>>> from ants import \*

>>> beehive, layout = Hive(AssaultPlan()), dry\_layout

>>> dimensions = (1, 9)

>>> gamestate = GameState(None, beehive, ant\_types(), layout, dimensions)

>>> thrower = ThrowerAnt()

>>> ant\_place = gamestate.places["tunnel\_0\_0"]

>>> ant\_place.add\_insect(thrower)

>>> #

>>> # Testing nearest\_bee

>>> near\_bee = Bee(2) # A Bee with 2 armor

>>> far\_bee = Bee(3) # A Bee with 3 armor

>>> near\_place = gamestate.places['tunnel\_0\_3']

>>> far\_place = gamestate.places['tunnel\_0\_6']

>>> near\_place.add\_insect(near\_bee)

>>> far\_place.add\_insect(far\_bee)

>>> nearest\_bee = thrower.nearest\_bee(gamestate.beehive)

>>> thrower.nearest\_bee(gamestate.beehive) is far\_bee

###########

# Layouts #

###########

def wet\_layout(queen, register\_place, tunnels=3, length=9, moat\_frequency=3):

    """Register a mix of wet and and dry places."""

    for tunnel in range(tunnels):

        exit = queen

        for step in range(length):

            if moat\_frequency != 0 and (step + 1) % moat\_frequency == 0:

                exit = Water('water\_{0}\_{1}'.format(tunnel, step), exit)

            else:

                exit = Place('tunnel\_{0}\_{1}'.format(tunnel, step), exit)

            register\_place(exit, step == length - 1)

def dry\_layout(queen, register\_place, tunnels=3, length=9):

    """Register dry tunnels."""

    wet\_layout(queen, register\_place, tunnels, length, 0)

>>> from ants import \*

>>> beehive, layout = Hive(AssaultPlan()), dry\_layout

>>> dimensions = (1, 9)

>>> gamestate = GameState(None, beehive, ant\_types(), layout, dimensions)

>>> #

>>> place = gamestate.places['tunnel\_0\_4']

>>> ant = FireAnt(1) # Create a FireAnt with 1 armor

>>> place.add\_insect(ant) # Add a FireAnt to place

>>> ant.place is place

? True

-- OK! --

>>> place.remove\_insect(ant) # Remove FireAnt from place

>>> ant.place is place # Is the ant's place still that place?

? False

-- OK! --