Designing Worlds

Computer Science - Week 7  
Sep 27, 2022 - Version 0.0.2

Please make sure that all members of the group place their UD **email** AND **name** below.

Choose roles following the [instructions here](https://blockpy.cis.udel.edu/assignments/reading/bakery_appendix_pogil).

You should work in groups of 3. If you cannot find 3 group members, then work in groups of 2.

| **Role** | **Name** | **Email** |
| --- | --- | --- |
| **Manager** | Zach Phillips | phillizr@udel.edu |
| **Speaker** |  |  |
| **Recorder** |  |  |

For today’s lab, you are going to be making a small game with Designer, by following a tutorial. Then, you will work together in groups to answer some critical thinking questions about Designer.

# 1) Try the Tutorial

To get started, open up a new file in Thonny and name it teleporting\_box.py  
Make sure to save the file in a folder that you can find later. In general, you need to start organizing your files and making sure you do not lose track of them!

Then, start following each step of this tutorial, copying code into your Thonny file and running after each step.

<https://designer-edu.github.io/designer/examples/world.html>

Everyone should help each other freely in following the tutorial. If you are uncertain about a line of code that you are adding, then you are encouraged to ask each other for help.

Once everyone in your group has finished the tutorial (up until “Wrapping Up”), or if 15 minutes have passed, then proceed onto the next step.

If you finish sooner than your group mates, then try some of the extra changes listed in the “Wrapping Up” section. However, if you do not reach those, you are not required to complete them.

# 2) Critical Thinking

For each of the following questions, work together and think critical to provide an answer.

Make sure you are writing complete thoughts in coherent sentences.

1. Why is a World dataclass necessary for making more sophisticated games?

| You should use the World dataclass because it allows you to compartmentalize the different game objects into a single object. It is also the view of the game, so what is in it determines what you can see. |
| --- |

2. On line 40, the expression world.box.x is used.

1.a. What type of value is world.box.x?

| int |
| --- |

1.b. What type of value is world.box?

| DesignerObject |
| --- |

1.c. What type of value is world?

| World |
| --- |

1.d. Why were there two dots in this expression?

| The first dot finds the object in the World data type, box. The second dot finds the object in the box object, the random integer between 0 and the width of the window. |
| --- |

3. On line 32, we only increase the angle by one. How much will the angle increase every *second*? Note that we are asking about *seconds*, and not how much per *update*.

| According to the documentation, the update happens 30 times per second on average. So the angle will increase 30 degrees each second. |
| --- |

4. Back in Chapter 2B1, we established two “scope rules”.

4.a. What are the two “scope rules” we established?

| Variables inside a local scope should not be used outside that scope. |
| --- |
| Variables outside a local scope should not be used inside that scope. |

4.b. Why do global variables like LENGTH\_OF\_GAME and WIN\_THRESHOLD not

violate the scope rules we established in Chapter 2?

| The two global variables are referred to as global constants when they are used this way. They are not changed in the program. |
| --- |

5. On line 37, the expression random.randint(0, 9) == 0 causes the box to teleport randomly one in ten times every step. What expression would we write instead to teleport the box *every 2 seconds*? Hint: use modulo!

| if world.timer % (2 \* 30) == 0 |
| --- |

6. Why do we never *call* the create\_the\_world function? What are we doing on line 75, instead of calling create\_the\_world?

| Instead of calling the function, we bind it to the “starting” event. So it only runs once, even though the program is constantly running. |
| --- |

7. In this game, we defined a custom event using a predicate function.

7.a. Which function that was defined in the game was a predicate function?

| The function, “the\_timer\_runs\_out” is the predicate function. |
| --- |

7.b. Why does the predicate function not require an IF statement?

| The function will automatically return False each update until the timer is greater than or equal to the LENGTH\_OF\_GAME global constant. When it finally returns True, it will automatically call the flash\_game\_over function, then the pause command. |
| --- |

# 3) Reflect and Review

Discuss among yourselves: what did you learn from this activity? What was surprising or interesting? If you didn’t learn anything, what do you think we were trying to teach you? How could this activity be improved?

| I learned about predicate functions that can sit dormant until a certain criteria is met, and then it will activate itself and other functions as needed. |
| --- |
| I learned more about scopes and global variables versus global constants. Also that the naming conventions for global contents are usually all caps. |
|  |

# 4) Extra Changes

If anyone was able to finish any of the extra changes suggested in the final section of the tutorial, then please share your code below in one of the boxes.

|  |
| --- |
|  |
|  |
|  |

# Final Submission

When your group is happy with your answers for all the questions, download this file as a Word Document (docx) and upload the file to the appropriate assignment on Canvas.

Only one member of your group needs to submit.