Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: ______________________________

Numerical Python

1. As we discussed in class, Python provides us with the built-in data structures of tuples and lists. Unfortunately, these are not intended to be used as mathematical data structure. However, we can convert these structures to the `NumPy` array type, and arrays can perform convenient mathematical operations.

5 (a) Import the `numpy` module, but for convenience, alias it to `np`.

5 (b) Assume that you have two tuples, `me = (3, 2)` and `opponent = (6, 5)`. Convert these types to arrays and call them `me_arr` and `opponent_arr`, respectively.

5 (c) Write Python code (using `me_arr` and `opponent_arr`) to compute the vector from `me` to `opponent`, and call the result `v`.

5 (d) There are still many libraries that cannot work with Numerical Python. Often, you will need to convert `numpy` arrays back to lists and tuples. For instance, you might have code like:

```
xx = np.array([1, 2, 3, 4, 5])
yy = np.array((3, 5)
```

Write Python code to convert `xx` and `yy` back into a regular `list`, `x`, and a regular `tuple`, `y`. Although we haven't covered this in class, see the built-in functions and find the two that will do what you want.[1]

[1] http://docs.python.org/library/functions.html

Newtonian Physics

2. In class, you worked with one-dimensional kinematic equations. This means that your formula allows allows movement along a single line (such as going left or right). We will now extend these kinematic equations to two (or more) dimensions, using vectors. The formulas thus become:

$$\vec{p} \leftarrow \vec{p_o} + \vec{v_o}t + \frac{1}{2}\vec{a}t^2 \tag{1}$$

$$\vec{v} \leftarrow \vec{a}t \tag{2}$$

Note that the only thing that has changed is that the variables have been re-written to be a vector instead of scalar! Otherwise, the formulas are exactly the same.

5 (a) Compute p when $p_o = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $v_o = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $a = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, and $t = 0.4$.

5 (b) Compute v given the parameters of the previous part.

10 (c) Write Numerical Python code to perform the calculations in part a and b.

3. When time intervals are small (as in the case when the frame rate is high), the contribution of acceleration to position will be tiny. In this case, we can use the Newton-Euler-1 integration update approximation. Given the vector equations:

$$\vec{p} \leftarrow \vec{p_o} + \vec{v_o}t \tag{3}$$

and

$$v \leftarrow \vec{a}t \tag{4}$$

5 (a) Compute p when $p_o = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $v_o = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $a = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$, and $t = 0.4$.

(b) Compute v given the parameters of the previous part.

Vectors

10 4. Draw a Cartesian coordinate system and plot the following vectors from $(3, 1)$: $\vec{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\vec{b} = \begin{bmatrix} -1 \\ -3 \end{bmatrix}$, $\vec{c} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$, $\vec{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and $\vec{e} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$

5. Perform the following vector operations:

5 (a)

$$\begin{bmatrix} \frac{1}{2} \\ 10 \end{bmatrix} \cdot 6 \tag{5}$$

5 (b)

$$\begin{bmatrix} -1 \\ -2 \end{bmatrix} - \begin{bmatrix} -3 \\ -4 \end{bmatrix} \tag{6}$$

5 (c) The dot product operation takes two vectors and returns a scalar. By itself, it isn't very useful, but it turns out to be a very handy operator when combined with other operations. Compute:

$$\begin{bmatrix} 5 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -7 \end{bmatrix} \tag{7}$$

6. Assume that your opponent is at $\vec{t} = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$ from you.

5 (a) At what point (x, y) is your opponent if you are at $(-6, -3)$?

5 (b) Compute the Euclidean distance to the opponent (show your work). Recall that the distance formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{8}$$

5 (c) Using $\vec{t}$, compute

$$|\vec{t}| = \sqrt{\vec{t} \cdot \vec{t}} \tag{9}$$

10 (d) Re-write the above equation using `numpy`. The `numpy` function for dot product is `np.dot`.

Normalization

5 7. Assume that you are at location $(-3, 2)$. You now move using the vector $\vec{t} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, then using $\vec{u} = \begin{bmatrix} -1 \\ -5 \end{bmatrix}$. What is your destination point?

5 8. Plot these vectors on a graph.

5 9. Assume that you are once again at location $(-3, 2)$. You now move using the vector

$\vec{r} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$. What is your destination? Plot this vector on the graph above.

10. Assume that you are at $(-3, -5)$ and that your opponent is at $(5, 6)$.

5 (a) What is the vector to get from you to the opponent? (Hint: the arrow head is the first coordinate).

5 (b) What is the vector to get from your opponent to you?

5 (c) Given the vector $\vec{v} = \begin{bmatrix} 8 \\ 11 \end{bmatrix}$, what is $|\vec{v}|$?

5 (d) Using $\vec{v}$, compute the normalized vector:

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|}$$

5 (e) What is the length of $\hat{v}$, that is $|\hat{v}|$?

Question	Points	Score
1	20	
2	20	
3	5	
4	10	
5	15	
6	25	
7	5	
8	5	
9	5	
10	25	
Total:	135	