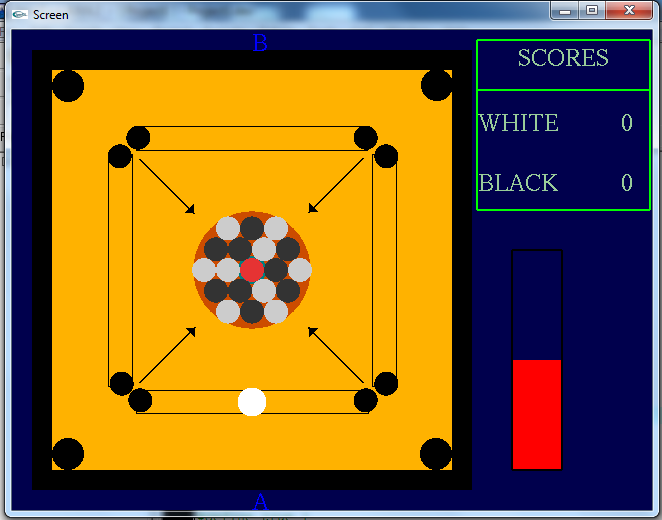
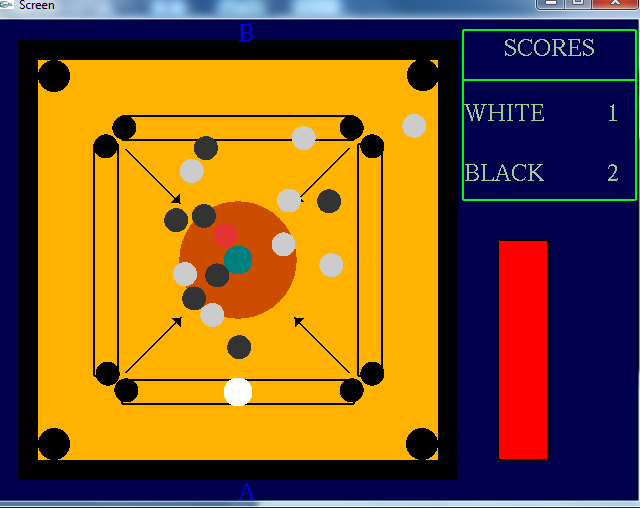
CAROM MANIA

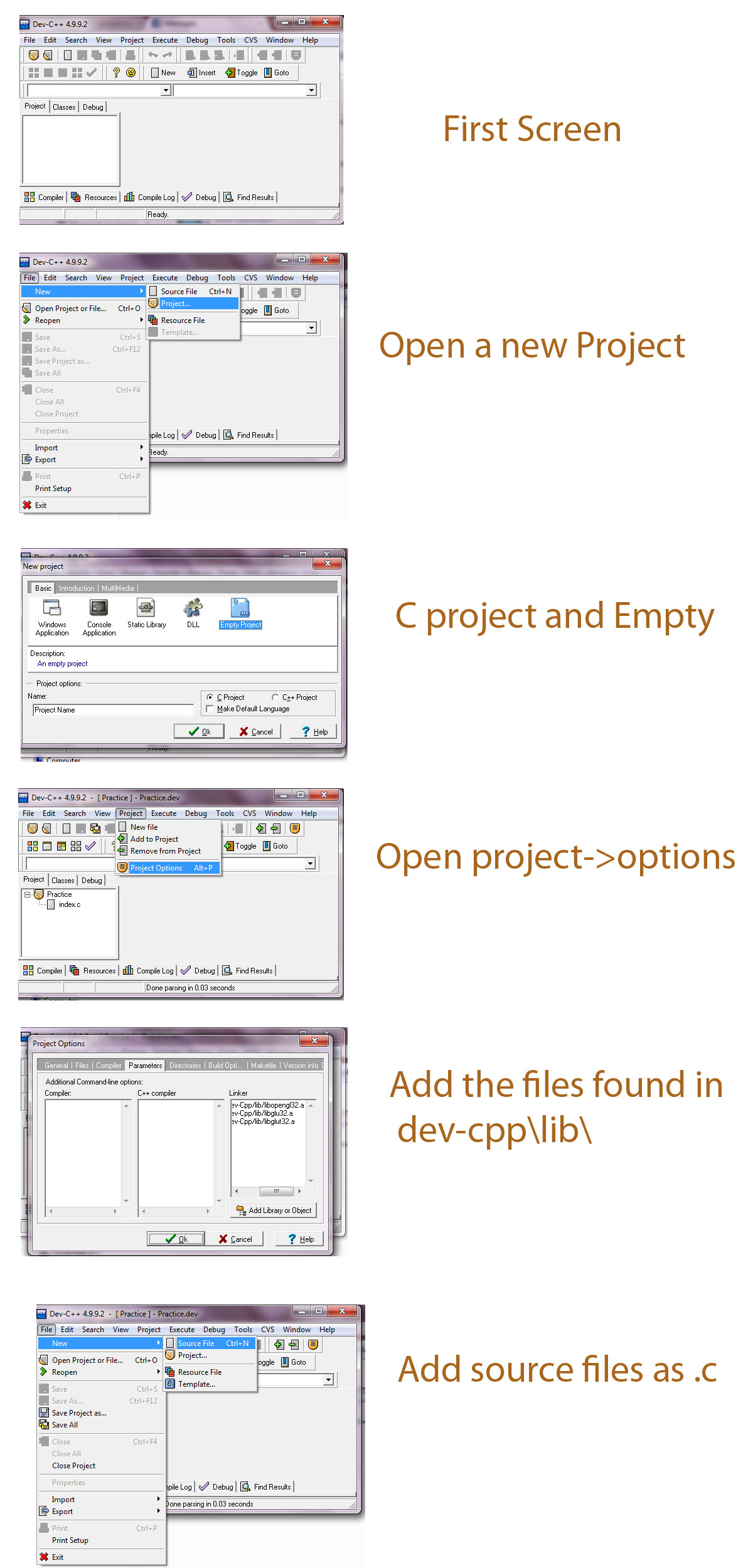
* INTRODUCTION:
  + This project aims at building the most commonly played game CARROM. It is a simple ‘Black and White’ 2 player game. We intend to make it a keyboard oriented game only where some specific keys would perform specific functions as defined in the program.
  + We are using OPEN GL for our graphics related portion.
* WORKING OF THE GAME:
  + The first screen is very user friendly. The keys itself tell the user to press ‘enter’ to play, ‘esc’ to exit and ‘H’ for the help menu.

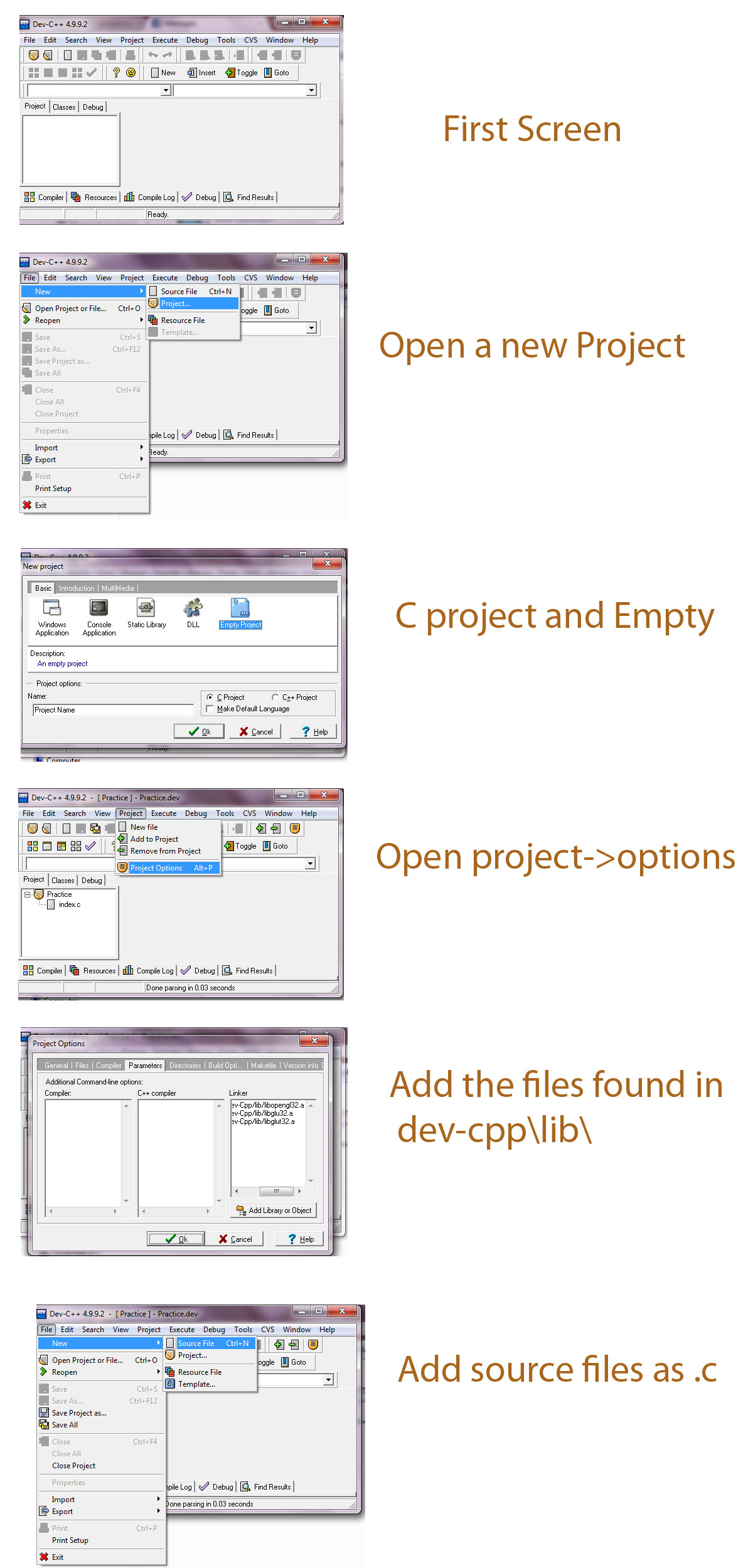


* + Once the play option is chosen, the above screen appears. First press of arrow keys sets the striker’s position. The second press sets the direction of striking. The third press sets the power level. This has to be done every time the user wants to strike.
  + Once the game begins and we set the striker in position and aim and shoot, collisions start.
  + For the second player’s turn, the board rotates and the next player strikes in the same way.
  + The score keeping is also being done side by side. Initially both the players’ scores are set to zero. There is an increment of 1 point for pocketing of each white or black coin and 5 points for the queen. The player who scores more wins.



* FUNCTIONS INVOLVED:
  + **void basedesign()**:
    - In this function the carom board was designed, along with the first screen and help window.An ‘if’ statement is incorporated to switch between the windows. If event == menu ,the first screen is shown to the user. If event == play, the carom board appears with coins and striker in place. The game starts with the pressing of arrow keys as explained above. The board , holes and the power bar are drawn using the commands glbegin(GL\_POLYGON) which initiates the drawing. glVertex2f(,) is used to tell the (x,y) co-ordinate. The order in which the points are written draws the line through those co-ordinates. glEnd() completes the process of drawing a rectangle, Square or circles.
  + **void printscreen()**:
    - By this function the text is printed. A string is passed to this function which prints the string character by character.
    - glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,str[ch]);is the command that is used for character printing. HELVETICA and TIMES NEW ROMAN are the two fonts that have been used. These font sizes that are pre-defined in OPEN GL.
  + **void myinit():**
    - This function creates initializes the window parameters.
  + **void initialize():**
    - This places the coins and striker in position.
  + **void draw\_board():**
    - This function calls the basedesign() function and also generates the coins. The positioning is done in a separate function.
  + **void striker\_set():**
    - Positions the striker and controls its movement on the board with the help of ‘up arrow key’. It also facilitates jumping of striker whenever a coin comes in between.
  + **void player\_change():**
    - It changes the player according to the rules.
  + **void foul():**
    - As the name suggests it controls the foul related activities. Whenever the striker is pocketed, it is a fowl. A point is deducted whoever makes the fowl.
  + **void hole():**
    - Funtion fowl() is called in hole(). It makes the coin disappear whenever more than half of the coin goes into the hole.
  + **void collision\_update():**
    - This function governs coin-coin collision and coin striker collision.
  + **void keypress():**
    - On pressing the arrow keys, the striker is set. Second press sets the direction and third press strikes.
    - Other keyboard related activities like pressing enter and escape are also governed by this function.
* SCOPE OF THE GAME:
  + This project can further be advanced to a 4 player game.
  + Or a computer versus player game where artificial intelligence would also be manipulated.
  + The collisions can be further modified. The mass less collisions can be converted into collisions where mass is taken into account with in-elastic collisions.
  + The mass of striker and coins could vary so that collisions are smoother.
  + The animations can be worked upon so that when the player changes , the board visibly rotates.
* HOW TO COMPILE:
  + Open Carrom.dev in Dev-Cpp or follow the following steps.
  + Download DevCpp from http://prdownloads.sourceforge.net/dev-cpp/devcpp-4.9.9.2\_setup.exe
  + Install the program with default settings
  + Extract the files
  + Copy **glut32.dll** to somewhere in your PATH like **c:\windows\system**. on 32 bit system  
    The files in the**include** and **lib** directories should be moved to **c:\devcpp**





* CONCLUSION:
  + We as a team learnt a lot from this project. We learnt a lot of programming, debugging and logic development through this task. Working with deadlines was a good experience for all of us. We know that it will prove helpful in future. Working in a team was another experience in its own way.
  + Once the project was completed and our game worked… the feeling is beyond words!!
  + The project was a wonderful experience for each one of us.