Changes

GUI

1. Added private Interface variable
2. Removed create file menu

Round

1. Need method getActivePlayer() as need a way to access players and while Game is in a wait any value returned from a method call is returned null thereby negating any purpose of get methods called there.
2. Added way to tell people who goes first otherwise how would the players know. Needed for both debugging and practical purposes for a game to be runnable in any sort of semi-organized fashion. See Interface # 6 for further arguments on reasons why informing players is necessary for gameplay.

Game

1. Must in constructor make computer always go first as in order for the program to run it is dependent on actionPerformed in Interface being called and never a place for the computer to go first.
2. Including extra parameter in constructor (int) as without it Game has no way of knowing whether or not a computer should exist as that is called in startGame() in Interface as it is there where the player choice is made.
3. In constructor needs to be while(!decideGameWin()) as design returns reverse values for what is needed
4. Constructor in while loop needs to call determineFirst() otherwise how would it change who is the respective first player each time in a round as according to design it changes.
5. In determineFirst() as stated before human player always needs to be first hence extra if checking it.

Interface

1. Interface needs to be able to make Player object and it does to use its methods as needed and checking whether or not Player is instance of computer for computerTurn().
2. In actionPerformed use if “New Game” else. Using if’s is too vague a purpose.
3. Added private Boolean ans as otherwise no way to keep track whether the answer from the database is correct or not to use as comparison in checkAnswer().
4. Need to use **if**(round.checkRoundWin()) **return**; in multiple places to check if round is over after all players go and notification to go out for round ending and the game to continue. Otherwise, the method was never called from anywhere else.
5. In design said call swapPlayer(). That method did not exist it was switchPlayer(Player).
6. Added extra if in case 2 humans playing. Provides data on whose turn it is for the human aspect of the game. Since the game has 2 human players the players must understand when they are to go or not. As the program is dependent on player choices having the question of whose turn it is can possibly cause confusion which should be avoided at all costs for proper gameplay to occur with the program concluding successfully.
7. checkAnswer requires a int as argument also. It needs it to use for updating GUI as well as checking for opponent win.
8. In run() added if to return if JOptionPane closes and timer no longer needed and should be stopped as it can interfere with gameplay otherwise.
9. In startGame() added if rep==-1 as need to take care of abrupt termination and how it should exit program not run through and cause problems with the rest of it (not know if have computer or not and can possibly get incorrect information on the matter).
10. Need to initialize numQuest in startGame() for obvious reasons. Need getTextArea there too as need references to it (again for obvious reasons).
11. In computerTurn() displaying all information is too vague. Displaying all relevant information that is not seen elsewhere.
12. Receiving both question and answer in loadQuestion(). Did not include loadAnswer(). loadAnswer() is completely and utterly useless and would be a never ending method as it has no way of knowing what the question chosen was to pull up an appropriate answer for it. Even giving it the question as a parameter is useless as hypothetically 1 question could be asked twice with different answers or be a part of a larger question causing it to pull the wrong values. Even if the database was constructed with that last point in mind it does not change the fact that it would still need to be changed to include a way to determine the proper question. As such calling the method as is would result in errors. While changing method parameters could be done to allow for it when needing to fix a problem it is logical to go the better way (call it in loadQuestion() then to fix a problem terrible way (like putting in loadAnswer()).This reasoning is for calling it in actionPerformed. However, it could technically be switched to be called in loadQuestion(). However, again it is useless as loadQuestion() in a way does all its work for it anyway. Therefore, loadAnswer() itself is a method with no purpose and would only be detrimental to call in the game.