Python

Animal Expedition – Lesson 5-6

Designing the biome setting functionality

Student Objectives

Define functionalities for grid changing (biomes)

Concepts Covered

1. If statements
2. Functions
3. Global Variables
4. Game view/attribute setting in runtime

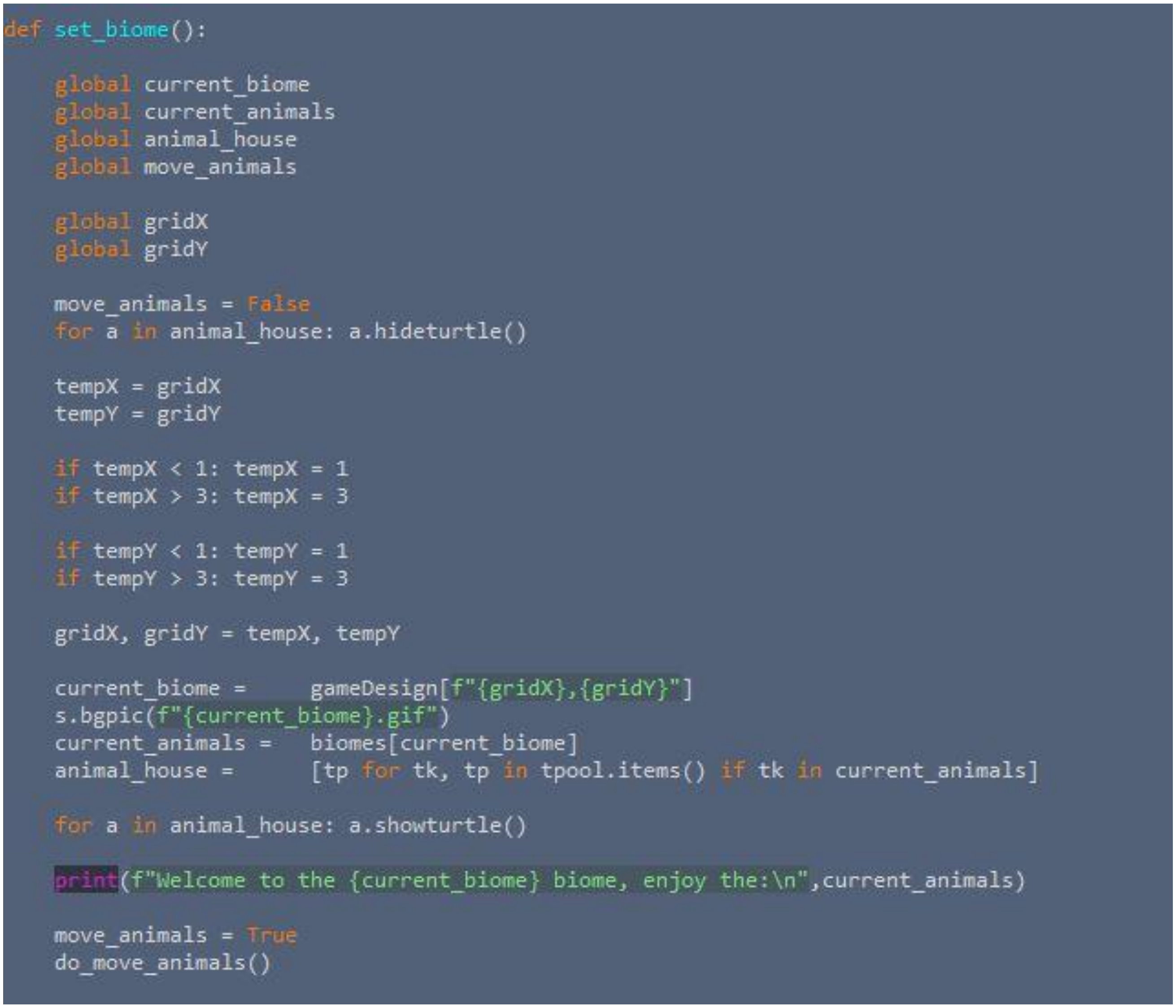
Today’ s Activity

Today, we will create the functionalities needed for the player to be able to change biomes. While the final piece (do_move_animals()) will be completed in the next class, we will see the player react and the backgrounds change.

Classroom Discussion

Create the functionality that controls the biome changing. This function will be the heart of the game, which will update the background images, which animals are showing/moving and handle calling the movement code for those animals.

Class Activity- Guided Work



```
def set_biome():

    global current_biome
    global current_animals
    global animal_house
    global move_animals

    global gridX
    global gridY

    move_animals = False
    for a in animal_house: a.hideturtle()

    tempX = gridX
    tempY = gridY

    if tempX < 1: tempX = 1
    if tempX > 3: tempX = 3

    if tempY < 1: tempY = 1
    if tempY > 3: tempY = 3

    gridX, gridY = tempX, tempY

    current_biome =     gameDesign[f"{gridX},{gridY}"]
    s.bgpic(f"{current_biome}.gif")
    current_animals =   biomes[current_biome]
    animal_house =      [tp for tk, tp in tpool.items() if tk in current_animals]

    for a in animal_house: a.showturtle()

    print(f"Welcome to the {current_biome} biome, enjoy the:\n",current_animals)

    move_animals = True
    do_move_animals()
```

Lesson Overview / Summary

1. Using global calls for the 6 major global variables that handle controlling the game, stop allowing the animals to move (set move_animals = False) and hide all the current animals in the animal_house (the variable used to store the current shown/moving turtles).
2. Using tempX/tempY, store the current grid locations and then use if statements to check and ensure that the grid values to not exceed the allowed min/max levels (this is essential to avoid errors that may result from border conditions running twice → the fix will sometimes cause the player to ‘ move oddly’ across the screen and is expected because it keeps the game from breaking). When set to proper values, reset the global variables to those values.
3. Extract the current biome name (using the gridX/gridY values to create the key name), set the background image to the proper .gif background file (image). Then extract the current animals (list from the resource dictionaries created in lesson 1) and extract ‘ copies’ (actually references that use pointers) of the current animal turtle objects and place them in the animal_house (now redefined).
4. Finally, show the turtles (using a loop), notify the player of the new biome and list of animals to see, allow animal movement and then call the do_move_animals() function.
5. ***NOTE: Comment out the >> s.bgpic(f” {current_biome}.gif”) << line of code until all background images are created. For example purpose, use s.bgcolor() and set random colors for background to demo the effects for now.
6. ***NOTE: Comment out the line of code that reads >> do_move_animals() << until after finishing the code in later lessons. You can however uncomment the lines of code >> set_biome() << in the checkBorders() function because the next step is completed.

Independent Play/ Game Customization

Allow your students to enjoy their new game! Give them the last 10 minutes of class to play their game, or explore what they can do to customize their world. Celebrate creativity!

Building Relationships

Share what your students can do! Take screenshots of your student’ s work, or pictures of them (absolutely no faces please!) with their computer screens, and use the handle @CodeAdvantage and the handle #CodeAdvantage to let others see and be inspired by your students’ success with coding!

(Tweet or Instagram example: “ My students completed @CodeAdvantege #Lets Float Around today! We designed and coded games! I’ m so proud of them! My students rock!)

Consider the plans for your next lesson and start preparing for how the kids are going to have a great time with @CodeAdvantage creating #(next lesson name)!

We value your opinion! Please send any lesson suggestions, edits, comments, or questions to feedback@codeadvantage.org.