Moon Lander Design

Program Description

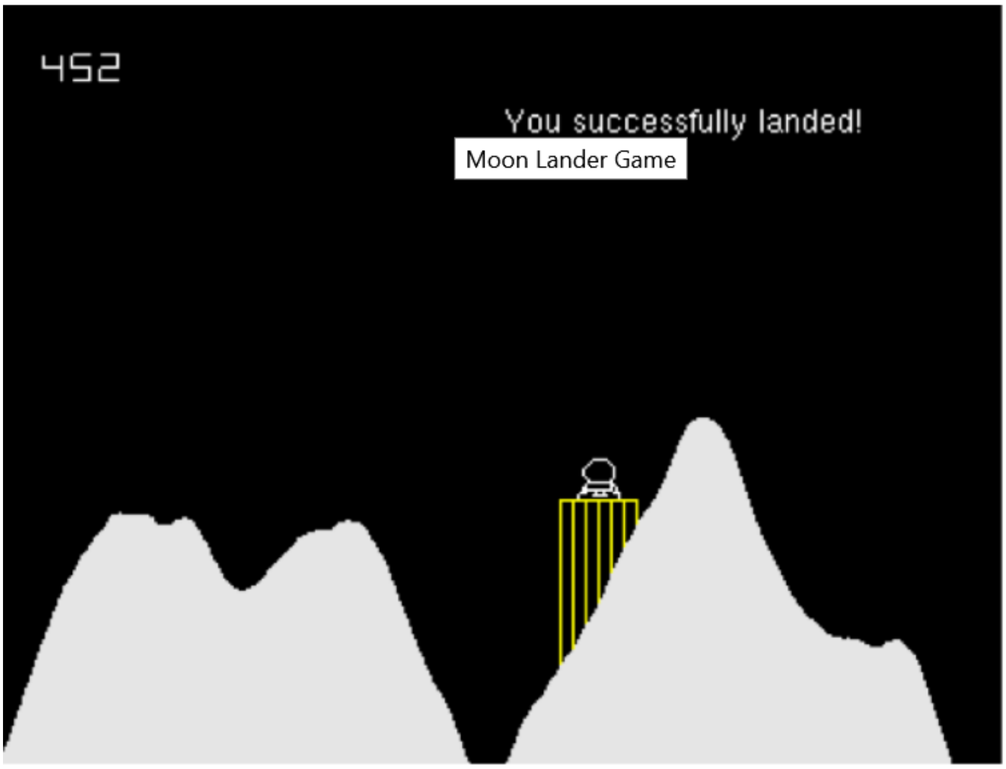
The program will allow one user to navigate the moon lander using thrusters to land the moon lander safely on the designated ground platform.

Design Overview

The game will consist of a random terrain design and a positioned ground platform for the moon lander to land on. The moon lander itself will be designed and movement of the moon lander will be by the user directions. It will have a number of conditions that the moon lander will have to comply with whilst under movement in trying to create a realistic moon landing event. The environment will be another point as we maintain a consistent gravity on the moon and keep physical inertia due to lack of friction in the vacuum of space. Also providing a text box to provide information to the user on a successful landing or not.

Interface Design

Output Screen Shot



452

You successfully landed!

Input

Pilot the lander

→ Right arrow key causes thrust on the right forcing the lander to the left.

← Left arrow key causes thrust on the left forcing the lander to the right.

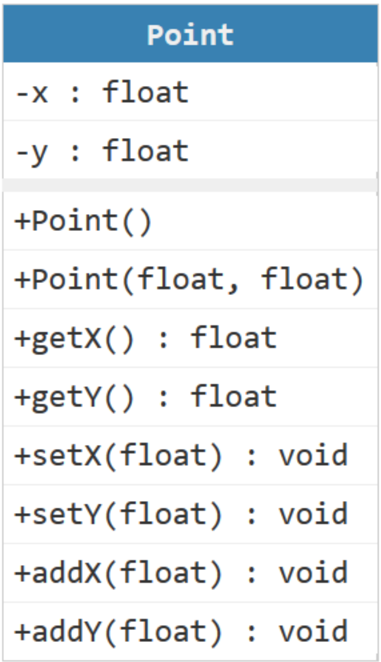
↑ Up arrow key causes thrust downwards forcing the lander to the ground.

↓ Down arrow key causes thrust upwards forcing the lander into the atmosphere.

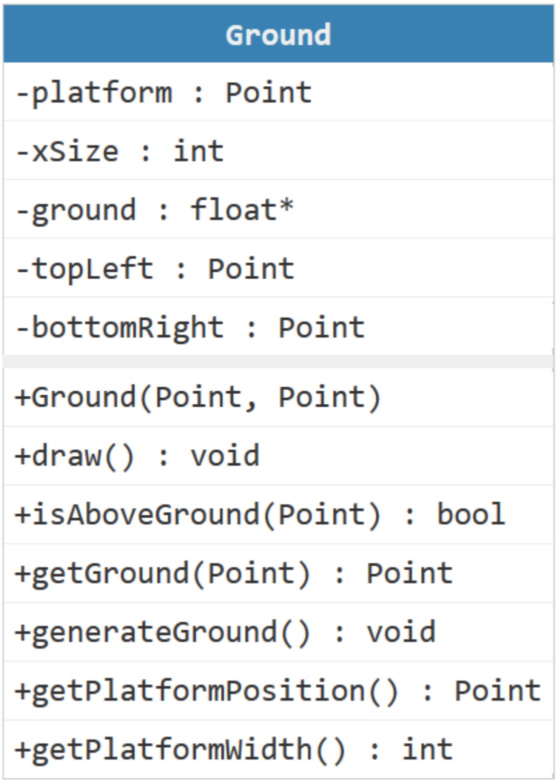
Algorithms

```
PROMPT for user Input
GET user input
IF (lander(isLeft))
    ThrustLeft();
ELSE IF (lander(isRight))
    ThrustRight();
ELSE IF (lander(isUp))
    ThrustUp();
ELSE IF (lander(isDown))
    ThrustDown();
drawLanderFlames(Point p, bool bottom, bool left, bool right)
bool isLanderalive(bool bottom)
drawLander(Point p)
```

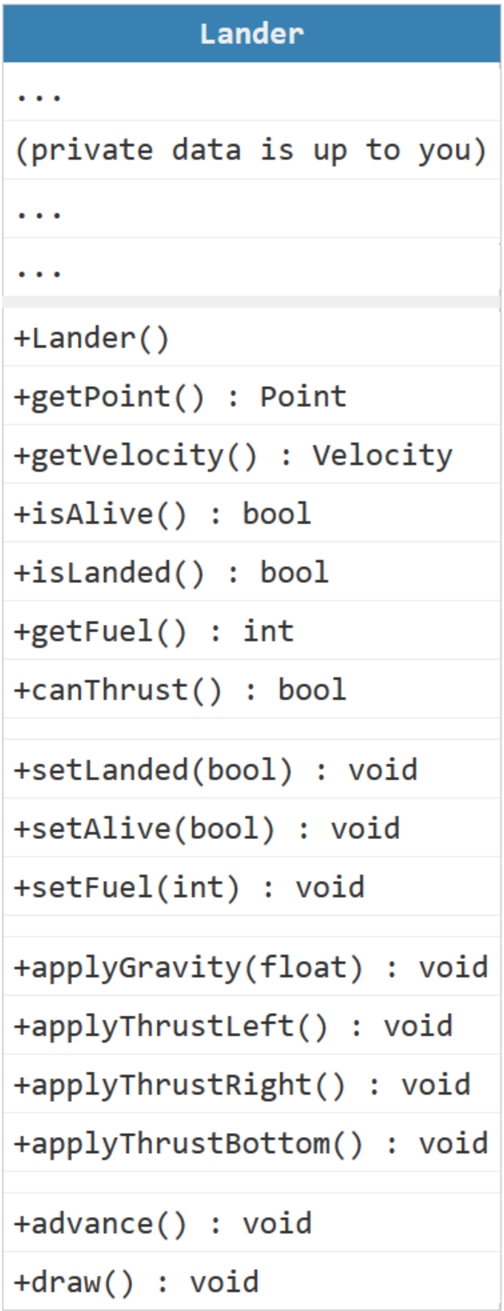
Data Structures



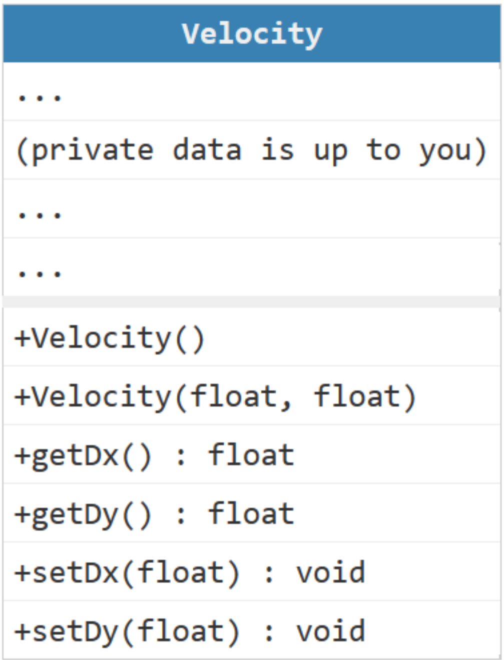
Point
-x : float
-y : float
+Point()
+Point(float, float)
+getX() : float
+getY() : float
+setX(float) : void
+setY(float) : void
+addX(float) : void
+addY(float) : void



Ground
-platform : Point
-xSize : int
-ground : float*
-topLeft : Point
-bottomRight : Point
+Ground(Point, Point)
+draw() : void
+isAboveGround(Point) : bool
+getGround(Point) : Point
+generateGround() : void
+getPlatformPosition() : Point
+getPlatformWidth() : int



Lander
...
(private data is up to you)
...
...
+Lander()
+getPoint() : Point
+getVelocity() : Velocity
+isAlive() : bool
+isLanded() : bool
+getFuel() : int
+canThrust() : bool
+setLanded(bool) : void
+setAlive(bool) : void
+setFuel(int) : void
+applyGravity(float) : void
+applyThrustLeft() : void
+applyThrustRight() : void
+applyThrustBottom() : void
+advance() : void
+draw() : void



Velocity
...
(private data is up to you)
...
...
+Velocity()
+Velocity(float, float)
+getDx() : float
+getDy() : float
+setDx(float) : void
+setDy(float) : void

File Format:

Not applicable

Error Handling:

If prompt not appropriate response use error handling cin.fail(), cin.ignore() and cin.clear() using a loop to ensure an appropriate response.