Instructor: Diana Cukierman.

CMPT 125 Final Project: "Wizards, witches and horcruxes" game, ©Diana Cukierman[1]

B. Description of the game in more detail:

1. At the start, the user is asked whether he/she would like to play. A new world is created each time a new game starts, with n cells (n is provided by the user). See details about the dialog with the user below and in the sample runs.

2. A number of dementors, elves and 7 horcruxes are placed in the various cells (but not in cell #0). The number of dementors and elves per cell is determined randomly but within some ranges (min and max per cell) based on user provided values. Voldemort and Dumbledore are also placed in some random cell (not cell #0). Any of these creatures and/or horcruxes could be placed in the same cell or not.

3. The two wizard-witch players start their search in cell # 0. The first player moves first.

4. The two wizards-witches players PLAY ALTERNATING, and their paths may cross , i.e. they may be at the same cell at the same time.

5. When it is the player's turn, a program-simulated player will roll a die once or two times depending on the player type. The die value will determine where to advance by moving forward from the current position as many positions as the die indicates. The board should be considered circular, with cell# 0 positioned after the last one (cell #(*n-1*)).

6. "user" players directly indicate the cell number they want to go to and do not roll the die.

7. Once a player arrives to a certain cell, the following happens and/or should be checked, in this order:
   a. If Voldemort is in that cell but Dumbledore is not, then the player dies.
   b. If Voldemort is in that cell but Dumbledore is also there, then neither Voldemort nor Dumbledore affect the player.
   c. If there are domestic elves in the cell then the player gains life points, as many points as there are elves (i.e. 1 point per elf)
   d. If there are dementors in the cell then the player looses life points – as many points as there are dementors (i.e. 1 point per dementor). Getting down to 0 points kills the player.
   e. Since the player had a fight with the dementors, after the player was deducted the points because of dementors, the cell will have one less dementor. This is good news for the other player or this same player if any of them visits the same cell again later.

[1] Some terminology and some characters are inspired in Harry Potter's books, by J.K. Rowling. Some very simplified and approximate explanations are provided about this terminology and characters. On the other hand, while using this terminology this game is not describing in any way any of the Harry Potter books or films.

f. If the other player was already in the same cell when this player arrived the interaction with the other player will depend on the type of players this player is – see below.
g. After the interactions with the creatures and with the other player in the cell, if the player is still alive and there is a/are horcurx/es in the cell then the player collects the horcrux/es.
h. Then the player will wait for the next turn in the game to continue travelling unless he/she died.

C. Types of players

8. ***Implementation notes:***
    a. To start implementing the system it is suggested that you only consider one type of player, with the behavior of the ***naïve*** player (explained next).
    b. Once you have developed the game for one type of player you are recommended to expand the game including the other classes of players, relying on inheritance and polymorphism in Java.

9. There are three types of computer simulated wizards-witches players: a ***naïve*** player, a ***helpful*** player and an ***evil*** player. A 4th kind of possible player type is the ***user*** player.

10. The two players in one game can be both naïve, one naive and one helpful, one naïve and one evil, one naïve and one user, and so on. That is, any combination of two types of players may play in one game. The user will be asked to determine which kinds of players will play, at the start of a game.

11. The types of players will differ on how they advance to their next position and what they do if they encounter another player in the cell where they arrive.

C.1 Rolling the die and the different types of players

12. A ***naïve*** or a ***helpful*** wizard-witch player rolls the die only once and moves forward in the board as many positions as the die indicates.

13. An ***evil*** player will be able to roll the die once more (i.e. the evil player can roll the die up to 2 times). The evil player will roll the die a second time if the first die value takes him/her to a cell that does not have the other player. That is, the evil player is trying to go to a cell where the other player is. No matter what the second die value is, the evil player moves according to that value.

14. A ***user*** player will be able to see all the information in the board and will simply type the cell number where he/she wants to move to (the user does not roll a die).

C.2 Arriving to a cell. Interaction of the different types of players

15. When ***any kind of player arrives to a cell*** he/she will be affected by all the creatures that are in the cell as indicated above (Voldemort, Dumbledore, elves and dementors) regardless of whether there is another player there or not.

16. After a player was affected by the creatures in the cell,
    a. <u>if the newly arrived player is still alive and there is no other player in the cell</u> , the newly arrived will pick up all the horcruxes in the cell (if there are any) and wait for the next turn.
    b. <u>if the newly arrived is still alive but the other player is in the same cell and is dead</u>, the newly arrived will additionally pick up all the horcruxes from the other player.

17. After a player was affected by the creatures in the cell, <u>if the other player is in that cell and both the newly arrived and the other player are still alive</u>
    a. If the newly arrived player is ***naïve,*** the newly arrived will just wait until his/her next turn (and cannot pick any horcruxes from the cell because the other player, who arrived first, picked them up already if there were any).
    b. If the newly arrived player is ***helpful*** he/she will (similarly to the naive player) not have any horcruxes to pick up, but the helpful player will do a deal about life points with the other player (who arrived first) as follows: they will both share their life points. That is, each of them will end up with half their combined life points (hence players could potentially have life points with some fractional value).
    c. If the newly arrived player is ***evil*** (and one may recall that the evil player tried, on purpose, to land in a cell where there is another player) ***the evil*** player will take all the horcruxes from the other player and will also take half the life points from the other player to his own benefit!
    d. If the newly arrived player is of ***user*** type, the ***user*** player will have been given the choice when the game starts, about how he/she wants to interact when arriving to a cell where there is another player, if as naïve, as helpful or as evil. (Notice that one can test the various player types by having the user act with the various player styles.)

<u>***D. End of the game – Showing results***</u>

18. Recall that the game is over when one of the (or both) players die or one of the players collected the seven horcruxes or until a certain number of turns (number provided by the user) took place. (One turn involves one player move).

19. The winner is the one who is alive and collected all the 7 horcruxes.

20. There may or not be a winner when the game is over (although , as a side comment, all is relative – being alive with no horcruxes is much better than being dead – but such are the rules of this game).

21. Once the game is over the user is shown final results including winning results and information associated to the current state of the board and the state of each player (see next item, dialog with the user).

.

E. Dialog with the user, initializations, showing results

22. The dialog with the user will be done via a text user interface and optionally (for bonus points) also through a graphical display (but just drawing output, using the methods in the DrawingPanel class provided in the Lab week #11 exercises) .

23. You should start the dialog displaying the name of the game and also the authors names and student numbers.

24. Information that will be asked from the user before starting the game :
    a. The maximum number of turns to play . One turn includes that one player moves.
    b. The number of cells in the board.
    c. The range of values in the cells (minimum and maximum of dementors and minimum and maximum of elves that each cell can have)
    d. For each wizard-witch player the user is asked:
        i. The player's name,
        ii. The kind of player he/she is (naïve, etc.) You should let the user know what types of players are available in your implementation in a brief list so that the user may choose from such.
        iii. Each player initial life points

25. Information that will be shown to the user (to show the user the evolution of the game):
    e. after each player moves it should be shown:
        i. the die value obtained (if there was any die rolling)
        ii. the board state (all the cells state), so that each cell indicates the information associated to it: number of horcruxes, dementors, elves, whether Dumbledore or Voldemort is there and which player is there at the moment
        iii. the players state including: name and type of player, current cell position, current life points, current horcruxes, path that the player has visited so far as a list of cell positions visited, ordered by visits done as the game unfolded.
        iv. any message whenever needed, indicating unusual events that happened – such as a player being robbed, losing all points, etc.

    f. and at the end of the game it should be shown:
        i. the board (same details as shown after every move)
        ii. the players state (same details as shown after every move)
        iii. who the winner is (if there is such)

g. Additionally, as an optional feature (providing bonus points), after each player moves and also at the end, the board will be graphically drawn. You could indicate where horcruxes are and/or how many creatures there are in each cell and where the special wizards and the players are. You can color code and/or add legends. It is suggested that you leave the graphical details to the end and only add more details to the graphic part when your problem is solved. (on the other hand, having a graphic visualization may help you follow the game better)

G. Validating user input – treatment of exceptions

26. As a first stage check the correctness of data provided by the user without checking exceptions, but just validating ranges of values. After your program is working, for bonus points, add treatment of exceptions. Exceptions will be seen later in class (and is available in the textbook). Including exceptions for this project involves:
    a. try/catch of one exception,
    b. declare some method which throws some exception,
    c. throw some new exception in some method

F. Some suggestions

27. Check the assignment and readings section in the course website as material is posted.

28. Sample runs and initial documentation are provided (considering only one type of Player) . Check the assignments section.

29. You have to display essentially the same information as in the sample runs but you may change the format.

30. Inheritance and polymorphism should be used to implement the different types of players. **Using inheritance is a required feature.** Yet, partial points will be provided for solutions not including inheritance.

31. You will have to do some changes to the design to include the various types of players with inheritance. You could change the design further as well, but keep in mind that an Object Oriented solution is expected (for example, you should not develop the whole game in one single class). Check the notes discussed in class about the design, ask if you have questions.

32. For drawing , use the DrawingPanel class provided (lab week #11) and see the examples there about how to use the various methods to draw . The drawing part is not required, it will help you better visualize the game, but it will also take time for you to develop. The drawing methods will not be seen during lectures and it will not be tested in exams either.

33. Consider working as a team!

H. What to submit

34. Documentation:

    a. Simplified UML diagram – (simplification analogous to the assignment RollTTT, i.e. without including the variables and methods, but just the class boxes with those areas empty). You should include however all the relations between the classes.

    i. A text file including the type, name and a one line English description of the instance variables in the classes if the variable name is not self evident. Organize this description by classes.

    j. A text file including the signature of the methods in each class and a one line English description of the method if the variable name is not self evident. Organize this description by classes.

    k. Flowchart of top level diagram of the whole application (you can draw it with a graphic tool or handmade and scanned)

    l. Text file with a couple sample runs.

    m. Text file describing how you distributed the tasks between the team members.

35. Code:

    n. All the source code (.java files) with all the classes you need to be able to run the game.

36. Keep a copy of everything that you submit.

I. Additional Assumptions

37. For bonus points you are invited to invent a different type of player.

38. Additional assumptions can be made as long as they do not alter this description of the application. Any additional assumption should be justified and be well documented. If in doubt, ask the teaching staff.

39. If you would like to show a demo of your game in the last class contact the instructor.

End of Project #2 detailed description