Computer Programming Using Kivy 1.8 for Python 3 - **Canvas 9** page 1of2 **- Floating Text for Player Name Tag**

GOAL: Add text to the world map and translate its location to the screen (showing player name for each character).



(you name the characters!)

|  |
| --- |
| You can skip this box if you have done Canvas 7 Pictures & SpritesTo make the program work on any computer (if has Kivy installed), first make a new folder so that your program and all of your program’s pictures are in the same place (this folder can be given to anyone):* Start, Computer, your home drive (such as T:\*username*), Click “**New Folder**” (if can’t find that, [if in Windows 7 push Alt then let go] click File, New Folder) name it **bin**

Now copy all of the files for the existing sample program to the folder:* Go to Start, Computer, R:\Classes\ComputerProgramming\Examples
* Copy the graphics for the program to your home drive if you have not done this before:

“**area1.png**” “**Car, Rear - lincoln\_tc\_fournel (towncar, cholasimmons at turbosquid) - MrG's Render1.png**”“**Rock 1 by BesideTheVoid - edited from IMG\_3319 by victorblagovici on morguefile.png**”“**Sprite\_Explosion\_of\_sorts\_by\_leileilol (BesideTheVoid plain opacity version) frame0012.png**”all to your **home drive** **bin** folder (such as T:\*username*\bin. Bin stands for binary, and that is where programmers put all files that will be given to the person who will use your program. If you don’t know how, follow these steps:* + Single-click on picture, then on left bar click Copy to Folder, then choose your home drive (such as T:\*username*\bin)

(if you have trouble with that, you can drag the file to your bin folder on the left instead which will appear after you click the arrow to the left of your home drive to expand it)* + Repeat for each of these image files
 |

Use kivyBlit as a template again and save as another new program (make sure you only type the highlighted parts—the rest of the code should already be in the program):

* Open Geany, File, Open, R:\Classes\ComputerProgramming\Examples\kivyBlit.py (or your canvas Main Event Loop or Pictures&Sprites)
* File, Save As, your home drive (such asT:\*username*\bin\) then name the file canvas9text.py
* In order to use Labels, we must import the Label class at the beginning of our program:



How the “from” statement works: The line above will import the Label class from the kivy\uix\label.py file in your copy of Kivy.

* In order to add labels that are positioned according to the world instead of the screen, such as to show the name of the character, you’ll have to create a way of keeping track of world positions for labels. You can accomplish this by adding a new class. Also, each label must have a unique ID, so you can create a labelIndex variable so each has its own number (later in your program, you can use this number then add one to it each time you create a label). Since label only has a position not a Rectangle, you’ll need a new method to calculate its screen position from its world position (calculateScreenPosUsingWorldPos). The global variable captionIndex and the class, and global function must not be indented and not be in another class. For example, you can put them before class Entity:



(continue to next page)

Computer Programming Using Kivy 1.8 for Python 3 - **Canvas 9** page 2of2 **- Floating Text for Player Name Tag**

(continued from previous page)

* In the Entity class, you will need both a display name for the character, and a label where its name and possibly its status is shown:
	+ Make sure you finished the previous section and created the global variableto keeptrack of how many labels are in the label list. To change global version, make sure you type global captionIndex before using it (see below)

* + In the Entity class, add “displayName” & “caption” members, so each character can have a caption that says its name.

 

* + Then, also indented under your existing Entity \_\_init\_\_ method, use then increment the global captionIndex:





* Each time a new Entity is created (each time “= Entity()” is used, that calls the constructor, and every time our program creates a character the resulting object is lastCreatedEntity [other uses of = Entity() are for non-character entities such as the world]), add a caption to the entity, such as player character and enemy, to show its name to the player (add only the highlighted line, after Entity() constructor call):

(DO NOT type lastCreatedEntity unless you want more players—you must find where it is already typed)

 #or instead, type any name for yourself instead of Player 1

* Repeat this for computer. You can make the next one’s caption Computer or whatever you want instead of Player 1.

(add the same code above under the other Entity() constructor call, the one for the enemy)

* Now find the CanvasForm class, and under that class’ \_\_init\_\_ method, make it change the caption to the player’s name (this must be done at the end of the init method, because it cannot add the labels until the entities exist):



* Last, find the CanvasForm class’ update method and make the caption change to the position of the player, and convert that to a screen position (do this using these two lines inside of the “for” loop so that each entity’s label position is changed to the position of the entity):



See picture on previous page for what it should look like:

if it worked it will show the name beside each character—whatever you named them!