[18fa]:

* (jennche): instead of allowing undo (which is easy) for collections, can maybe require both undo/redo functionality (which makes it a bit harder) for it to count
* (jennche): reconsider giving points for undo in games where it really doesn't make sense (e.g. minesweeper).
* (emsu): add note to subtyping/inhertance that says that they \_have\_ to make a \_new\_ interface/abstract class with a method that the implementing classes implement differently (they can't just provide different implementations for a pre-existing method)
* (jennche): change collections -> data structures (to include map theoretically)
* (jennche): Change GameTest (submission test for compilation) to something like SubmissionCompilationTest. People keep on getting confused and thinking its their file.
* (jennche): better instructions (literally give them screenshot by screenshot) on how to submit
* (jennche): in the collections rubric, we don't have anything for like whether or not the collections usage actually works/displays in the game properly?
* (jennche): think about banning SortedMap/NavigableMap as a way to sort for high scores? It makes it kinda easy/dumb idk
* (jennche): force Emma and Jane and (probably whoever is in DMD) to rewrite the collisions/3d graphics part of the rubric
* (emsu): for testable component, make a note that their tests \_have\_ to be testing the game model, not something else
* (Placeholder for todos added from Slack)

-- 17fa --

Emma said he made changes so make sure to pull first.

[PP]: add more examples of things that count for advanced concepts

* [MC]: we intentionally got rid of those so people didn’t just use libraries to add functionality

[PP]: mention JOptionPane

[JC]: the codio zip file needs to zip subfolders within each subfolder (e.g. if there is an img folder inside of files)

[GS]: readjust the rubric points if the student only read/writes for IO in hw9

Notes from staff meeting 18fa:

* ~~Everything that says codio should say Eclipse~~
* ~~Links that have to be changed - change to 18fa from 18sp~~
* ~~If you took 110 and made this game, can’t make the same game for this class~~
* ~~Updated the deadlines from April 2018 to November/December 2018~~

-- various --

**ACTION ITEMS:**

* Add instructions to homework HTML on how to zip the assignment properly, also pass those instructions on to Piazza committee
* From everything below probably have to discuss

**HW9 -**

* JC: Common question: “Can I reuse code from HW08?” - add to instructions or FAQ
  + ~~DF: Should say in instructions that they cannot reuse code if they are counting that concept~~
* JC: I think difficulty is too low now -- tic tac toe can theoretically work for this -- and we are getting more snakes/connect four than in the past for some reason. Is there a way we can reward people who work on harder games more?
* ~~JC: More rubric detail on I/O highscore~~
  + ~~Does it need to be sorted? Do they have to show the name along with the score? Do they need only the top score or multiple?~~
* JC: For cheat detection, pull multiple github sources to check against (at least for the most popular few games)

All of this is for homework 09 instructions: strikethrough means done

-BH (1/15/17): in general, we need a way to create less ambiguity

1. ~~Provide a master list of common things that do~~ **~~not~~** ~~count as concepts~~
   1. Will involve going through piazza/old games from previous semesters
   2. Devote a committee to this?
2. Limit the assignment to a specific list of games + requirements students are allowed to choose from
   1. Will require something similar- going through previous semesters and finding games that worked well
   2. For each game, list 4-5 specific concepts that students would have to choose among to implement (ex. Chess = checkmate, 2d array for board, pieces for inheritance, smart ai, highscores, they can pick 4)

-----------------------

* ~~Clarify that changing concepts is okay (but not recommended) as long as they make a Piazza post to get feedback~~\*

^done but along these lines, do we make a Google Doc where TAs can write down if

they approved something (to avoid “a TA told me this was fine during office hours”)?

And if this is done update instructions accordingly

* Testing:
* Clarify that code should be designed so it is unit-testable (define what this is as well) ← this still has to be done