Following are the assumptions that have been set for the game:

* The user can create a maximum of 5 users. The number 5 has been allocated because the maximum length of the words in input file is 6. So, having around 20 players to guess a 6-letter word would seem pointless.
* The user is supposed to input the file name. **The name of file that should be INPUTFILE.txt**
* In manual mode, after the game starts, each user is given a chance to guess letter on a round robin basis. E.g. After player 2 finishes guessing, player 3 gets his chance. Based on each user’s guess correctness, the output on the screen is updated.
* **YOU CAN ONLY RUN THE GAME ONCE IN MANUAL MODE. NO OPTION TO REPEAT THE GAME HAS BEEN PROVIDED.**
* **In automatic mode, once the number of players and the number of times the computer must play has been given as input, the computer plays for the number of times specified with the same set of players and different question picked randomly from the input file for each game.**
* **After the computer plays for the specified number of times, it prints the results and the program exits.**
* **You cannot switch between modes in the program i.e. Once manual mode has been chosen, you play in that mode and the program exits after the game is over. Same applies to Automatic mode.**

**Activity-3**

**The reason I explain the classes and methods of Activity 3 is because some of the classes of activity 3 is same as activity 2. Activity 3 only has an additional class of autoplay. P2A3\_SIVARAMANVENKATARAMAN\_GAME\_aswinras class**

**Functions**

* **Public void setFileName()**
  + The main purpose of this function to set the filename attribute of game class
  + Each game has a filename attached to it. Through this function, we make sure the user enters the correct input file to play the game.
  + **Error Scenarios:**
    - **If you give any other file name other than INPUTFILE.txt as input, then the function will ask the user to re-enter the filename.**
* **public void setNumPlayers()**
  + **Each game contains a certain number of players attached to it.**
  + **This function has been used to set the number of players who are going to play this game**
  + **Error Scenarios: If the number of players entered is less than 1 or greater than 5, this function will throw an error**
* **Public void createQuestion()**
  + **Each game also contains a question.**
  + **This function reads the input file and takes a random line from it.**
  + **Then from the line, it extracts the question and hint.**
  + **Then using this, it creates a question object**
  + **No error scenarios have been handled**
* **Public void displayInstructions()**
  + **This function displays the instructions required to play the game to the user**
* **Public String getRandomLine()**
  + **This function reads the input file named “INPUTFILE.txt” and fetches a random line from it.**
* **Public void createPlayers()**
  + **This function instantiates a certain number of player objects, based on user input**
  + **For each player, the player name is obtained as input from the user.**
* **Public void resetPlayerValues()** 
  + **This function has been written keeping autoplay functionality in mind**
  + **Since in autoplay, we use the same set of players to play the game for certain number of times. After each game, we use this function to clear the old values which these players had guessed, their old number of Ticks.**
  + **To summarize, this function sets all the player’s attributes to default values at the start of every game for auto play mode.**
* **Public void start()**
  + **This function is used to start the game.**
  + **The game ends if one of the players guesses the entire word correctly**
  + **The game can also ends if, for each of the users the number of ticks amounts to 6.**
  + **This function has a while loop, which checks the above two conditions i.e. if none of the users have guessed the entire word correctly and none of the users have ticks count which equals to 6.**

**P2A3\_SIVARAMANVENKATARAMAN\_AUTOPLAY\_aswinras**

* **This class extends from the game class. All the fields and methods in the game class has been used by this class.**
* **The only difference between this and the game class, is that this overrides the start method from the game class.**
* **In this start method, the input for guessing is taken from a random character generated between (a-z), instead of taking input from user.**

**P2A3\_SIVARAMANVENKATARAMAN\_PLAYER \_aswinras**

* **public P2A3\_SIVARAMANVENKATARAMAN\_PLAYER\_aswinras**

**(String name,int numTicks,int totalGuesses,boolean gameEnded)**

* + **Each user has a name associated with him.**
  + **Each player has a variable numTicks associated with him, to track the number of wrong guesses he makes. The color of the bomb is decided based on this variable numTicks**
  + **For each player, we have a field name gameEnded, to check if he is still alive in the game. This especially useful in multiplayer mode.**
* **Public void resetValues()**
  + **We use this function to reset the values associated with this player in the previous game i.e. number of ticks, the bomb color etc.**
  + **This function will be useful especially in Autoplay mode, where we play the game multiple times**
* **Public void setName(String name)**
  + **This function is used to set the name of the player**
* **Public void updateNumTicks()**
  + **This function is used to update the number of ticks i.e. wrong guesses made by the user. Based on this value, the bomb color of a player is updated accordingly**
* **Public void setGameEnded(Boolean flag)**
  + **This function is used to set the game Ended property of the user.**
* **Similarly we have accessors and mutators for Guesses and BombColor as well.**

**P2A3\_SIVARAMANVENKATARAMAN\_Question \_aswinras**

* + **Public P2A3\_SIVARAMANVENKATARAMAN\_QUESTION\_aswinras (String question,String hint)**
    - **Each question has a secret word and a hint associated with it.**
    - **This constructor accepts those arguments to set that.**
  + **public void updateInput(char c)**
    - **If the character guessed by the user is correct, the character is added to the guessed word.**
  + **Similarly, we have accessors and mutators for question, hint and input fields.**