Students, in this phase, you have to add software interrupts in your project.

1. In this phase, you have to implement a start screen in your project. The start screen should show your roll nos, your(developer) names, name of the game, any loading page(optional). It can have its own animation or animations. Be creative. Do whatever you wanna do. Show whatever you wanna show.

2. You also have to implement an instruction screen that should appear after the introduction screen and should present the instructions, such as the role of arrow keys in movement, role of fuel, collection of coins, quitting method, etc.
3. The main screen should be static and should show a message,"Press any key to start the game."

4. You also have to implement a confirmation message screen, i.e. the box should appear during the game on pressing esc key, in the middle of the screen and prints the string, "Do you want to exit?" and shows options, y or n This should not appear during the introduction or instruction screen. Be creative in how you implement the confirmation message screen. This should take you back to the main screen.

5. You should also create an ending message screen that will be shown when the user quits, the fuel ends, or the car crashes into any other car. Do not worry yourself with collision yet, you will implement that in the next phase. There will be a message screen showing the cause of game end, similar to confirmation message screen, a box in the middle of the screen.

6. Then the ending screen will show the player's name, his roll no., his coins and an instruction to press enter to go back to the main screen by pressing space or to exit the game by pressing escape and using confirmation screen. Yes you have to input player's details before the game starts, i.e. before the instruction screen. and store it somewhere, anywhere.

7. If the game has not started and the user presses esc during the main screen or during name input screen, or the instruction screen, i.e. after the introduction screen, the confirmation screen should appear and on pressing y, the game should end such as during the end screen. The Dosbox screen should go black with dosbox cursor appearing and taking input. The stack should be managed properly, i.e. after leaving the game other programs should work fine. All interrupts should be unhooked properly.

Remember, it's all about presentation at the end of the day.
Also, the required screens can be kept simple and can be creative. Do not overwhelm yourself. Just do, how much time and other courses allow you to work on the project. We have more phases incoming.

Note: Your main screen should contain a blue car placed randomly. The random part is the selection of the lane. During the game, after a certain interval(random or hard coded(recommended)), a new car should appear, placed on a random lane. In any case, there should be enough time spent, i.e. enough space is created between two blue cars that the game is playable and looks pleasant.
Note: The blue cars should not randomly pop on the screen, but they should smoothly enter and smoothly exit the screen as in the original game, i.e. they should fade in and fade out, rather than popping on the screen.

Important Instructions:

Make best use of programming practices. Make subroutines properly, name functions and variables properly. One function should do only one task, so properly distribute tasks in different functions. Controller functions should just be a sequence of function calls. Code should be readable; properly commented and aligned. Avoid hardcoding as much as possible, keep everything configurable. Use global variables for configuration, i.e. anything that you're hard-coding should be placed as global variables, so that they're easily changeable.