LESSON PLAN – Computer Programming

Title: Hobbits vs. Nazgul – Predator Continued

Essential Questions	Can we model life-like behavior with Python? How can we apply our expanded understanding of data types and storage?

Learning Objectives:	Students will be able to: ● Adapt prey movement and breeding logic to predator rules ● Create or modify functions to enact predator movement and breeding

Standards (CSDF)	
9-12.CT.4	Implement a program using a combination of student-defined and third-party functions to organize the computation.
9-12.CT.5	Modify a function or procedure in a program to perform its computation in a different way over the same inputs, while preserving the result of the overall program.
9-12.CT.7	Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.
9-12.CT.8	Develop a program that effectively uses control structures in order to create a computer program for practical intent, personal expression, or to address a societal issue.
9-12.CT.9	Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.
9-12.CT.10	Collaboratively design and develop a program or computation artifact for a specific audience and create documentation outlining implementation features to inform collaborators and users.

Teaching Materials:
Student handouts (attached)
Student computers

Procedure:

I. Checklist from previous day
    A. Confirm that previous day's tasks are complete
    B. Adjust plan to allow for time outside of class if progress is an issue
II. Division of labor
    A. Students will decide on their approach to predator functions
        1. Assign parts of the code among themselves
        2. Decide where to modify existing code and where to create new functions
    B. Focus is on finishing movement and adding breeding logic
    C. Will need to add predator death functionality
III. Student work time

Day 8 Checklist:
Adding predator breeding and death logic

You must have the following tasks accomplished before the end of class today.

Accomplished	Task
●	Make sure you have completed all of the tasks from yesterday's (Day 7) checklist Do you need to schedule time outside of class in case progress is an issue? Discuss with your team.
●	Division of labor and assignment of tasks. ● **Who works on what?** ● **Again, who should take on what responsibilities unless that/those person(s) are out for the day?** ● **What needs to happen if a team member is stuck or becomes frustrated?** *Collaboration is encouraged if the team feels it necessary to be successful.*
●	● Finish predator logic ● How should your team approach the predatory functions? 🌕 Work out who works what of this part of the code amongst yourselves. 🌕 Keep in mind the questions posed in the previous checklist item

	🌕 What parts of the code need to be **modified**, and what other parts may need more functions to work more efficiently.
●	Finish predator motion logic ● How do predators move? ● How does the number of prey (hobbits) affect their motion?
●	Begin work on predator death functionality ● Think back to how the Nazgul died in your models on paper? ● What defines when they finally die? ● How do we express this in Python?