|  |  |  |
| --- | --- | --- |
| Class Name | Interface | Description |
| Attributes | - GUI awesomeGUI | Points to a gui representation for use with the game to interact with the players |
|  | - Game awesomeGame | Points to a game that will be used through the interface by the players. |
|  | - TextArea awesomeTA | Points to a textArea in the GUI which will provide the scores of both Players and the countdown for the question timer. |
|  | - Thread awesomeThread | Points to a timer (15 seconds) that will be shown to the user to let them know how long they have left to answer the question. |
|  | - Boolean [] numQuest=false | Keeps track of which questions have been used |
|  | -Round round | keeps track of currentRound(). |
|  | -boolean showedPane | Keeps track if JOptionPane is showing for chooseAnswer(String). |
|  | - JButton benjaminButtons[9] | Buttons representing the board for the game |
| Methods |  |  |
|  | + actionPerformed(ActionEvent):void | Converts ActionEvent to string. Using if’s actionPerformed acts as follows:  Calls loadQuestion() and loadAnswer() into a String. Then it loads that String into chooseAnswer(String) which uses a JOptionPane to get the response of true or false which is converted into a boolean. If, they do not answer in time they are assumed to have gotten it wrong.Calls checkAnswer(boolean) and uses return value for assignSquare(int, boolean) in Round. If computer is next, run through computer turn by calling computerTurn().Uses swapPlayer() after every turn to change active Player.  Also, has a if statement for "New Game". If chosen dispose of gui, and set Game to null and set round to null to remove reference for garbage disposal. It then calls new Interface(). |
|  | + loadQuestion():String | Loads a question from database to be displayed to User based on randomness of unused questions by checking the numQuest array. If full, reset array to false. Add new used question to array by setting corresponding value to true. |
|  | + loadAnswer(): String | Loads a answer from database to be displayed to User based on randomness of answers belonging to question asked. |
|  | + checkAnswer(boolean): boolean | Compares Boolean choice of user to Boolean gotten by database of answers. Returns true if they are the same. False otherwise. If true, it should update the JButtons to reflect an 'X' or 'O' on the GUI (in method itself get information from method in Game to get Player representation 1 is Player X, 2 is Player O) and disable associated JButton. Displays information in awesomeTA. |
|  | + run(): void | Starts the timer from 15 seconds with a for loop.Shows timer in TextArea and updates it every second by using sleep on the thread at 1000 milliseconds. If the timer runs out it sets showedPane to false and if showedPane is false it calls JOptionPane.*getRootFrame*().dispose() to eliminate pane and continue on. InterupptedExceptions are caught and ignored. |
|  | +computerChooseSquare(): int | Chooses open square based on if center open choose it. If not, choose an open corner (represented by 0, 2, 6, 8) otherwise chooses other squares randomly. Returns the int representation of square . |
|  | + chooseAnswer(String): boolean | Starts thread after initialization to this interface. Sets showedPane to true. Creates a JOptionPane with a option dialog containing with parameters, in order (null, String of question and answer, "Choice", JOptionPane.YES\_NO\_CANCEL\_OPTION***,*** JOptionPane.QUESTION\_MESSAGE, null, String array containing true and false (create in method before with true first and false second), null).  containing a String retrieved from loadQuestion() and a String retrieved from loadAnswer() with the options of True or False. Returns a representation of the users choice as true or false (0 or 1). Convert to boolean with expression (ex: Assume I is integer. boolean b= i==0). Timer (thread) is started and displays in TextArea the countdown. If the 15 seconds run out without an answer the JOptionPane is disposed of (returning a -1 value, also occurs if the choose to exit out of it)and they are automatically judged as incorrect (called by Round). After JOptionPane closes either by user or by force it changes showedPane() to false.  Informs user if correct or not by updating the TextArea with that information. |
|  | Interface() | calls startGame(). |
|  | + startGame(): void | Shows JOptionPane to ask if there is one computer or two humans (ask if want computer), displays rules of the game in TextArea. Initializes GUI, get the buttons from GUI and Game. After puts Game Over in awesomeTA and displays scores of Players from Game. |
|  | +resetBoard(): void | resets board by calling setButton in GUI. |
|  | +computerTurn(): void | calls computerChooseSquare(), loadQuestion(), load answer(), and chooses true or false randomly (50% chance) and calls checkAnswer(boolean) and calls assignSquare(int,boolean) in Round. Displays all information to awesomeTA. |
|  | +assignsRound(Round):void | called by Round and assigns argument to round. |