



INTERACTIVE STORY PROJECT TEACHERS' NOTES



Interactive Story Project - Teachers' Notes

Project Overview

This project consists of six 30-minute lessons. It introduces basic coding and screen interface concepts as well as basic interactivity. Students will use direct commands and student-created programs as well as interactive control objects (such as buttons), graphics, text, and sound to tell a story.

Although this project is focused on describing a trip or adventure, students can use the tools and coding skills they learn for any kind of narrative-based project.

Prerequisites

None. This is an Introduction to using Lynx. These *Level 1: Tech Trainee* lessons are mandatory for students to complete in advance of their first Hackergal Hackathon.

Standards Addressed

In this project, students will meet the following learning standards:

- Describe the topic, purpose, and audience for media texts they plan to create.
- Construct meaning through the combination of several media “languages” – images, sounds, graphics, and words.
- Plan and write simple computer programs, applying fundamental programming concepts and that include clear internal documentation.
- Develop creative solutions for various types of problems.

Before You Start

1. At this point you should already have your teacher account set up on Lynx and have rostered your girls in a team.
2. Give students some time at the beginning of *Lesson 1* to do the following together with a projector:
 - Create their own account in Lynx by following the steps in the *How to Create a Lynx Account*.
 - Review the *Terry Fox* sample at lynxcoding.org to understand how to use Lynx to create an interactive, multimedia story. Point out the different components – background image, text, animation, multiple pages and page navigation, buttons, etc.

Interactive Story Project - Teachers' Notes

3. Have a conversation with your students about the importance of saving your code. It is important to tell students that Lynx does not automatically save their work, so they must remember to save their projects often! Saving is quick and easy on Lynx using the icon on the left sidebar.

Learning Targets and Checklist

Projects should demonstrate the students understanding of the following skills. Each of these should contribute, in a logical way, to telling the story.

✓	SKILL
	Use basic coding commands to move a turtle (sprite) to create simple animation
	Use provided and imported clipart to create a background, stamp images, and change turtle shapes while learning about image screen layers
	Add and format text
	Write, edit, and debug simple user-defined programs
	Use commands to create more complex animations and explore multiprocessing
	Add interactive control with buttons and programmable sprites
	Add additional multimedia features, such as sound effects and music
	Create multiple pages and an interactive navigation system

New Lynx Commands

```
setshape, setsh  
setsize  
cleargraphics, cg  
forward, fd  
back, bk  
right, rt  
left, lt  
wait  
glide  
setheading, seth  
repeat  
to, end  
stamp  
showtext
```

Interactive Story Project - Teachers' Notes

Common Errors and Debugging

Debugging is an excellent medium for developing problem-solving skills. Lynx has a number of tools, to support students as they learn to code in Lynx, including auto-complete, tool tips, error messages, and, of course, a *Help* page.

To begin, it's good to identify the type of bugs a student may encounter. There are two main types of errors that programmers make. Most errors at this level will be the first type.

1. Typing or syntax errors, for example:

- Forgetting to put a space between a command and its input, like writing `fd30` instead of `fd 30`.
- Typos, for example typing `fe 20` instead of `fd 20`.
- Using the wrong type of input or syntax punctuation, for example, writing `repeat 4{fd 1 wait 1}` instead of `repeat 4 [fd 1 wait 1]`

When an error occurs, a message appears describing the error. Often, students ignore messages, but since these messages highlight where the error is, students should be reminded to pay attention to them.

For example, if a student types `fd30`, they will get the message:

I don't know how to fd30

Lynx interprets this single word as a new command that is neither a built-in command nor a procedure name. Unlike people, Lynx can't "interpret" or "figure out" what the student means, it can only respond to what is typed.

2. Errors in logic

- Sequence of actions - Commands need to run in a specific order, depending on how the coder wants events to unfold. This idea of order is something everyone commonly deals with in life. For example, we need to put on our socks before we put on our shoes and not the inverse. This algorithmic thinking is a main computational thinking skill.

Although this seems easy, as programs become more complex, the sequence of events may become more difficult to follow. In this project, understanding the logic should be fairly easy, but it's a good idea to help students be aware of how they determine what actions they may want to have happen before others and to be able to put their thinking into words.

Interactive Story Project - Teachers' Notes

- The flow of information from different types of commands, including those that send (or output) information to another command. These types of problems may appear once students are coding more complex projects.

Final Notes

Each lesson should take about 30 minutes (although, if students have more time and want to continue, there's lots to explore!)

Students can spend a long, long time looking for the artwork they want. You may want to limit the time they take for this task and/or suggest, if there are additional images they want to add, they can search for them between sessions (at home, at the library, etc.) and take note of their location.