

# PIPE IT

Q1) The game to be designed has the following statement. The players who ever play the game must be given the following scenario:

“BITS-Pilani has dug wells at many places in and around Pilani. Now they are planning to interconnect those wells. They need your expertise so as to connect these wells, in the best way possible ie utilising the pipes available to the maximum extent and connecting the wells with least number of pipes. Also due to some constraints BITS-Pilani cannot buy the set of pipes required all at a time. There is only one shape of pipe available at a particular time and the next available shape is visible. These pipes should be placed in the grid given such that a connection is formed between the two wells. One pipe can be placed in one box of the grid. Hence the placement of the available pipe is crucial so as to use all pipes efficiently. If a pipe that is laid is removed you have to pay some extra cost which gets you a few negative points. As BITS-Pilani is under time constraints, you will have limited time to join the two wells.”

NOTE: Putting another pipe on already filled grid (replacing the pipe) costs a penalty.

NOTE: The immediate next shape of pipe that is going to be available should be made known to user.

Also provide how points will be generated. Time taken to complete the game, number of pipes used and number of overwriting of pipes are a few factors that are to be taken into consideration though a few other relevant factors are welcome.

The game should contain multiple levels (minimum 3) with increasing difficulty.

The game statement and points scheme must be clearly mentioned to the user.

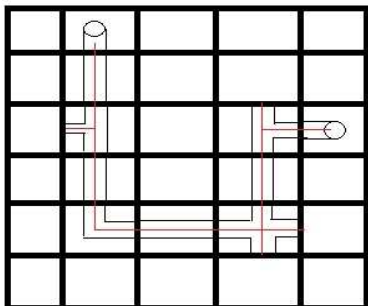
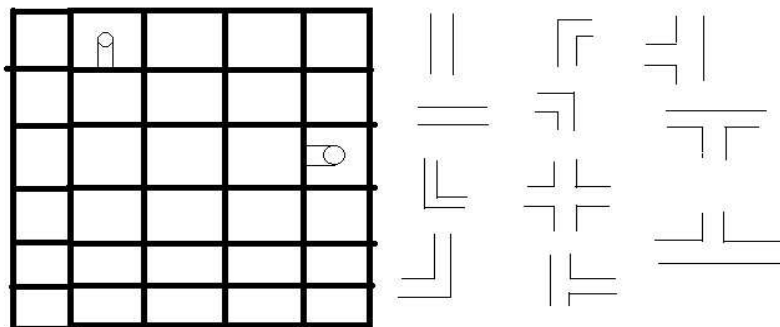
The top 10 scores, along with the user name, must be displayed at the end of every game.

The minimum number of components for the game are : timer, grid, pipes, mechanism to place the current pipe at a particular grid, score calculation and display, display of top 10 scores

The marking scheme for this question goes as follows:

1. Development of GUI or the interface for game-play (including the shapes of pipes and timer) -> 20%
2. Development of various symbols such as pipes that appear in the game -> 20%
3. Development of the module that checks for the connectivity of pipes -> 5%
4. Algorithm for different difficulty levels-> 10%
5. Storing and displaying the top 10 scores -> 5%
6. Efficiency of the code -> 10%
7. Good programming style (proper comments and indentation) -> 5%
8. Overall completeness of the game (compilation and execution) -> 25%

- The figure given below is a sample interface.
- The grid can be of any size depending on the difficulty but not lesser than 5x5.
- The start point and endpoint must be specified. (The two wells in the game).
- Also given are the pipe shapes that are to be utilised(One pipe fits in one box only)



The sample game is shown in the diagram. Though this is not optimal solution, it connects the two wells. The lines in red represent the connectivity (flow) between the wells.