# 06 PREPARE : CHECKPOINT B - MOON LANDER DESIGN

1. Problem Description

   Tish program will allow e user to simulate moon land, by accepting inputs from the user to move the spaceship and control their action until it safely land on the platform.

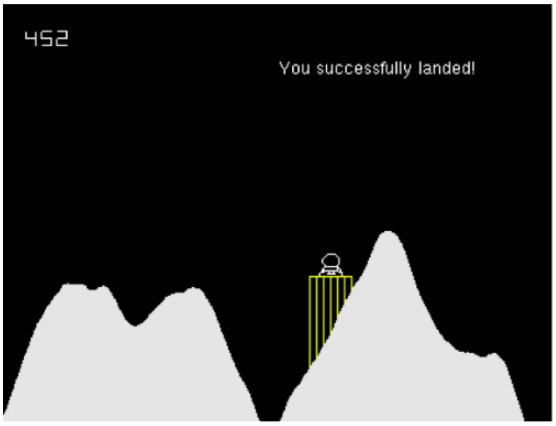
2. Design Overview

The following describes the rules and game play of Moon Lander:

- The dimensions of the screen are: (-200, -200) to (200, 200).
- The lander begins with 500 units of fuel.
- To land successfully, the lander must:
  - have its center within the horizontal boundaries of the platform.
  - be within 4 pixels vertically, above the platform.
  - arrive at the platform with a velocity of no more than 3 pixels per frame in any direction.
- Gravity on the moon can be modeled as 0.1 pixels per frame.
- The left arrow key causes thrust on the left of the lander which propels it to the right (and similar for the right and down arrows).
- The left and right thrust amounts are 0.1 pixels per frame, and consume 1 unit of fuel.
- The upward thrust (caused by the down arrow) amount is 0.3 pixels per frame, and consumes 3 units of fuel.
- The lander should have inertia, in other words, once it begins moving left, it should continue moving left unless additional thrust is made.
- The lander should not continue to move after crashing or landing.
- After successfully landing, the game should display, "You have successfully landed!"
- After crashing, the game should display, "You have crashed."
- After running out of fuel, the lander should not be able to apply thrusters (but can continue falling).
- Any other contact with the ground or platform results in a crash.

3. Interface Design

   First, you need to set up game environment for OpenGL projects.




Some Inputs:

KEY DOWN CALLBACK

When a key on the keyboard has been pressed, we need to pass that on to the client. Currently, we are only registering the arrow keys

INPUT key: the key we pressed according to the GLUT_KEY_ prefix

x y: the position in the window, which we ignore

KEY UP CALLBACK

When the user has released the key, we need to reset the pressed flag

INPUT key: the key we pressed according to the GLUT_KEY_ prefix

x y: the position in the window, which we ignore

INTERFACE : KEY EVENT

Either set the up or down event for a given key

INPUT key which key is pressed

fDown down or brown

## 4. Algorithms

```
BOOL trust()
    IF (alive() && landed() && getFuel())
        RETURN
VOID graviy()
    verticalGravity()
VOID trustLeft()
    IF trust()
        horizontalTrust()
        SET fuel

VOID trustRight()
    IF trust()
        horizontalTrust()
        SET fuel

VOID trustBottom()
    IF trust()
        verticalTrust()
        SET fuel

VOID horizontalTrust
    SET velocity
```

VOID verticalTrust

SET velocity

SET point

END

5. Data-structures

The lander structure is divided in three members variable:

Horizontal

Vertical

Bottom

6. File Format

The graphics and game play will be done with OpenGL.

7. Error Handling

User wrong/invalid input:

Re-prompt user with IF statement

File errors/corrupted:

Re-prompt the user with IF statement

Internal/code errors:

Use asserts library and/or others debuggers