|  |  |  |
| --- | --- | --- |
| **FINAL GAME PROJECT** | | |
| **Successes**  ***Strong areas of your work*** | **Project Requirements** | **Suggestions**  ***How you can strengthen your work*** |
|  | **Creativity and Design**  A player can easily understand how to play the game, or there is an instructions page with clear instructions.  Shape and color commands are intentional, effective, consistent, and contribute to the project as a whole  Big aesthetic/design choices are explained/justified in your description  High-level idea for project is clear and cohesive  Your game has a start screen, a game screen, and a game over screen.  The game scales in terms of its challenge |  |
|  | **Skill Application: Logic**  If-statements are concise (compound boolean expressions are used as appropriate)  Else and else-if are used appropriately, and all if-statements can possibly be executed without being overridden (i.e. don’t have ineffective code)  Your game is able to reset, and functions exactly the same upon reset |  |
|  | **Skill Application: Randomness**  Your randomness is “controlled” randomness, which means you have restricted what is randomly possible  The restrictions and decisions you make with respect to your controlled randomness enhance your project |  |
|  | **Skill Application: Variables**  Variables are used when values are changing, need to be calculated, or are related to each other  Variable names clearly reflect their purpose and enhance the readability of your code |  |
|  | **Skill Application: Functional Programming**  Your code is organized into functions: large sections of code are broken up into smaller functions  Function names clearly reflect their purpose and enhance the readability of your code  Provided collision-checking functions are used appropriately  Function parameters are used when necessary |  |
|  | **Skill Application: Loops & Arrays**  Repetitive patterns are created using loops  Arrays are used to store any related series of data  Loops are used to cycle through all values in an array |  |
|  | **Communication**  Your name, date, description, and (if applicable) sources are included at the top of your project  You submit a zipped folder as your submission, with the folder having your name in its name  Comments are used to delineate between different sections of your code  Your code is thoughtfully organized: similar pieces of code are near each other and in a sensible order  If applicable: Sections of your code that are more challenging to quickly understand at a glance are clearly explained with comments  If applicable: Sections of your code that are borrowed from an external source have been labeled and explained with comments, and the source link is copied again near that code |  |
|  | **Resilient Learning**  You included 2 peers edits at the bottom of your code and addressed what you did in response to their feedback (i.e. did you agree and decide to follow the advice, or disagree and decide not to, or make an attempt but run out of time to implement the suggestions)  Your description includes some amount of process documentation (e.g. What would you do with more time? What part of the project was especially challenging? What part of the project was especially fun? What did you have to figure out on your own? What helpful input did you get from others? |  |