# Lesson 2.07: Project

## Learning Objectives

Students will be able to… \* Use knowledge of lists, Booleans, conditionals, and while loops to create a text-based adventure game.

## Materials

* [Project Spec - Text Monster](project.md) ([printable Text Monster project spec](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/project.pdf)) ([editable Text Monster spec](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/project.docx))
* [Project Spec - Ogopogo - Text Monster (Canadian version)](project_canada.md)
* [Project Spec - To-do List](alternate_project.md) ([printable To-do List project spec](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/alternate_project.pdf)) ([editable To-do List project spec](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/alternate_project.docx))
* [Text Monster Starter Code](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/text_Monster_Starter_Code.py)
* [Editable Grading Rubric](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/rubric.docx)
* Sample Solutions (access protected resources by clicking on “Additional Curriculum Materials” on the [TEALS Dashboard](http:/www.tealsk12.org/dashboard))

## Preparation

* Read through the [Associated Readings 2.7](https://tealsk12.github.io/2nd-semester-introduction-to-computer-science/readings.md#associatedreadings/2.7)
* Try creating your own variation on the Text Monster code so you are familiar with the potential challenges and bugs your students will hit.
* Review [4 Steps to Solve Any CS Problem](https://github.com/TEALS-IntroCS/2nd-semester-introduction-to-computer-science-principles/raw/master/units/4%20Steps%20to%20Solve%20Any%20CS%20Problem.pdf)
* Update the Project Spec of your selected project as needed to meet your grading requirements

### Day 1 Pacing Guide

|  |  |
| --- | --- |
| **Duration** | **Description** |
| 10 Minutes | Project Overview/Demo |
| 40 Minutes | Design |
| 5 Minutes | Debrief |

### Days 2 - 9 Pacing Guide

|  |  |
| --- | --- |
| **Duration** | **Description** |
| 10 Minutes | Review |
| 40 Minutes | Project Work |
| 5 Minutes | Debrief |

## Instructor’s Notes

### 1. 4 Steps to Solve Any CS Problem

* Review [4 Steps to Solve Any CS Problem](https://github.com/TEALS-IntroCS/2nd-semester-introduction-to-computer-science-principles/raw/master/units/4%20Steps%20to%20Solve%20Any%20CS%20Problem.pdf)

### 2. Project Overview/Demo

* Distribute the project spec to all students and walk them through the goals and requirements of the project.
* Show a demo of a completed project.
* Go over specific design considerations from the project spec:
* Introduce the concept of global variables and how they will be useful here.
* Identify the importance of the “User Pocket” and how to use a list along with append and remove for this information.

### 3. Design

* Have students stay at their desks and write down what lists they’ll need.
* They should break up the project into parts: parsing user input, keeping track of players position, returning what is at the player’s position .

### 4. Debrief/Review

* During discussion and warp up at the end of class, get a feeling for where students are in the project.
* During the review the next morning cover the topics/areas that students are struggling on and present tips, suggestions, and goals for that day.

## Accommodation/Differentiation

* Make sure to do status checks with all students throughout the project.
* Identify students that are struggling on the project after the first few days and provide additional scaffolding & support as needed.
* For any students that are advancing rapidly through the project, give them extension ideas such as adding a new feature or floor to the game.
* Advanced students can also be paired as tutors/helpers with struggling students.

## Grading

### Objective Scoring Breakdown

[Editable Grading Rubric](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/rubric.docx)

Student correctly identifies data types (Lesson 2.01) - Assessed in Unit 3

|  |  |  |  |
| --- | --- | --- | --- |
| Points | Percentage | Objective | Lesson |
| 3 | 12% | Student correctly uses conditionals to maintain flow of control | 2.02, 2.03 |
| 9 | 36% | Student correctly uses lists | 2.04 2.05 |
| 3 | 12% | Student can correctly use the while loop | 2.06 |
| 5 | 20% | Student can decompose a problem to create a program from a brief |  |
| 5 | 20% | Student uses naming/ syntax conventions and comments to increase readability |  |
| **25** | **Total Points** |  |  |

### Scoring Consideration

You may need to adjust the points in order to fit your class. Treat the percentages as a guide to determine how to weight the objectives being assessed.

## Forum discussion

[Lesson 2.07: Text Game (TEALS Discourse Account Required)](https://forums.tealsk12.org/c/2nd-semester-unit-2/lesson-2-07-text-game)