## entry / archer.scribble("halt horse") archer.scribble("identify next marshal point") archer.scribble("field wrap wounds on self and horse") archer.scribble("drink water") archer.arrows = HorseArcher.MAXIMUM\_ARROW\_CAPACITY chart.post\_fifo( Event(signal=signals.READY), times=1, period=archer.to\_time(60), deferred=True) waitin entry / archer.yell(Event( signal=signals.Other\_Re payload=archer.name)) ready = True archer.snoop\_scribbel("{} format(archer.name, arc time\_to\_wait = random.ra

Ready

## waiting\_to\_advance archer.yell(Event( signal=signals.Other\_Ready\_War\_Cry, payload=archer.name)) ready = True archer.snoop\_scribbel("{} has {} arrows". \ format(archer.name, archer.arrows) time\_to\_wait = random.randint(130,300) for name, other in archer.others.items(): if other.dead() is not True: ready &= other.waiting() else: archer.snoop\_scribble( "{} thinks {} is dead". format(archer.name, name) if ready is False: archer.snoop\_scribble( "{} is impatient he will attack in {} seconds". format(archer.name, time\_to\_wait) archer.post\_fifo( Event( signal=signals.Advance\_War\_Cry), times=1, period=random.randint(time\_to\_wait), deferred=True) else: archer.snoop\_scribble( "{} thinks unit is ready to attack". \ format(archer.name)) archer.post\_fifo( Event(signal=signals.Advance\_War\_Cry)) exit / archer.cancel\_events( Event(signal=signals.Advance\_War\_Cry))

