Criteria	Grading Scale			
Conception & Design Students are encouraged to pursue projects that inspire them personally, engage their attention, spark creative thinking, encourage innovative solutions, and challenge them in ways that demand meaningful growth in both coding and physical computing expertise.	**18** Distinguished: Project chosen meets or exceeds all stated objectives for conception & design.	**16** Proficient: Project chosen meets most objectives for conception & design but did not sufficiently challenge the student or inspire creativity and innovation.	**14** Competent: Project chosen was equal to current ability level and offered little challenge, opportunity for growth and skill development, or demonstration of creative or innovative thinking.	**12** Unsatisfactory: Student selected a project beneath their skill level such that it offered no challenge, required no acquisition of new knowledge or development of new skills.
Code Execution Students are encouraged to write concise, logically sound, elegant programs that function as intended, are free of bugs, and include extra features that exceed requirements and demonstrate creative or innovative thinking, or acquisition of more advanced skills.	**18** Distinguished: Program is functional and refined, with extra features that exceed the requirements.	**16** Proficient: Program works in the way the student intended and within the parameters set forth by the project designers.	**14** Competent Program mostly works, and only has minor flaws that could be eliminated with careful debugging.	**12** Unsatisfactory: Program does not work, or has major flaws that prevent its intended use.
Code Practice Students are encouraged to write programs that are clean, organized, readable, free from redundancies, and follows the conventions appropriate to the language.	**18** Distinguished: Program is well organized, easy to read, free of unnecessary or redundant programming statements, makes good use of white space and comments, and variables have helpful names.	**16** Proficient: Program is well organized, easy to read, and understand.	**14** Competent Program can be read in and is in a logical order.	**12** Unsatisfactory: Program is difficult to read. Code contains lines that do not work or are out of order.
Personal Reflection & Peer Support Students are encouraged to reflect on all aspects of their project plan from conception and development through execution and presentation. And as part of a community of learners, students are asked to support and assist each other, especially with regard to helping others debug their code.	**18** Distinguished: Student can describe how their code works, how they wrote it, and can justify choices regarding structure and reasoning. They are proactive in helping others develop and debug their code in ways that support mutual growth and respect.	**16** Proficient: Student can describe how their code works and can make changes that have desired effects; and/or they willingly assist others with questions or debugging when asked.	**14** Competent Student can mostly describe how their code works; and/or they prefer to work alone and avoid assisting others.	**12** Unsatisfactory: Students cannot describe how their code works; and/or they refuse to assist others.
Habits of Mind Students are encouraged to develop healthy habits of mind such as identifying, articulating, and adhering to program goals, seeking out new ideas and alternate solutions, making sense of problems and persevering in solving them, soliciting feedback and insight from experts to generate creative or innovative ideas, and always giving credit when deserved by commenting their code.	**18** Distinguished: Student embraces the goal of the program and chooses to try out new ideas and multiple solutions, even when they are challenging. They seek out creative or innovative ideas from others and always give credit to others by commenting their code when credit is due.	**16** Proficient: Student understands the goal of the program, has their own ideas, rarely goes off task, and attempts to solve problems first before asking for help; and/or student does not credit colleagues who assisted them in significant ways.	**14** Competent Student is aware of the goal of the program, returns to the task when asked, has some ideas when prompted, and asks for help when stuck. But the student struggles with self-management and would not complete the program without supervision.	**12** Unsatisfactory: Student is not aware of the goal of the program, is frequently off task, does not offer their own ideas, and gives up when it gets difficult.
Peer Feedback Peers tested each other's games and rated them in the following categories: [1] I understand what the game is, how to play, and what the goals are. [2] I can easily read and understand how the code for this game works. [3] The code of this game makes good use of the programming constructs we learned. [4] This game is enjoyable and fun to play. They also finished the statements "I like…", "I wish…", and "What if…" to provide constructive and useful feedback to the game developer.	**5** Distinguished: When field tested by classmates, users rated the program highly in all four categories.	**4** Proficient: When field tested by classmates, users rated the program highly in three of four categories.	**3** Competent When field tested by classmates, users rated the program highly in two of four categories.	**2** Unsatisfactory: When field tested by classmates, users rated the program highly in one of four categories.
Response to Feedback Game developers responded to peer feedback by responding to each reviewer with the following information: [1] What piece of feedback was most helpful to you? Why? [2] What piece of feedback surprised you the most? Why? [3] Based on the feedback you received, what changes will you consider making in your game?	**5** Distinguished: Response to feedback was positive, detailed, and thoughtful; showing that the game developer values the input of the player and will modify the game code in response.	**4** Proficient: Response to feedback was positive and thoughtful; showing that the game developer values the input of the player, but did not provide detailed ways the game code would be modified in response.	**3** Competent Response to feedback was positive but vague; showing that the game developer only minimally considered the input of the player.	**2** Unsatisfactory: Feedback to the player was negative and/or insufficient to show to the player that the game developer considered the input and will modify the game in response.

Total pts: 100