

## LESSON PLAN – Computer Programming

### Title: Hobbits vs. Nazgul – Prey Continued

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

<b>Learning Objectives :</b>	Students will be able to create functions to: <ul style="list-style-type: none"> <li>• Check for open spaces, allowing for hobbit teleportation</li> <li>• Choose a space to move to</li> <li>• Move to a new space</li> <li>• If a hobbit is ready to breed, produce a new hobbit in an adjacent open space</li> </ul>
------------------------------	---

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:**

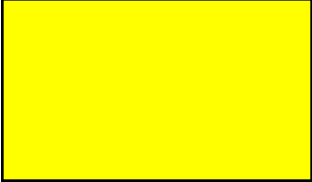
- I. Checklist from previous day
  - A. Confirm that previous day's tasks are complete
  - B. Adjust plan to allow for time outside of class if progress is an issue
- II. Division of labor
  - A. Students will decide who will work on what portion of the code
    1. Finishing movement logic
    2. Developing breeding logic
  - B. Students will likely need to collaborate on this code, as some parts will be logically complex
  - C. Students may choose to subdivide the above tasks into separate functions
- III. Student work time

## Day 4 Checklist:

### Finish prey movement, add breeding

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

Accomplished	Task
●	<p>Make sure you have completed all of the tasks from yesterday's (Day 2) checklist</p> <p>Do you need to schedule time outside of class in case progress is an issue? Discuss with your team.</p>
●	<p>Division of labor and assignment of tasks.</p> <ul style="list-style-type: none"><li>● Who works on what?</li><li>● Again, who should take on what responsibilities unless that/those person(s) are out for the day?</li><li>● What needs to happen if a team member is stuck or becomes frustrated?</li></ul> <p><i>Collaboration is encouraged if the team feels it necessary to be successful.</i></p>
●	<p>Finish movement logic</p> <ul style="list-style-type: none"><li>● Test and <b>retest</b> your code every time you make revisions to your code. Do not wait until you have made many changes</li><li>● This way it will be easier to catch any errors</li></ul>
●	<p>Develop breeding logic</p> <ul style="list-style-type: none"><li>● AGAIN... Test and <b>retest</b> your code every time you</li></ul>



make revisions to your code. Do not wait until you have made many changes

- This way it will be easier to catch any errors