advance_War_Cry as e\ archer.post_fifo(e)
Senior_Skirmish_War_Cry as e\ archer.post_fifo(e)
Senior_Retreat_War_Cry as e \ archer.post_fifo(e)

Skirmish

entry /
 archer.post_fifo(
  Event(signal=signals.Officer_Lured),
  times=1,
  period=archer.to_time(
   random.randint(40, 200)),
  deferred=True
 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

Ammunition_Low /
 chart.post_fifo(
  Event(signal=signals.Retreat_Ready_War_Cry))

Retreat_Ready_War_Cry /
 ready = True
 for ip, other in mongol.others.items():
  if other.state_name != 'ead':
   ready &= other.state_name == 'Waiting'
 if ready:
  # let's make sure Gandbold isn't a chicken
  archer.post_fifo(
   Event(signal=signals.Retreat_War_Cry),
   times=1,
   period=archer.to_time(
    random.randint(10, 30)),
   deferred=True)

exit /
 archer.cancel(Event(
  signal=signals.Ammunition_Low))
 archer.cancel(Event(

Officer_Lured /
 archer.scribble("Knight Charging")
 archer.post_fifo(
  Event(signal=signals.Retreate_Ready_War_Cry))

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)

Advance

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

exit/ archer.cancel_events(
 Event(signal=signals.Close_Enough_for_Circle))

Senior_Advance_War_Cry / {}
Other_Advance_War_Cry / {}

Circle and Fire

second /
 if archer.ticks % 6 == 0:
  archer.arrow -= random.randint(1,3)
  if archer.arrows < 20:
   archer.post_fifo(
    Event(signal=signals.Skirmish_War_Cry))
 archer.ticks += 1

Close_Enough_For_Circle

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

Other_Advance_War_Cry as e, ip: /
 archer.post_fifo(
  Event(
   signal=/
    signals.Advance_War_Cry))
 archer.other[ip].dispatch(e)

Other_Skirmish_War_Cry as e, ip \
 archer.post_fifo(
  Event(
   signal=/
    signals.Skirmish_War_Cry))
 mongol.other[ip].dispatch(e)

Skirmish_War_Cry /
 archer.yell(Skirmish_War_Cry)

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)