CS2450

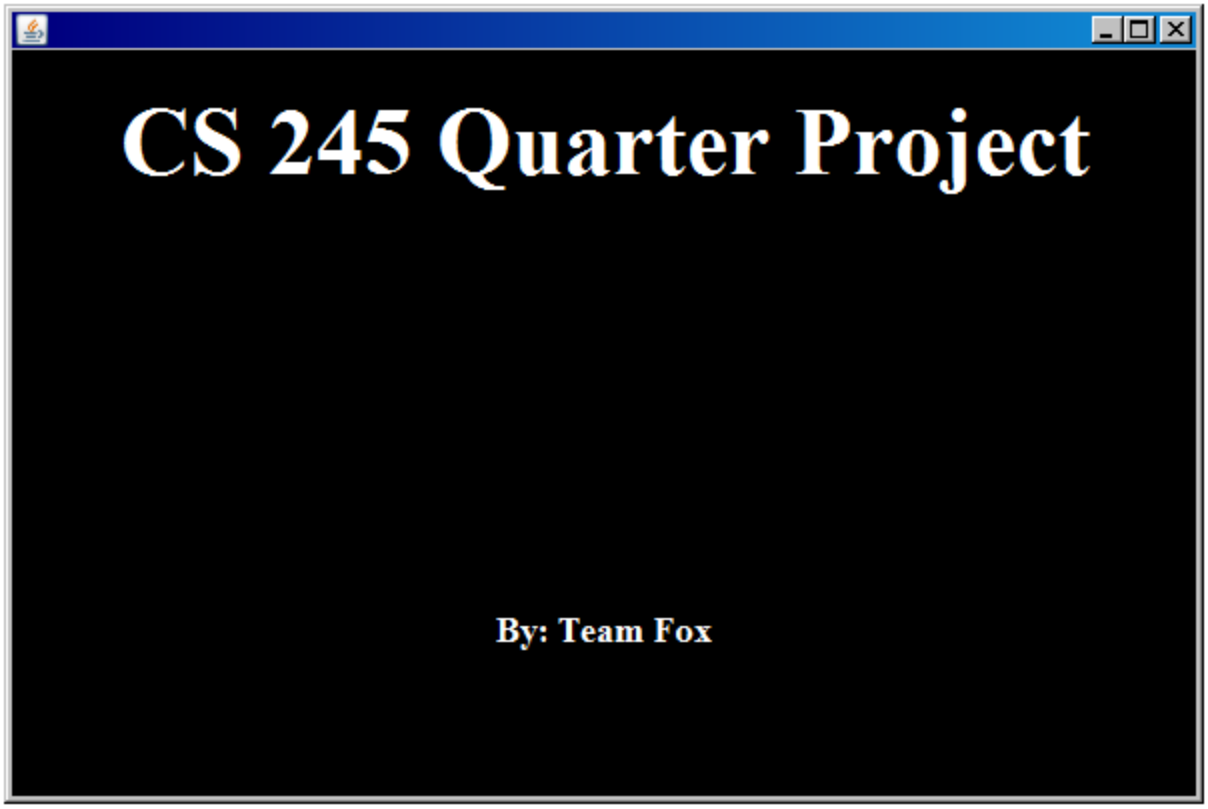
Point and Click Game – v.1.3

Introduction – "Point-and-click" games are a common type of game where the player typically uses a mouse to interact with the environment and solve puzzles.

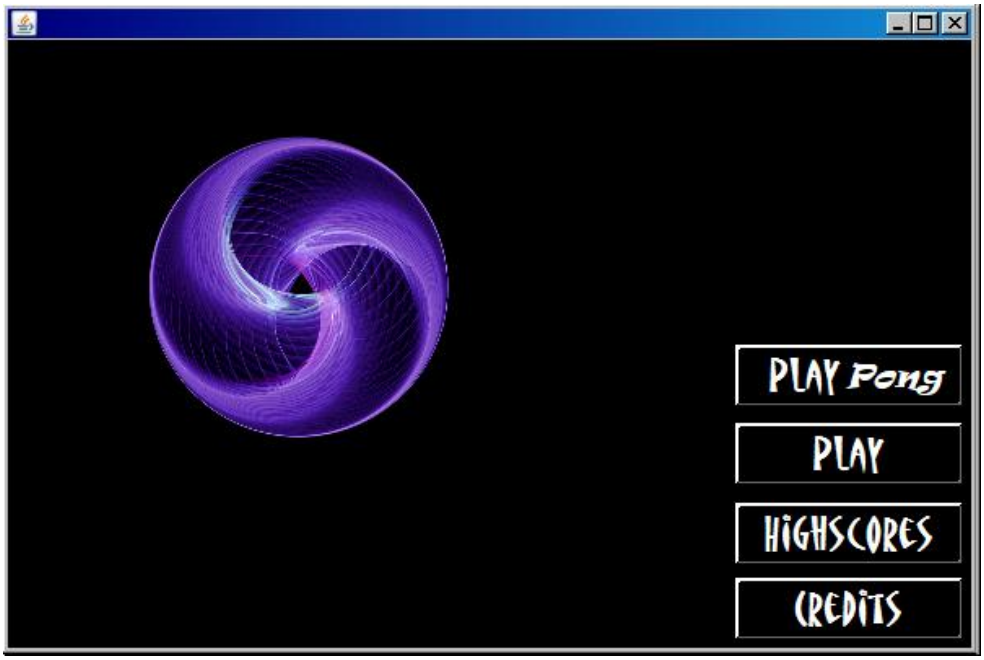
Task – create the following GUI:

Your program must have the following functionality:

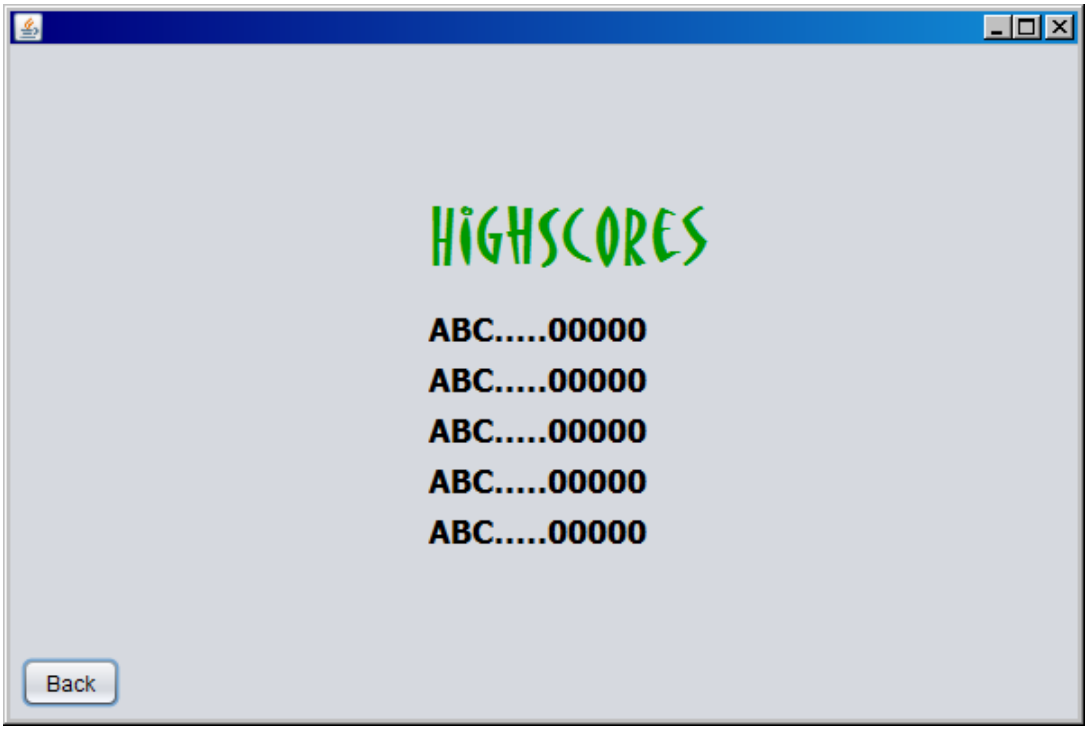
1) It will be centered on the screen in a 600 x 400 pixel window.
2) The program will have a start up screen which will display a name chosen by your group for the project, as well as a chosen team name for your team. This information will be displayed for 3 seconds, and then disappear - the ***function buttons*** will then be displayed in the display area.



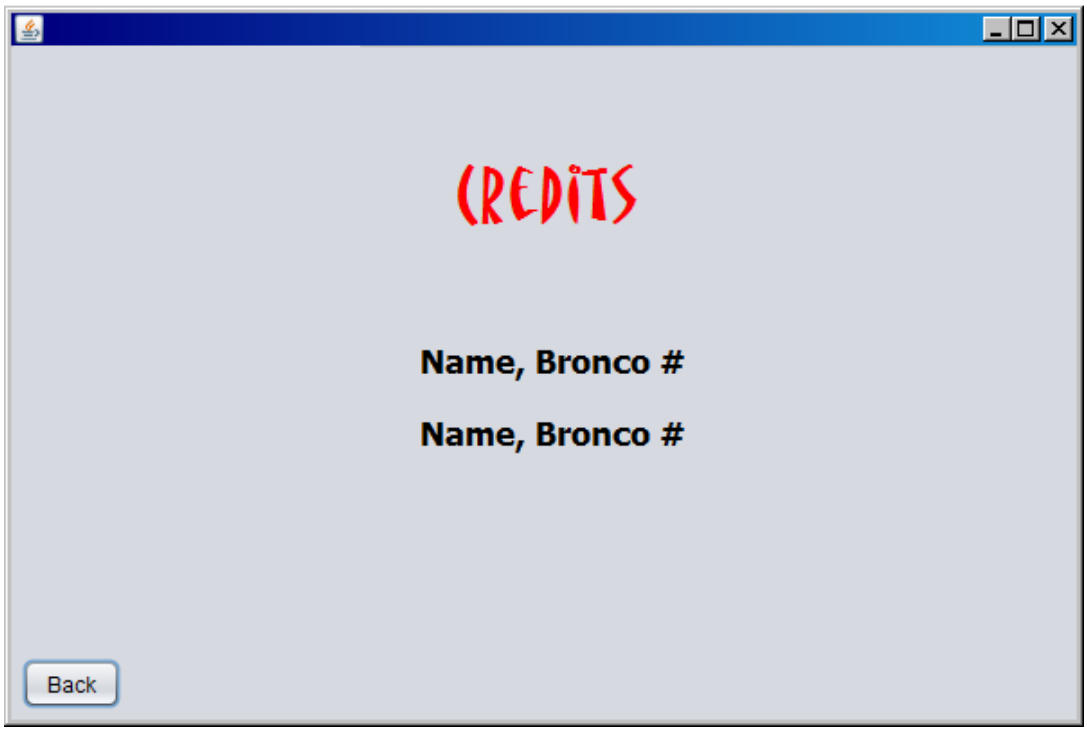
3) After the start up screen the display will switch to the following where a chosen symbol for the project will be displayed along with the three function buttons, Play, High Scores, and Credits. **You will add a new button titled Play Pong. It is from here that you will play the pong game.**



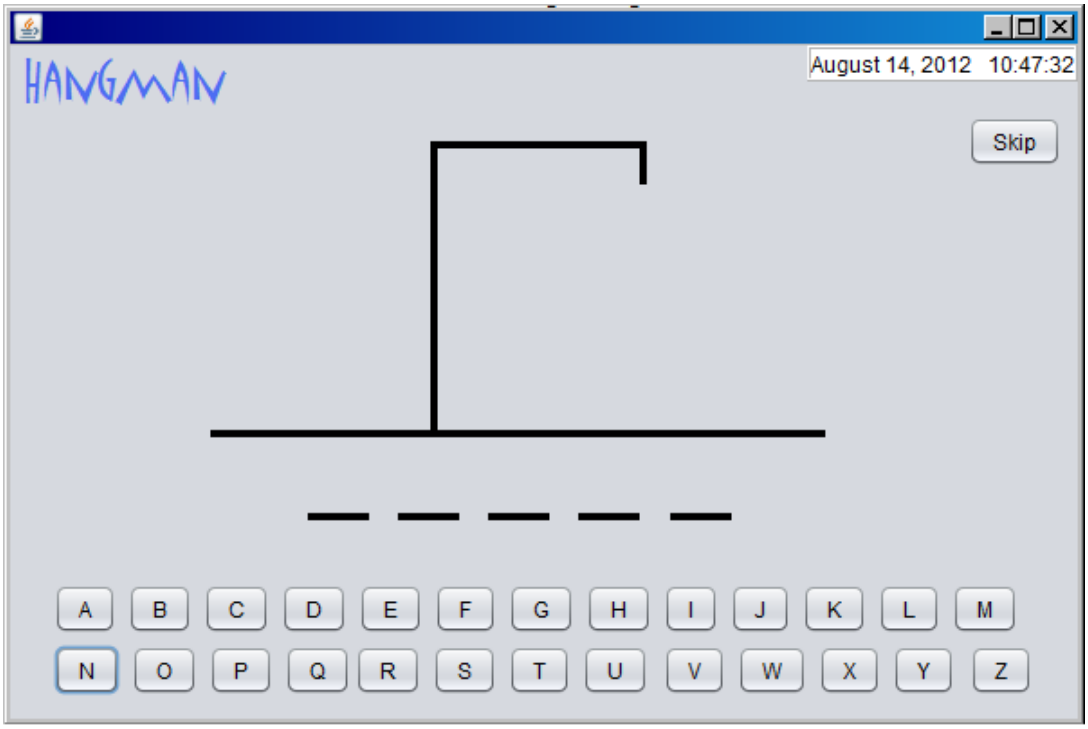
4) When High Scores is clicked, the display will switch to the following. The high scores will be saved to, and read from a file. Hitting the Back button takes the user back to the function buttons screen.



5) When Credits is clicked, the display will switch to the following. The Names and Bronco #s should be of those from every member of your team. Hitting the Back button takes the user back to the function buttons screen.

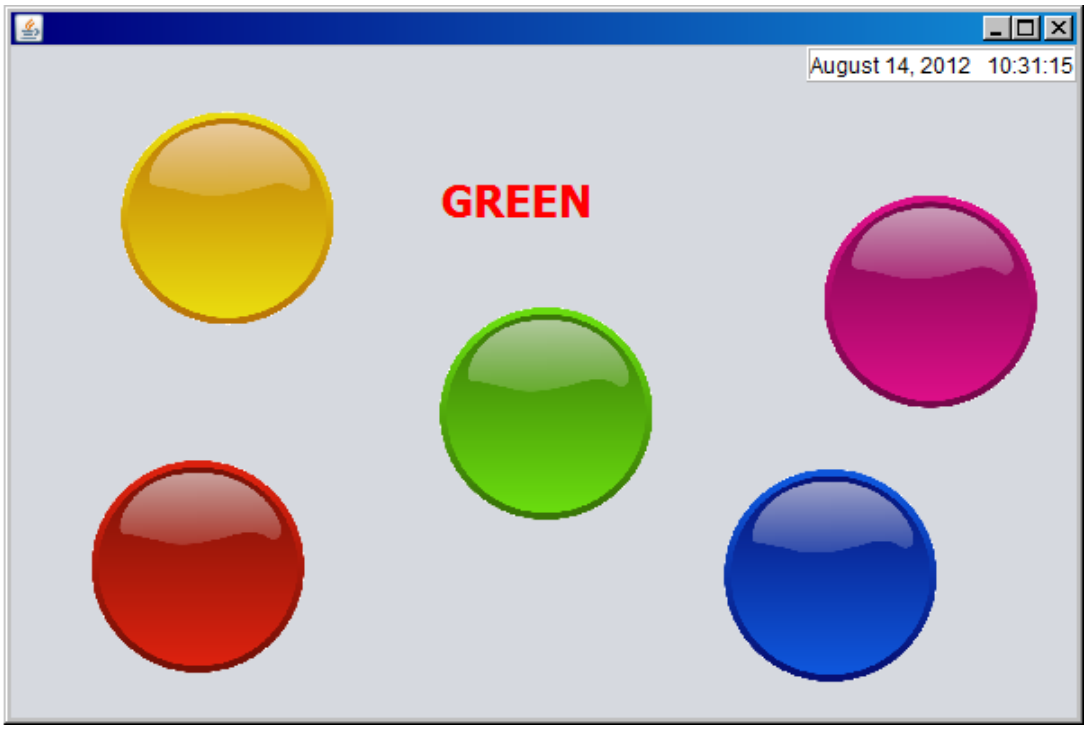


6) When Play is clicked, the display will switch to the following.



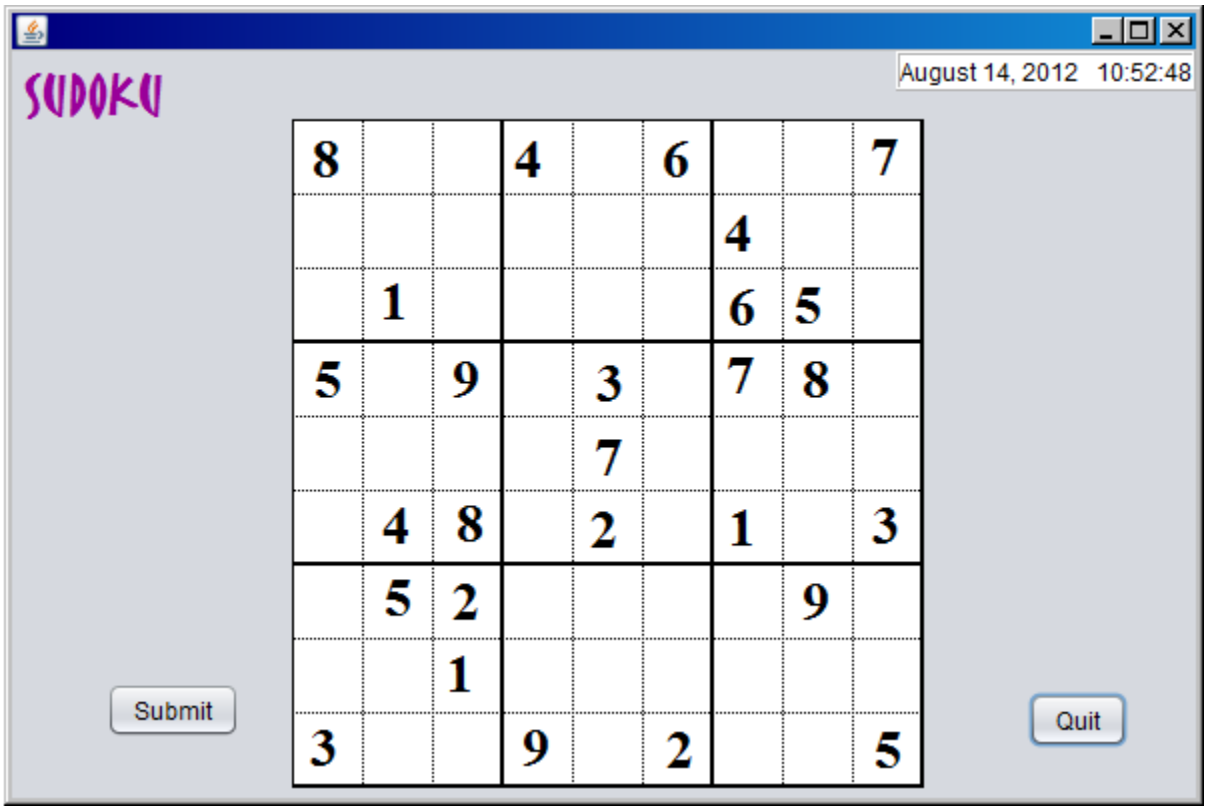
7) Must display current time and date (it should update each second).

8) For this game a random word will be selected from the following list (abstract, cemetery, nurse, pharmacy, climbing). A typical graphic for a game of “hangman” is displayed with a number of lines beneath it corresponding to the number of characters from the selected word. The user may then click on one of the buttons to guess a letter which may be in the selected word which they are guessing. Each time the user guesses a letter if that letter is found within the selected word then the letter is drawn to the screen above its appropriate space marked by the lines below the “hangman” graphic, and the button for the letter which was guessed should be disabled. If a letter is guessed and it is not found within the selected word, then you should alert the user, draw a portion of the man being “hanged”, disable the button which was guessed, and allow the user to try again. You should keep a running tab of the user’s score and display it on screen; the user begins with 100 pts possible if they correctly guess the word without any mistakes. For every incorrect letter which is guessed, 10 pts should be taken away from what is left of their total possible points. For every correct letter which is guessed, their score is left unchanged. The user’s score should never be lower than 0, or exceed the mentioned points possible. After six incorrect guesses or once the user has correctly guessed the word or if the user clicks “Skip” to skip the game, the display will change to the following (If the user decided to skip the game, then their score should be 0).



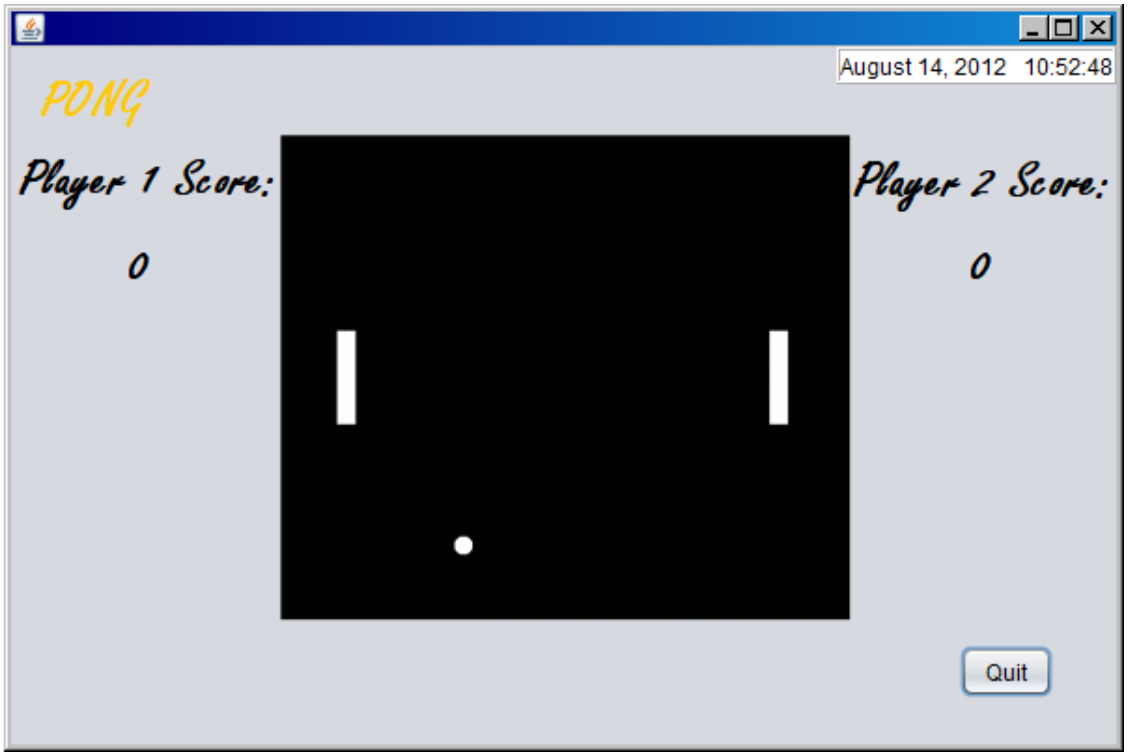
9) For this game the name of a color will randomly be given, for example Green, from the following list (Red, Yellow, Green, Blue, and Purple). The text will be randomly colored of the five colors from the list. Five buttons, one of each color, will be randomly placed in the window. When the user’s mouse moves over one of the button’s the button will become highlighted. The user is to

click on the button which shares its color with the color of the text (Not the word given by the text). Give the user 100 pts for every correct button pressed, 0 for every incorrect button. After five rounds, the display will change to the following.



10) For this game, the Sudoku board will be displayed for the user with the shown layout. The user will be able to click on the empty boxes and input a number from 1-9. (If the user inputs a number that is not 1-9, alert the user and have them correct it). The user will be able to change/edit any box that they were able to input a number for from the beginning. Once the user has a solution, they will click "submit" to check their solution or "quit" to end their game of Sudoku and move on. Scoring will be done in the following fashion: the user will begin with 540 possible pts which will be given if the overall solution is correct. If the overall solution is incorrect, 10 pts will be taken for each box that is incorrect (only once per game). Points can only be taken from the total possible, never added, and points are only taken the first time a box is submitted with an incorrect answer. If their overall attempt is incorrect, alert the user upon submission and allow the user to try again.

11) After the user either correctly solves the Sudoku puzzle or quits, display an end page where the user's score is displayed in similar fashion to the high scores screen including an "End" button (instead of back) to take the user back to the function buttons screen. If the user's score is higher than the lowest score saved in the high scores, give the user the option to save their score to the high score's list. Provide a method for the user to input their initials. The game should allow for repeat play through with all elements reset as if playing for the first time.



12) You will now be creating a game of Pong. This game will only be accessible from the function button screen by clicking on the "Play Pong" button as it will allow 2 players to play. The game should display the title of the game, should display the current time and date (updating each second), there should be a "Quit" button which will allow the user to end the game and return to the function button screen, there should also be a title of "Player 1 Score:" and "Player 2 Score:" with the appropriate score for each player listed below it. For this game there should be a circle drawn to the screen as the "ball" which bounces off the top and bottom of the game area. If it hits the right side, it will stop and display that Player 1 scored. If it hits the left side, it will stop and display that Player 2 scored. The ball can bounce off of a player controlled rectangle (known as the paddle, player 1 controlling the left paddle and player 2 controlling the right paddle). Player 1 will use the w,s keys to move their paddle up (w) and down (s). Player 2 will use the arrow keys to move their paddle up (up arrow) and down (down arrow). To start the game and begin having the ball move, or to have the ball move after someone has scored, the players should press the "Space" bar. Have a random 2D direction decided by your program and fire the ball off to kick start game play. Until the ball begins moving, the ball should remain stationary, but visible, in the middle of the game area. For scoring, both players will begin with a point total of 0. Each time either player scores, their score should increase by 10 pts. Their score should never decrease. Once either player's score becomes 100 the game ends and the appropriate player who reached this score wins. Notify the users that the game is over and who won, then allow for them to end the game and return to the function button screen. The game should allow for repeat play through with all elements reset as if playing for the first time.

13) Implement the following: 1. tool tips for every component, 2. pressing F1 pops up display showing: your names, Bronco #s, project name, and term, 3. pressing the escape key exits the program.