Drop APCS1 pd 4
Jenny Han, Levi Olevsky, Arif Roktim 2016-01-06

Final Project Proposal

Remake of Bejeweled

We plan to make Bejeweled in the terminal.
Features:

- Class Gem
    - Each gem will be an instance of class Gem
    - Different gems have different colors -- colored output will be achieved by using ANSI escape sequences
    - The gems will be stored in a 2D array of type Gem, which will represent the board
- Gems can be swapped with adjacent gems to create horizontal or vertical chains of 3 or more gems of same color
    - Gems may only be swapped if the result creates a chain of at least 3 gems of same color
    - Earn points for each gem destroyed
    - After gems are destroyed, the gems on top of the destroyed gems will "fall" to replace the previous gems
    - New gems will be added from the top of the board; the color will be random and sometimes, special gems will be added randomly
    - Uses 2d array manipulation -- swapping elements and shifting columns
    - Uses ArrayList of type Integer[] to keep track of positions of all items in a chain
- Powers/Special gems:
    - Abstract class SuperGem extends Gem
            - Created when more a chain of more than 3 gems is created
        - Will take the place of one of the gems in the chain
        - Each subclass of SuperGem does something special when used such as destroying all gems in a cross, destroying all gems of the same color, etc.
    - "Locked" gems will appear randomly.
        - These need to be used in a chain more than once to go away.
        - Not a distinct class -- being locked is a feature of class Gem
        - Therefore SuperGems can also be locked gems
    - Make a shop from which players can buy special power ups with points earned.

- Scoring system -- only for limited moves game mode
  - Leaderboard system
  - Rankings are saved in a .csv file
  - Uses binary search to add new entries in proper spot
- Two game modes
  - Unlimited moves: Player keeps playing until no possible moves left or until they quit -- point penalty to keep it balanced
  - Limited moves: Player has limited number of moves
- Arrow key inputs
  - User inputs w/a/s/d to move their “cursor” in the 2d array and enter to select first gem to swap. User will do the same thing to select the second gem.
    - Uses Scanner class to get text input entered by the user
  - Will use ANSI escape sequences to seamlessly update screen

Reference: http://ascii-table.com/ansi-escape-sequences.php

Minimum viable product:

- At minimum, the user would be able to view the gems on the screen.
- Select gems to swap to create chains of gems
- Created chains are removed and replaced by a gem of random color
- Player will play for a certain number of moves after which the game ends.