

Name _____

Date _____ Per _____

	Exemplary (4)	Good (3)	Marginal (2)	Incomplete (1)
Presentation Skills	Presentation is clear (audibly, visually), well paced, engaging, and organized. Font size (code portions, slides) is 20 pt or greater with attention to brevity on slides (presenter should say more, slides should say less).	Presenter makes an effort to engage the viewer (but is not entirely successful or loses the audience after some time, perhaps due to lack of clarity or irrelevant/extraneous content). Slides are of quality described at left.	Presentation does not seem organized well, so viewer gets lost or disengages more than once. Presentation has small points of ambiguity or overly technical language. Likely that viewer is engaged only out of politeness, but may appear bored or lost. Entire presentation is audible, but slides feel jumbled without attention to conciseness.	Presentation seems unplanned and jumbled. Several opportunities to explain clearly are missed; presentation seems disorderly or unfocused. Speaker may be hard to hear or use ambiguous or overly technical language. Likely that viewer feels disengaged or embarrassed for presenter.
Educational Goal and audience	Educational goal and audience are defined clearly, including what educational context game would be best suited for .	Goal and audience are clear, but either goal or audience is too broad for what game shows. Context for how game would be used is ambiguous or too broad.	Goal and audience are clear but too broad for what game shows. Context for use of game is not clear.	Goal and audience are ambiguous or audience does not match goal. Test subjects do not match desired audience for designed game.
How to Play	Process of game play is clear for user (rules, goal, feedback) as well as designers (logic, flow of data). User Interface includes layout and number of components to make game play intuitive.	Process of game play for user and designer has some ambiguity, but is generally clear and is complete (includes rules, goal, feedback for user, logic, flow of data for designer). UI has layout and components that make game intuitive, though there may be one extraneous region, step, or component or one issue with navigation.	All aspects of game play (described at left) presented, though some are only implied. UI has layout and components that are relatively intuitive, though there may be two extraneous regions, steps, or components, or issues with navigation.	Presenter skips aspects (described at left) or game play for user and designer. UI layout and components not assigned in orderly, readable, logical way based on conventional rules of interface design. Navigation problematic.
Multiple levels (or game variations) <i>It is not expected that more than the most basic level is coded/running.</i>	Describes varied levels or game play (even if they do not function) that make sense based on feedback, challenge, educational goal, or context meant to be used in teaching. Design of existing game is clearly compatible with later integration of levels or variations.	Levels make sense, and are based on feedback, challenge, ed. goal, or teaching context, but it is unclear how they would be integrated with the current version of the game.	Levels make sense, but are not clearly based on feedback, challenge, ed. goal, or teaching context and integration based on current game design doesn't make sense / is unclear.	Levels don't make sense (but are included in presentation).
Arrays/ Data Org and Management	One case of array or file use is explained, using appropriate language from class to indicate specifically how it is used.	One case of array or file use is explained, using appropriate language from class to indicate generally how it is used.	One case of array or file use is explained, to indicate how it is used, but some language choice is poor.	One case of array or file use is provided, but how it is said to be used is lacking context related to attributes of that structure.

Data structure choice – wise for specific representations	One example of input is described/demonstrated that changes how the game functions/output. One example of a data set exists in a compact form (e.g. file, object, array) when a group of related data exists. For code shown, variable names are meaningful, best data type is used for them, there are no magic numbers (final constants and variables are used).	Input changes how game functions/output. One poor decision was made in terms of storing data compactly and use of magic numbers. Most variable names are meaningful with best data types.	Input changes game functions/output. Not consistently thoughtful about storing data compactly, not using magic numbers, picking the best data type, or good naming of variables.	Input changes function or output but how this works is ambiguous. No evidence of storing sets in appropriate compact forms or using files. Variable names poor.
Math/Data Type Conversion/Data Analysis	Math or String methods, operators, or casting are presented (for one case) clearly using appropriate language from class. Parameters, return types, and/or limitations of process (e.g. operation) in Java context shows thorough understanding.	Math or String methods, operators, or casting are presented (for one case) clearly using appropriate language from class.	Math or String methods, operators, or casting are presented with some poor use of language.	Math or String methods, operators, or casting are presented but not explained or inaccurately explained.
Control Structures / Decision Making	One block is easy to follow in code (indentation, large blocks broken up into methods and/or classes, uber-complex one-liners broken up) OR in diagram. Sequence of steps, selection (conditional choice), and iteration (loops or recursion) are made clear with appropriate vocabulary from class.	Block is shown in detail in code (see details at left) or a diagram, but language choice is sometimes not appropriate/specific in description of sequence, selection, and iteration.	Block is shown in code or diagram with one detail (at left) missing and language choice is ambiguous in description of sequence, selection, <u>or</u> iteration.	A block of code/diagram is shown with ambiguous description of sequence, selection, <u>and</u> iteration.
Events and GUI Interface	A specific event related to GUI is explained with clear reference to the event, listener, or event handler method, using appropriate language from class.	A specific event related to GUI is explained with clear reference to the event, listener, or event handler method, but some language choice is poor.	A general event is explained with reference to the event, listener, or event handler method, with ambiguity in context and language.	Use of the event, listener, or event handler method is mentioned, but it is not clear what general or specific event is related to it, or how that event is associated with larger game play.
Game project was iterative and used feedback from ed. advisor, peers, and UI testing	Mentions at least two specific pieces of feedback <i>and</i> each of their impacts on game design, across the duration of the project. It is clear that feedback was very helpful and designer sought it actively at different points of the project.	Mentions one specific piece of feedback and its impact on the game design. OR... missing some part of the “exemplary” requirement	Mentions one specific piece of feedback but not its impact on game design or some variation of this.	Feedback is present, without impact to design of the game... OR... Obvious flaws are present that should have been removed following feedback iterations.

Note, if one of the categories is not dealt with, it is possible to earn 0 points for that category.