Developers:
Dhruv Raj Bangad – bangaddj@mail.uc.ed | Nishil Faldu – faldund@mail.uc.edu

# THE GAME PLAN

**Tesla: "save" Elon Musk's prized Roadster from being sent into space.**

1. The conceptual layout of the 15x15 room
2. Display of the 15x15 room
3. Display of the game status to the user
4. User interaction with the game
5. Tracking the number of times, the user came across Yugo and Pinto
6. Tracking if the roadster has been found or not
7. Finding the users way into SpaceX (15x15 room) (Randomly occupied location)
8. Hideout of the roadster (Randomly generated location)
9. Hideout of the Yugo (Randomly generated location)
10. Hideout of the Pinto (Randomly generated location)
11. Hideout of sleeping Elon Musk (Randomly generated location)
12. How Elon Musk wakes up: No sooner the user detonates a land mine, Elon Musk wakes up
13. Random movement of Elon Musk through the building looking for the user
14. Getting away from Elon Musk, if he lands near the user (ONLY ONCE)
15. Getting caught if the user gets into either Yugo or Pinto
16. Display game information:
    a. Board Graphic
    b. Descriptive Menu
    c. A combination of both will be employed in the Tesla Game
17. Showing the location of the cars and Elon Musk only when they are in the immediate vicinity (within the 3 by 3 grid area with the user at the center).
18. The movement of the user: w (forward/north), a (left/west), s (backward/south), d (right/east) for movement directions.
19. The user can quit any time (when the user presses q twice)
20. Guiding the user to the proper location of the roadster
21. Asking the user to play again once the game is over
22. If the user chooses to play again, tracking statistics about the number of turns it took to find the roadster, the number of times the roadster was found, and the number of times the user became "Starman"

Developers:
Dhruv Raj Bangad – bangaddj@mail.uc.ed | Nishil Faldu – faldund@mail.uc.edu

# LAYOUT OF THE GAME

## ADDITIONAL FEATURE TO THE GAME:

We have decided on including an extra feature in the game, which will help us to wake Elon Musk and control his movements. We have decided on adding 20 mines in the game. Each mine will occupy a single box, whose location will be unknown to the user. If he steps on it while musk is sleeping, he will wake up and head toward the direction the mine bursts, if the user steps when Musk is awake Musk will proceed in the direction where the mine bursts. This does no harm to the user. A mine can burst multiple time.

### 1. The conceptual layout of a 15x15 room

We will incorporate a "Board" object. Since the room is two dimensional, we will be dealing with a two-dimensional array. As we have an array, we will also be dealing with a lot of two-dimensional coordinates assigned to various points. We plan to divide the 15x15 room in 225 boxes (which means that we will have an array with 225 boxes and the indexes of each box will serve as the two-dimensional coordinates of those boxes. This two-dimensional coordinate system will help us assign locations to different objects. These locations will, of course, be generated randomly. In conclusion, the Board class/object will represent the playing board (room of SpaceX) itself. It will help us keep a track of which objects are in which positions at a given movement. For now, we plan on using ASCII characters 219 and 255 which represent a black and white box, respectively. This layout will be like a chess board.

### 2. Display of Game Status to the user and User Interaction with the Game

The user/player will be able to know the status of the game by knowing his/her position on the board. The movement of the user is by the following keys: w (forward/north), a (left/west), s (backward/south), d (right/east) for movement directions. The user can only move a box at a time. The position of the player will be highlighted by the initials of the players name. Also, the location of the cars, or Elon Musk will not be displayed unless they are in the immediate vicinity of the player (within 3 by 3 grid area with the user at the center). When the location of the cars, or Elon Musk is displayed it will in the form of their initials. The game status will also be displayed in the form of a Descriptive Menu, when necessary. Major events like detonating a land mine, waking up of Elon Musk due to the detonation, and the player becoming a "Starman" will be displayed on the screen of the user in the form of comments. Along with the game status, hints and clues will also

be displayed on the screen to help the user find his/her way to Tesla Roadster.

3. **Tracking the number of times, the player came across Yugo and Pinto and MUSK**

We plan on displaying a separate counter which will be incremented every time the player is in the vicinity (within 3 by 3 grid area with the user at the center) of either of the cars or Elon Musk.

4. **Tracking if the roadster has been found or not**

If the user's coordinates are the same as that of the tesla roadster, the car will be found, and the user will "save" Elon Musk's prized Roadster from being sent into the space.

The header files are the building blocks of the Tesla Game Project. Each of them has been designed to perform certain specific task in game controlling. You can find the detailed description about the header files here. The header files which we plan on creating are listed as follows:

1. **Board Representation (gameArena.h)**
   The game board will be generated (like a chessboard), which can be used to represent the randomly generated locations of the user, Elon Musk, the cars, and the mines.

2. **Basic Movements of Elon Musk (muskMove.h)**
   This header file and class will be responsible to add functionality for the basic movements of Elon Musk – how Elon moves randomly about the room.

3. **Basic Movements of the User/Player (starmanMove.h)**
   This header file and class will be responsible to add functionality for the basic movements of the Player – how the Player moves about the room with the controls mentioned above in the description.

4. **Functioning of the Game (clues.h)**

   Adds the functionality to play the complete game of Tesla, provides clues to the user, display of the game in the form of a Descriptive Menu.