Lesson 3.05: Project 3 ## Learning Objectives Students will be able to... * Use project planning skills to complete a longer-term project * Create functions to organize a project * Apply skills learned in units 1-3 to create a functioning program ## Materials/Preparation * [Project Spec - Oregon Trail] ([printable project Spec]) ([editable project spec]) * [Alternate Project Spec - Daily Planner] ([printable alternate project Spec]) ([editable alternate project spec]) * [Oregon Trail Starter Code]((https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/05_lesson/Unit_3_Project_Oregon_Trail_Starter_Code.py)) * Solution (access protected resources by clicking on "Additional Curriculum Materials" on the [TEALS Dashboard](https://www.tealsk12.org/dashboard/)) * Update the Project Spec as needed to meet your grading requirements * Try creating your own variation on the Oregon Trail code so you are familiar with the potential challenges and bugs your students will hit. * Review [4 Steps to Solve Any CS Problem] * [Editable Grading Rubric](https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/2_unit/07_lesson/rubric.docx) ### Day 1 Pacing | **Duration** | **Description** | | ---------- | ----------- | | 10 Minutes | Project Overview | | 40 Minutes | Planning | | 5 Minutes | Debrief | ### Days 2-9 Pacing |**Duration** | **Description** | |--|--| | 5 Minutes | Review Day Plan | | 45 Minutes | Project Work| | 5 Minutes | Debrief | ## Instructor's Notes ### 1. 4 Steps to Solve Any CS Problem * Remind students of the [4 Steps to Solve Any CS Problem] ### 2. Project Overview * Demo the Oregon Trail finished project. * Give out the project spec and go over game rules. ### 3. Planning 1. Have students draw out the game play 2. Students should plan to create functions for each user interaction by figuring out where the repeated code will be. 3. Students should list out which variables they will need. 4. Have students plan out their next 7 days. Suggested timeline/checkpoints below: * Set up user inputs with dummy functions, make sure game loop works * Create variables necessary to run the game, start implementing basic functions * Focus on the random functions * Figure out how to move the days * Finish day updating * Connect functions together * Wrap up and game over check is correct ## Accommodation/Differentiation * Advanced students can add in random events like cholera or snake bites. * Students can also have a list of travelers instead of just 1, where each traveler is affected differently by each action. * The planning phase of this project will be essential, * especially for students who you think may struggle with this project. * Provide more guidance and scaffolding to those students that need it. ## 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) | Points | Percentage| Objective | Lesson | | :---: | :---: | --- | --- | | 3| 10% | Student correctly identifies data types|2.01| | 3 | 10% | Student correctly uses lists |2.04, 2.05| | 3 | 10% | Student correctly uses built in functions | 3.01| | 12 | 38% |Student can program using user-defined functions |3.02, 3.03, 3.04| | 5| 16% |Student can decompose a problem to create a program from a brief|| | 5| 16%|Student uses naming/ syntax conventions and comments to increase readability|| | 31 | | **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 3.05: Oregon Trail (TEALS Discourse Account Required)](https://forums.tealsk12.org/c/2nd-semester-unit-3-functions/lesson-3-05-oregon-trail) [Project Spec - Oregon Trail]:project.md.html [Alternate Project Spec - Daily Planner]:alternate_project.md.html [Oregon Trail - Example Code]:oregon_trail.py [TEALS Dashboard]:http:/www.tealsk12.org/dashboard [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 [printable project Spec]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/05_lesson/project.pdf [editable project spec]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/05_lesson/project.docx [printable alternate project Spec]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/05_lesson/alternate_project.pdf [editable alternate project spec]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/05_lesson/alternate_project.docx