Note: This is only the compulsory part, look at pdf for more.

Setup:

  Need Three entities -> player 1, player 2, destination

  Placed randomly on the cartesian plane (-800, 800)

Movement

  Player 1 and 2 will take turns moving to the fixed destination

  Main Restriction:

    1. Players can only move on the hypothnuse of a right-angled triangle

    2. Lengths of the sides of the triangle must form a primative pythagorean triple

      -> a primative pythagorean triple is a triangle that all sides are coprime.

      -> coprime means that the highest common divisor is 1.

        For example: (3, 4, 5)'s hcf is 1

      -> We get a list of these triples (first 127)

    3. the non-hypothenuse sides, a and b, have to be parallel to the x and y axis of the cartesian plane.

  Figure out how to make a reusable means of achieving translations.

    -> Can't use trigonometry

    -> split into 1-8 directions, by splitting each quadrent into 2

      Eg. 1 will be 0-45 degrees in cartesian

        -> cartesian means 0 degrees is East and the positive direction is counter-clockwise.

    -> a player can request to move x units in direction y

      Eg. 5 <space> 3, where 5 is the distance and 3 is the direction.

    -> Use the pythagorean triples

      Idea: get the hypothenuse from the list that matches the distance, then

    -> Also means left side will be b,a right will be a,b

Dictionaries:

  The players and destination will be dictionaries, called player\_one, player\_two, destination

    -> Player dictionaries will contain -> current coordinates, distance from destination, midpoint coordinantes to other player, gradient of the line connecting player to destination and personal space buffer of 10 units

    -> In the destination dictionary should only contain current coordinatines and personal space buffer

    -> Both distance and gradient are to 1 decimal place.

Print:

  print current information in dictionaries

    -> optional termcolor

Winning:

  1. player wins if their coordinates are within the destination personal space buffer

  2. if they reach the other players personal space buffer

User Input:

  if a number without a triple is selected use closest lower triple number

  reject negative Lengths and directions beyond 1, 8

More:

  Add comments and commits

  Submit through a presentation

  Use functions, no import except numpy, random and time

ReplyReply allForward