L2 PROBLEM 8 (1/1 point)

Suppose we wanted to create a class `PolarBearDrunk`, a drunk polar bear who moves randomly along the x and y axes taking large steps when moving South, and small steps when moving North.

```
class PolarBearDrunk(Drunk):
    def takeStep(self):
        # code for takeStep()
```

Which of the following would be an appropriate implementation of `takeStep()`?

1. Option A)

```
directionList = [(0.0, 1.0), (1.0, 0.0), (-1.0, 0.0), (0.0, -1.0)]
myDirection = random.choice(directionList)
if myDirection[0] == 0.0:
    return myDirection + (0.0, -0.5)
return myDirection
```

2. Option B)

```
directionList = [(0.0, 0.5), (1.0, -0.5), (-1.0, -0.5), (0.0, -1.5)]
return random.choice(directionList)
```

3. Option C)

```
directionList = [(0.0, 0.5), (1.0, 0.0), (-1.0, 0.0), (0.0, -1.5)]
return random.choice(directionList)
```

4. Option D)

```
directionList = [(0.0, 1.0), (1.0, 0.0), (-1.0, 0.0), (0.0, -1.0), (0.0, -1.0)]
return random.choice(directionList)
```

- ( ) Option A)
- ( ) Option B)
- (•) Option C) ✔
- ( ) Option D)

EXPLANATION:

Option A) is incorrect because it produces tuples of length 4. The logic is otherwise correct, but it should be written as:

```
directionList = [(0.0, 1.0), (1.0, 0.0), (-1.0, 0.0), (0.0, -1.0)]
myDirection = random.choice(directionList)
if myDirection[0] == 0.0:
    return (myDirection[0], myDirection[1] - 0.5)
return myDirection
```

Option B) is incorrect because it produces directions not along axes.

Help

Finally Option D) is incorrect because it produces a bias toward moving South, but does not alter step size.

Check | Hide Answer

Show Discussion

New Post