NATIONAL UNIVERSITY
of Computer & Emerging Sciences, Lahore

Computer Programming CS103
Semester Projects

Deadlines:

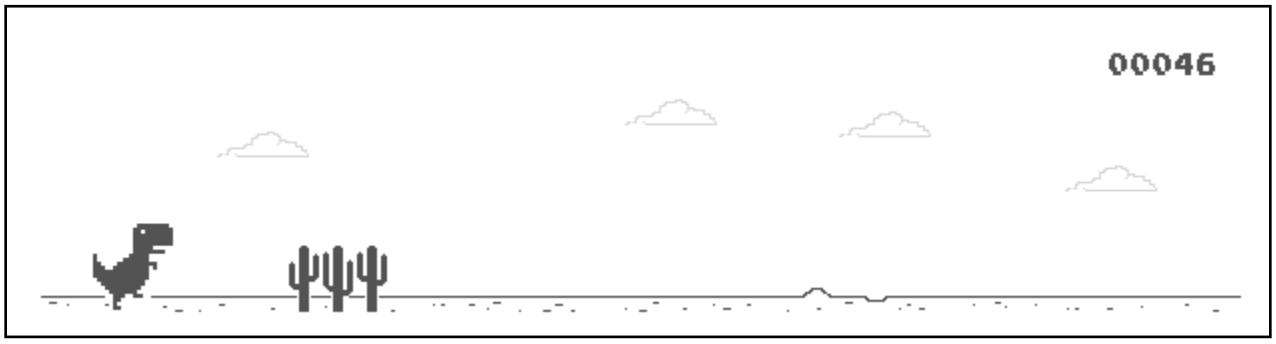
1. Project Selection and Group Members:	**5:00 p.m** - Tuesday, 6 Nov, 2018
2. **Project Deliverable 01**:	**5:00 p.m** - Monday, 12 Nov, 2018.
3. **Project Deliverable 02:**	Will be announced later

Project 01: Dinosaur's Game

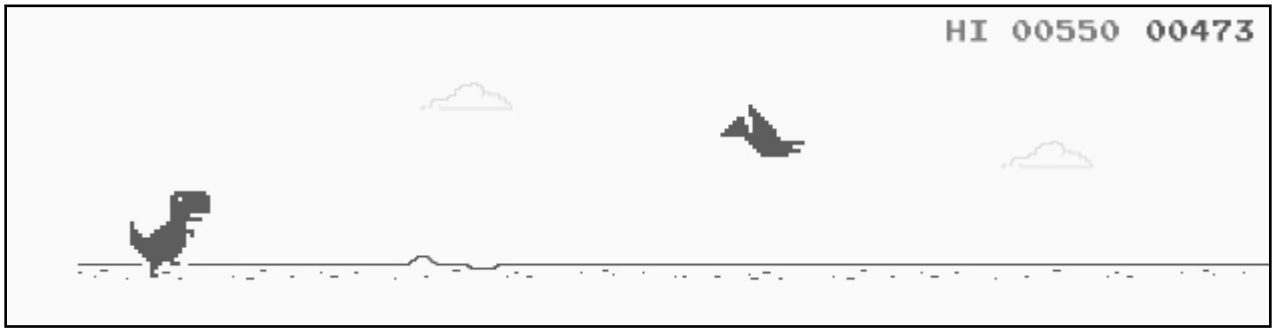
Maximum No. of Groups Allowed: Nine groups each consisting two members at most. Project will be allocated to **Nine** groups based on first come first served bases.

Project Description:

An often-overlooked Easter Egg in Google Chrome is the dinosaur game, which appears when you try to visit a website while disconnected from the Internet.
The Chrome dinosaur game is a simple infinite runner, which sees you jump over cactus, and dodge underneath obstacles. Controls are basic. Press **space** to jump (and to start the game), and the **down arrow** to duck. The goal is to survive for as long as possible. As shown in following figure.



As you see multiple objects (obstacles) moving towards dinosaur which includes cactus, clouds and birds flying over dinosaur at different heights decided randomly when score reach to 400. As shown in following figure.



Score is calculated according to the moving speed of objects towards dinosaur. As the score crosses 400 figure the speed also increases so the score as well. For more details about this game refer to this link https://www.youtube.com/watch?v=sB_IGstiWlc.

Your task is to implement this game according to the description given above in C++ by applying object oriented programming concepts. Moreover, for this you can also use C++ graphics library if you want to give a good looking interface to your game. Any group who used graphics library will get extra +2 points.

Marking Criteria:

Marking will be done based on the description given above but some of the marking aspects are given below.

1. Project Deliverable 01: In this phase you are required to submit at most 5 pages (at least 3 pages) report containing the following details
    - How you will implement this game and what are the classes you are going to implement in your C++ program. In this you will also give functions prototype that will be used in each class.
    - **UML Class diagrams** and interconnectivity based on Association (Aggregation, Composition and Inheritance) between the classes you are going to implement.
    - **Note:** Before submitting **Deliverable 01**, think deeply as many times as possible because the final implementation will be compared to your Project Deliverable 01 and if any deficiency of a class or a function found then marks will be deducted accordingly.

2. Project Deliverable 02: In this phase you will submit your full and final implementation of your project according to as explained in description. Evaluation of this phase will be performed as given below.
    - Full working and execution of the game as given in description
        - ♦ No. and types of objects given in description
        - ♦ Speed increase after score of 200. In game actually it is 400 but you need to increase the speed when score reached to 200
        - ♦ Appearance of birds after the score 200 at different heights decided randomly by your program.

- Appearance of Highest and Current score at upper right corner as given in figures above.
- Any graphics library used. (+2 points bonus)
- **Viva Voce**
- Comparison to Deliverable 01: Marks will be deducted in case if classes or functions deficiency found that is already discussed in Deliverable 01 but not implemented in final phase.

Project 02: Snake Game

Description:

A very famous game published by Nokia, for monochrome phones. Graphics consisted of black squares, and it had 4 directions. It was programmed in 1997 by Taneli Armanto, a design engineer in Nokia and introduced on the Nokia 6110.

In this project you are required to implement Five levels snake game in C++ programming using OOP concepts. In each level there are three entities as explained below.

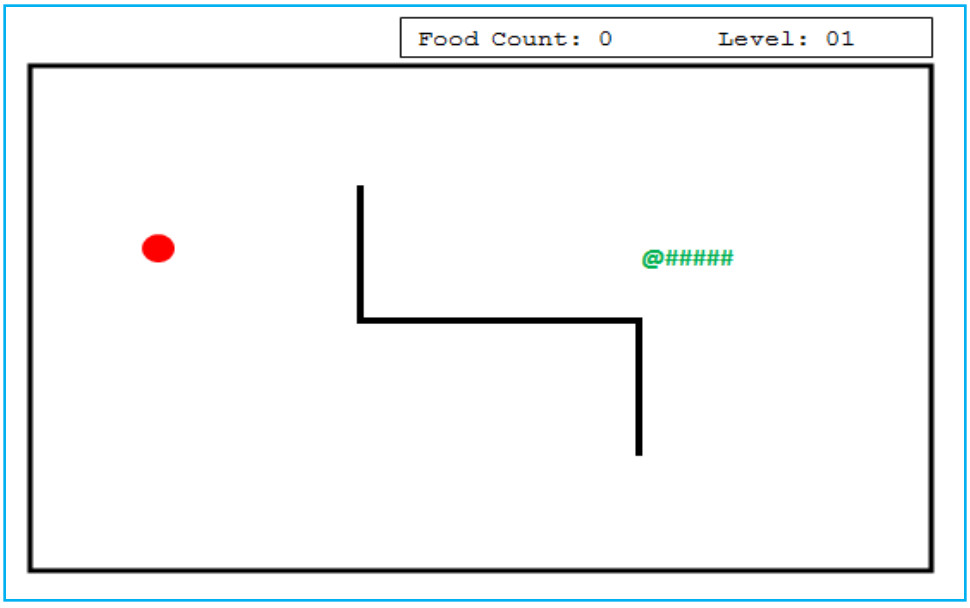
1. **Snake** initially represented as a combination of one @ and five "#" characters (i.e @#####). Snake move and grows continually as it eats the food.

2. **Frame or obstacles** are represented with black lines. When snake touches obstacles or its own body at most 3 times the game will terminate. You need to introduce here the concept of lives and set a limit of three lives. If snake touches obstacles or its body three times the game will terminate.

3. **Food** represented with a red round dot. You will direct the snake towards the food and when snake touches (eats) it the food will disappear and will appear randomly at another location within premises of the rectangle (game boundary) and the size of the snake increases by two hashes "#". When snake completes eating 10 food items the level will be upgraded to the next one and the food count will be reset to zero.

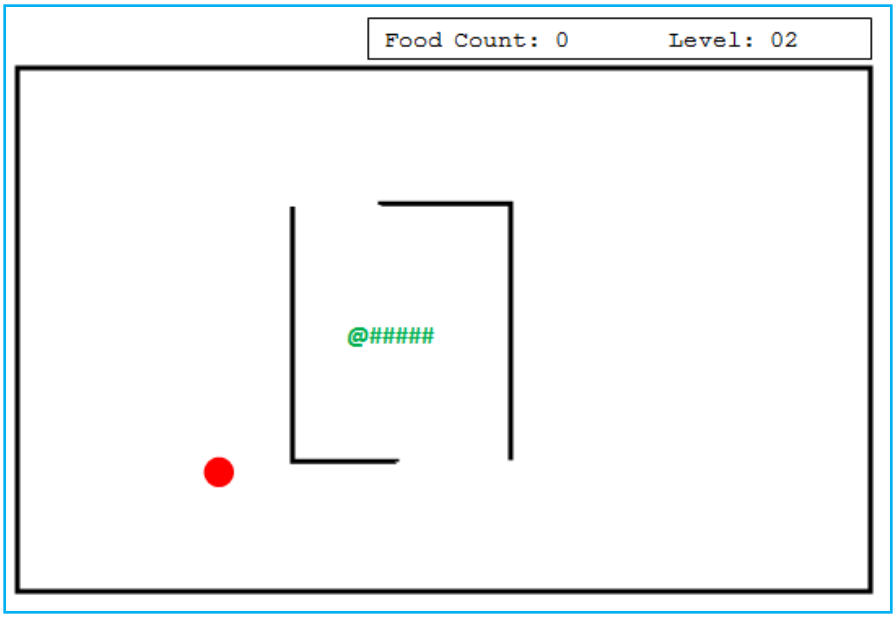
Levels of the Snake Game: In this game you will implement the following five levels.

Level-01

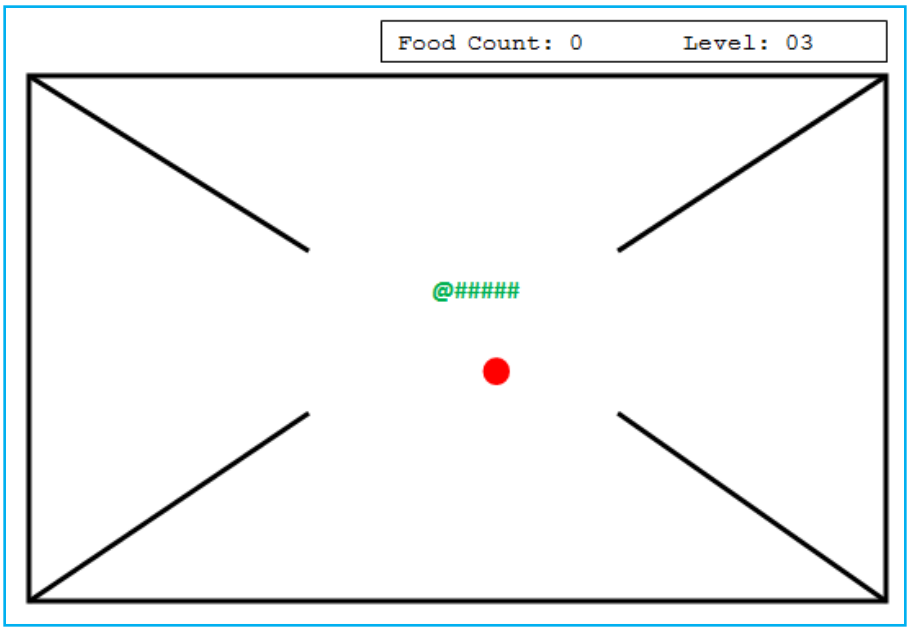


In level 1 the screen layout is shown. In upper right rectangle Food Count is shown which represent food items eaten by the snake and level No. The second rectangle is the area where the snake will move and you will try to keep it away from the obstacle and will direct it towards food. Screen shots of each Level given below.

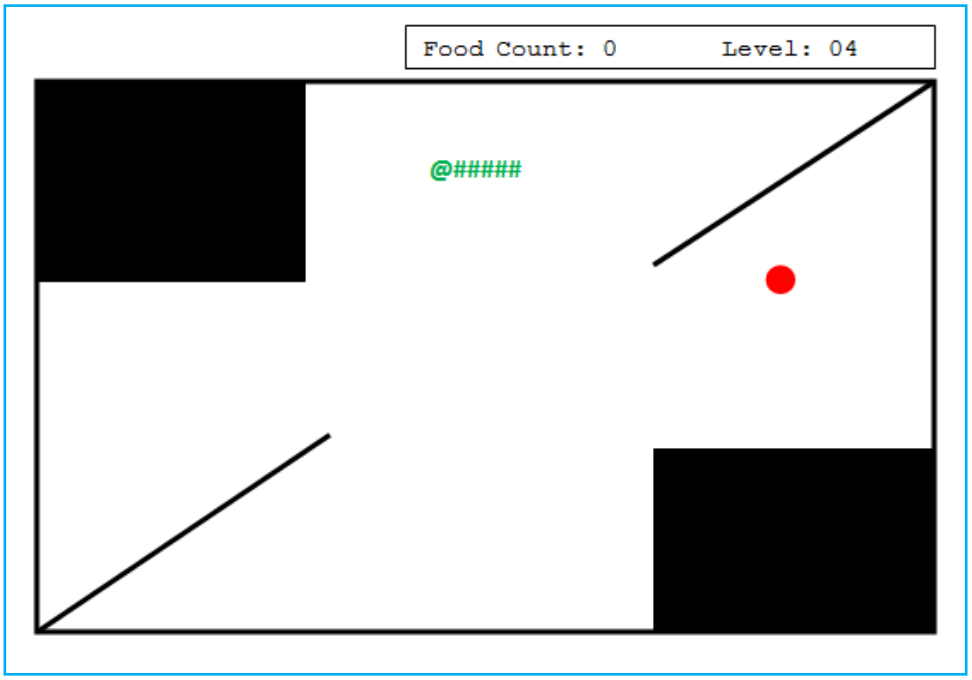
Level - 02



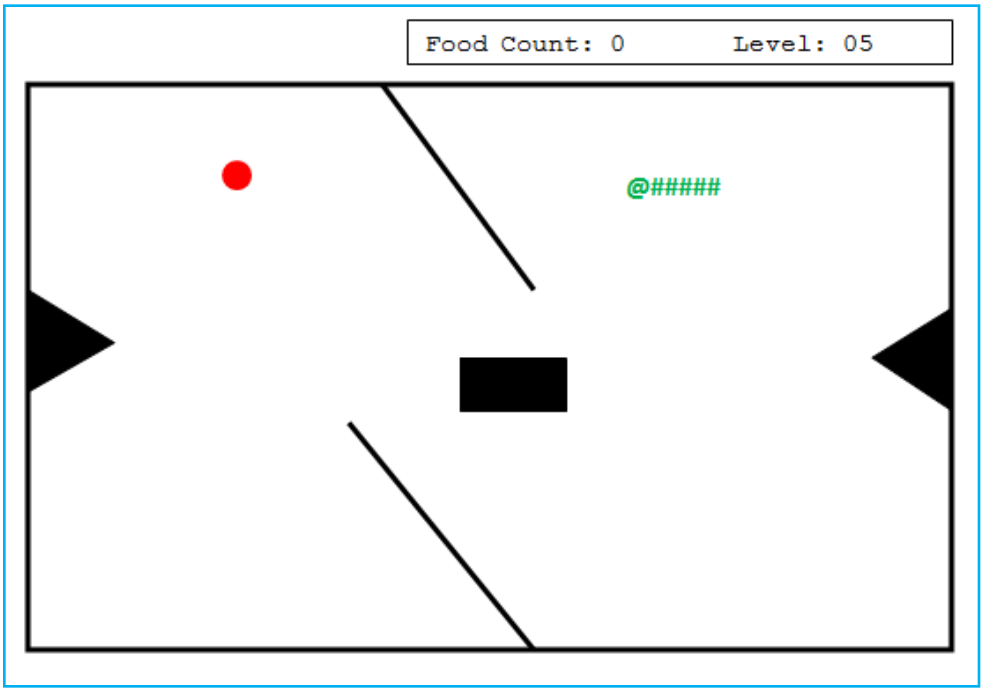
Level -03



Level-04



Level-05



Maximum No. of Groups Allowed: Nine groups each consisting two members at most. Project will be allocated to **Nine** groups based on first come first served bases.

Marking Criteria:

Marking will be done based on the description given above but some of the marking aspects are given below.

3. **Project Deliverable 01:** In this phase you are required to submit at most 5 pages (at least 3 pages) report containing the following details
    - How you will implement this game and what are the classes you are going to implement in your C++ program. In this you will also give functions prototype that will be used in each class.
    - **UML Class diagrams** and interconnectivity based on Association (Aggregation, Composition and Inheritance) between the classes you are going to implement.
    - **Note:** Before submitting **Deliverable 01**, think deeply as many times as possible because the final implementation will be compared to your Project Deliverable 01 and if any deficiency of a class or a function found then marks will be deducted accordingly

4. **Project Deliverable 02:** In this phase you will submit your full and final implementation of your project according to as explained in description. Evaluation of this phase will be performed as given below.
    - Full working and execution of the game as given in description
        - ♦ No. and types of objects given in description
        - ♦ Five levels of the game
        - ♦ Appearance of the Level No. and No. of Food items taken at upper right corner as given in figures above.
        - ♦ C++ graphics library used. (+2 points bonus)
        - ♦ **Viva Voce**
        - ♦ Comparison to Deliverable 01: Marks will be deducted in case if classes or functions deficiency found that is already discussed in Deliverable 01 but not implemented in final phase.