In this assignment, you will add animation and undo to your project. You have so much time because of Thanksgiving and the next midterm exam. The solution (without extra credit) will require very few lines of code.

**Animation**

Add another command to your parser, animate, for which you will also have to define a token in the scanner. The command toggles the animation status: it turns on animation, if animation is currently turned off; and turns it off, if it is on.

When animation is turned on, the move command animates the movement of knights. Other commands may also animate, if you attempt extra credit.

You can choose the number of steps in an animation and the pause between animation steps.

For extra credit, you can also allow customization of the number of steps in an animation and the pause between animation steps. You should define special commands to do so.

For extra credit, animate the passed and failed commands. If you are animating the movement of a knight from one collection to another, you will have to animate it while it is in one of the collections. Animating it while it is in the original collection will not work because animation should be the last step in the deletion method. Therefore, animate it after it has been put in the new collection.

For extra credit, rotate and move a knight/guard as it falls down.

For extra credit, use the observer pattern to create coordinated animations, where the end of one animation starts another animation. This means the object affected by the second animation should be an observer of the object affected by the first animation. You can use this approach to ensure, for instance, that when a knight passes or fails, first this knight moves, and then the other knights move their position in the original collection. Depending on what animations you coordinate, this feature may require rewriting your original code.

All of these extra credit features will take time, so attempt them only if you are caught with work in other courses.
**Undo**
Add undo and redo commands to your parser, which undo and redo the last move method invocation. In animation mode, the undo of a command should also be an animation.

For extra credit you can also undo and redo other actions such as the say, passed, and failed commands.

**Global Objects**
There are no constraints on how you do your implementation. However, depending on you have implemented your project and how much extra credit you do, you may find it useful to keep track of the animations status in a special animation controller object. In addition, you may wish to keep references to both the history undoer and animation controller (which are shared by all classes in your program) in a global object that provides static getter and setter methods to set and get these two objects.

**No Reclaiming of Points for Assignment 12 and 13**
Because of the lag between assignment submission and grading, you will not get this and the other assignment back in time to fix all errors by December 9. So take extra care while doing this and the next assignment – your mistakes are not undoable!