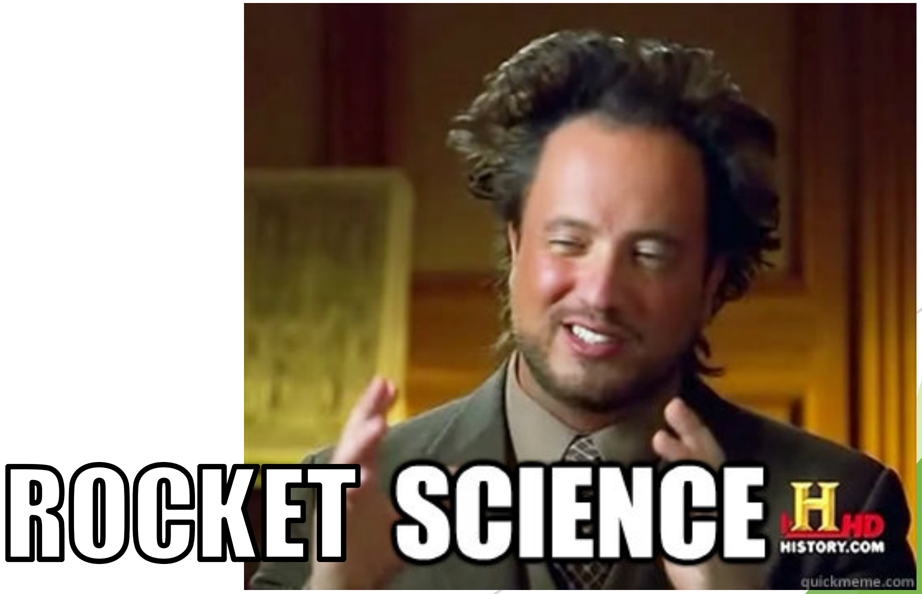
Modelling Physical Systems



By:
Anders Nørgaard
Alexandros Panagiotis Giakalis
Ulrik Kroge Sloth

Agenda

- Physics of a Rocket
- Implementation in Unity3D
- Stabilization and issues

Physics of a Rocket

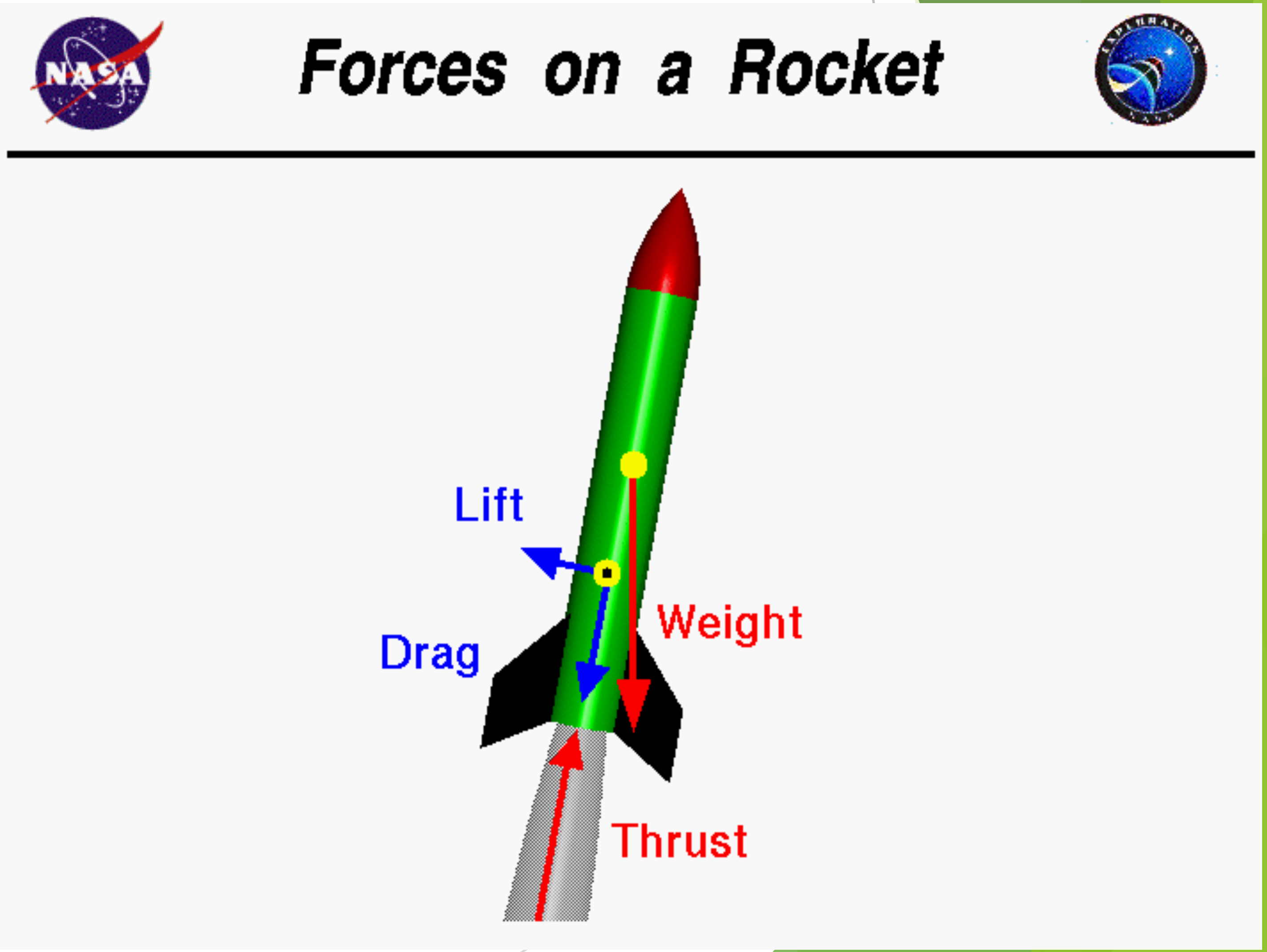
- Earths Gravitational Field
  - g ≈ 9.81 m/s^2

$$F_g = m * g$$

$$F_g = 500kg * 9.81m/s^2$$

$$F_g = 4905\frac{kg * m}{s^2} = 4905N$$

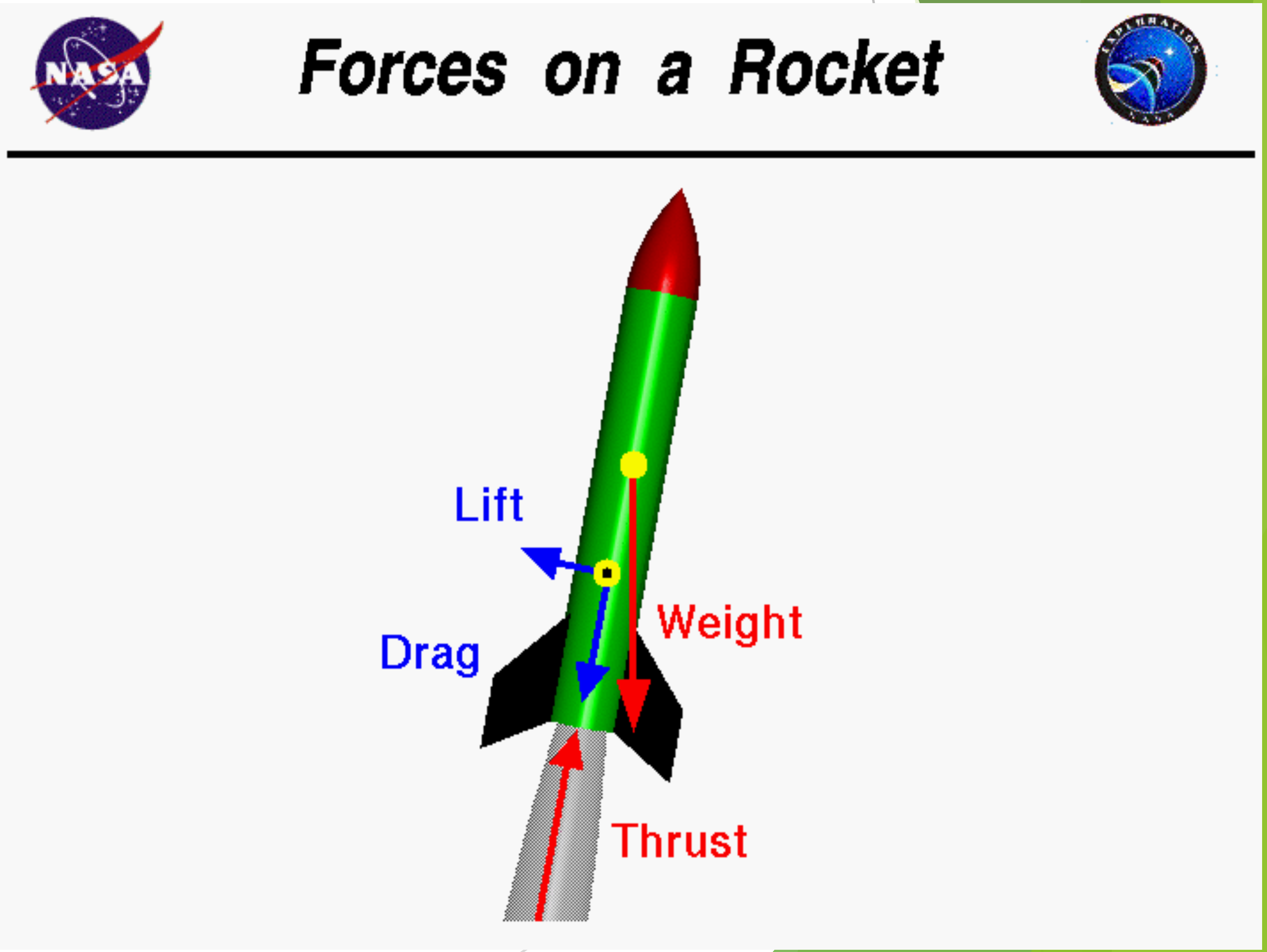
- Pull, wont move



Physics of a Rocket

- Thrust
  - Measured in N
  - Must be larger than F to accelerate
  - Thrust to propulsive power

    $P = Tv$
  - As fuel is used by thrust, weight decrease

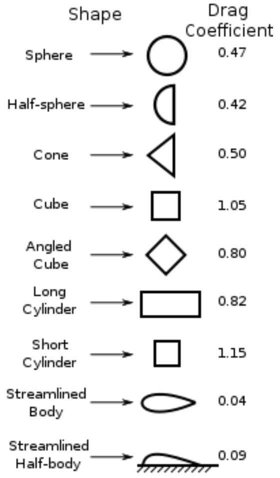


Physics of a Rocket

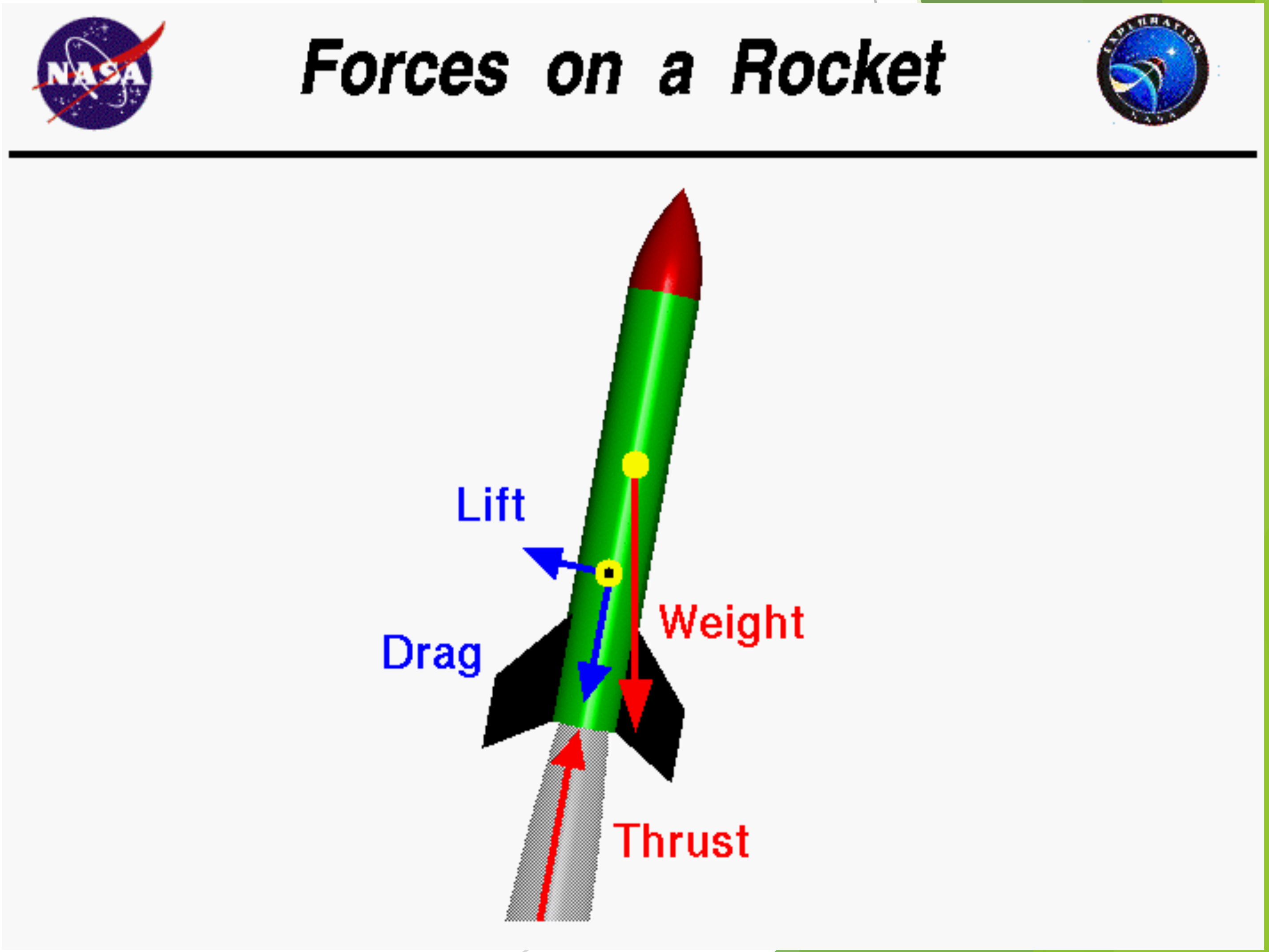
- Drag
  - Air resistance
  - Measured in N

$$F_d = \frac{1}{2} p v^2 C_d A$$

  - Drag coefficient Cd for different shapes



Measured Drag Coefficients



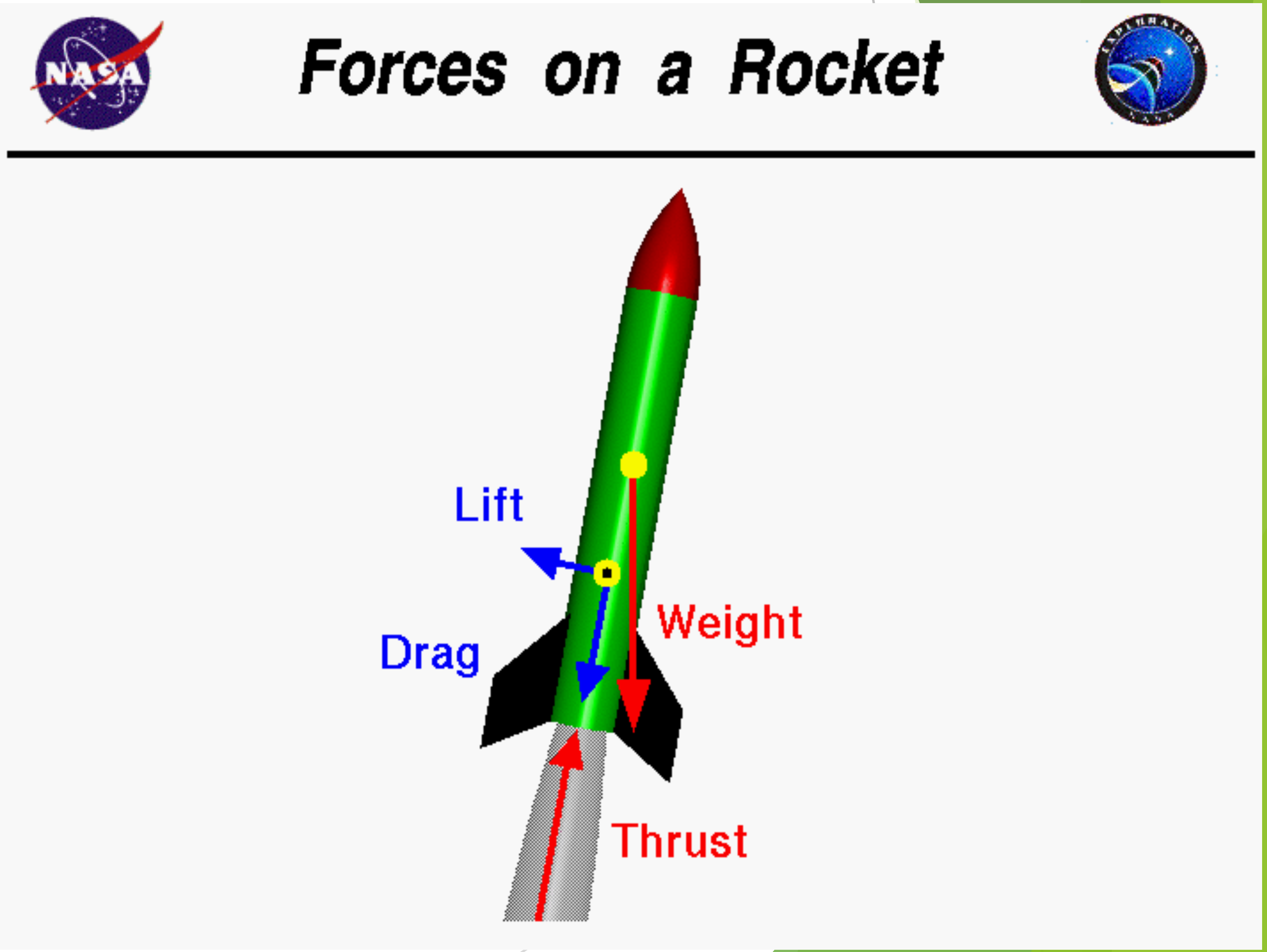
Physics of a Rocket

- Lift
  - Used on rockets for stabilization and direction control

    $$L = 0.5 * dv^2 sCL$$

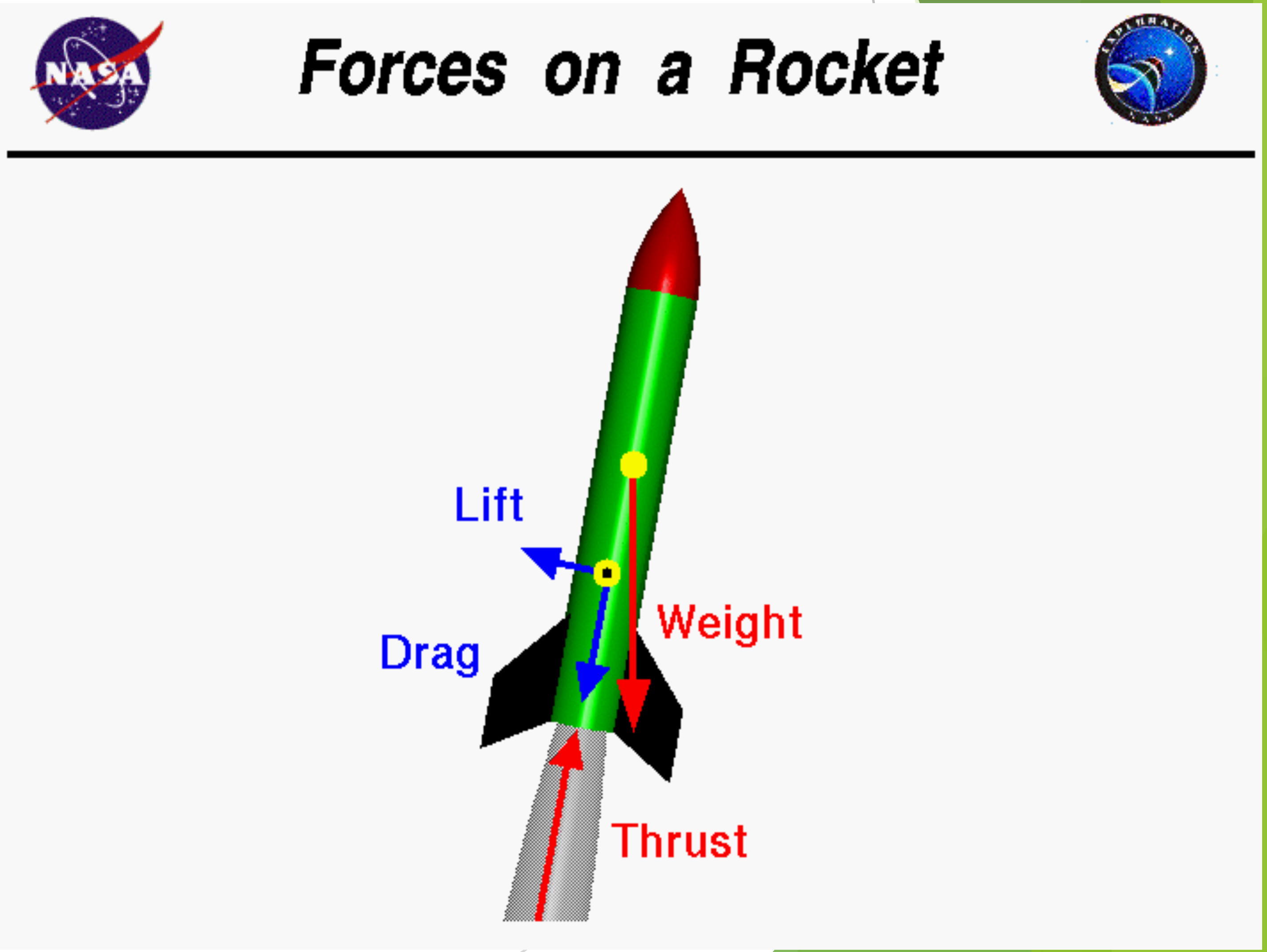
  - We use the Mach number as CL

    $$M = \frac{u}{c}$$



Physics of a Rocket

- Center of Pressure
  - All aerodynamic forces act through this point
  - Normal calculation
  - Our estimation



Implementation in Unity

- Gravity pulls down the rocket.
- Used rigidbody for the mass of the rocket and also for applying the gravity force directly to the center of gravity of the rocket.
- Used Earth’s gravity $g = 9.81 m/s^2$

```
using UnityEngine;
using System.Collections;

public class WeightForce : MonoBehaviour {

    Vector3 gravityDirection;
    Vector3 gravityForce;
    public float gravityAcceleration;

    Rigidbody rb;

    // Use this for initialization
    void Start () {


        rb = GetComponent<Rigidbody>();
    }


    void FixedUpdate()
    {
        gravityDirection = Vector3.down * gravityAcceleration;
        gravityForce = rb.mass * gravityDirection;

        rb.AddForce(gravityForce);

    }
}
```

Implementation in Unity

- Thrust is applied in order to force the rocket to take off and fly into the space.
- The direction of the thrust is along the longitudinal axis of the rocket through the center of gravity.
- 60% of the mass is defined as fuel and fuel consumption is used.

```
public class ThrustForce : MonoBehaviour {

    public enum ThrustState
    {
        Off, On
    }

    public float fuelAmount;
    public float fuelConsumption = 1.0f;
    float thrust;
    public float maxThrust;

    Vector3 thrustDirection = Vector3.zero;

    Rigidbody rb;
    public ParticleSystem fireParticle;

    public Image thrustAmount;

    public ThrustState currentThrustState = ThrustState.Off;

    public Transform centerOfThrust;


    void Start () {
        rb = GetComponentInParent<Rigidbody>();

        fuelAmount = rb.mass * 0.6f;
    }
```

Implementation in Unity

- Thrust is applied in order to force the rocket to take off and fly into the space.
- The direction of the thrust is along the longitudinal axis of the rocket through the center of gravity.
- 60% of the mass is defined as fuel and fuel consumption is used.

```
void FixedUpdate()
{
    switch (currentThrustState)
    {
        case ThrustState.Off:

            break;
        case ThrustState.On:
            if (fuelAmount > 0.0f)
            {
                float usedFuel = fuelConsumption * Time.fixedDeltaTime * thrust / maxThrust;

                fuelAmount -= usedFuel;
                rb.mass -= usedFuel;

                thrustDirection = transform.up  * thrust;

                rb.AddForceAtPosition(thrustDirection, centerOfThrust.position, ForceMode.Force);
            }
            else
            {
                fireParticle.Stop();
                currentThrustState = ThrustState.Off;
            }

            break;
    }
}
}
```

Implementation in Unity

- Drag force is opposite to the motion.
- Use of linear drag with a coefficient of 0.1 as an approximation.
- Lift force is generated when the rocket is inclined from the flight path.
- Add both force to the center of pressure.

```
public class AirForces : MonoBehaviour {


    public Transform centerOfPressure;
    //We use air density at zero degrees celcius
    float airDensity = 1.2922f;
    public Transform wingObj;

    Rigidbody rb;

    CapsuleCollider noseCol;

    void Start () {
        rb = GetComponent<Rigidbody>();
        noseCol = GetComponent<CapsuleCollider>();
    }
```

Implementation in Unity

```
void FixedUpdate()
{
    Vector3 lift;
    Vector3 drag;

    float angle;

    float noseArea = Mathf.PI * Mathf.Pow(noseCol.radius, 2);
    float totalWingArea = 0.0f;


    foreach (Transform child in wingObj)
    {
        BoxCollider wingCol = child.GetComponent<BoxCollider>();

        float smallWingArea = (wingCol.bounds.extents.y*wingCol.bounds.extents.z)*2;
        float largeWingArea = (wingCol.bounds.extents.x * wingCol.bounds.extents.z)*2;

        angle = Vector3.Angle(transform.up, child.up);

        float visibleWingArea = Mathf.Lerp(largeWingArea, smallWingArea, Mathf.Sin(angle));

        totalWingArea += visibleWingArea;
    }

    float totalArea = noseArea + totalWingArea;

    Vector3 machNumber = -rb.velocity / 343.2f;

    lift = (0.5f * airDensity * rb.velocity.sqrMagnitude * totalArea * machNumber);
    drag = -0.1f * rb.velocity;
    //areas of wings found by assuming the top part of the lerp(2, area of wing, sin(angle))
    rb.AddForceAtPosition((drag + lift), centerOfPressure.position);
}
```

Stabilization

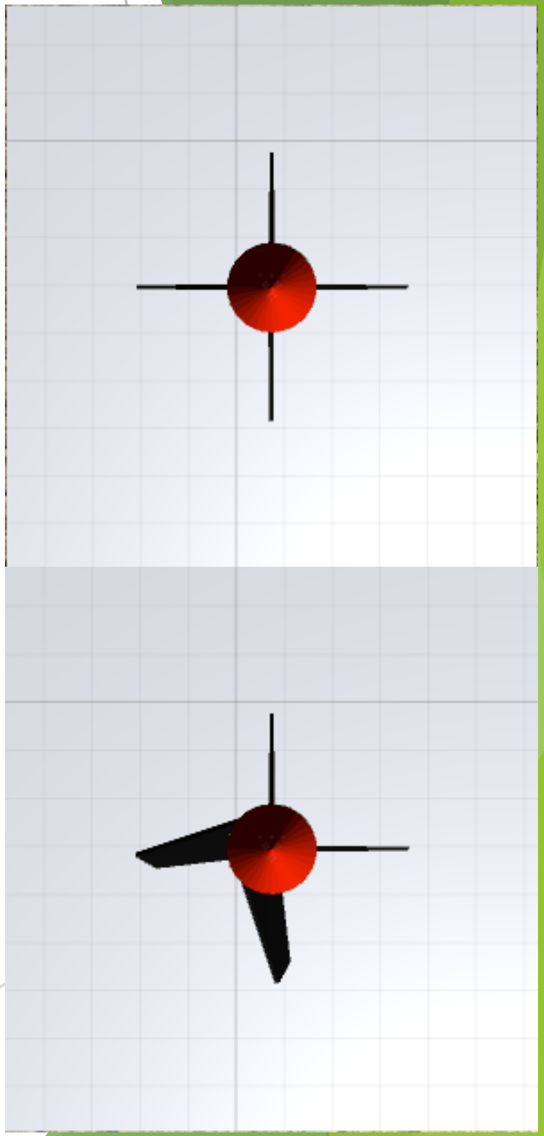
- Small windgusts and thrust instabilities will affect a rocket.

- The result of this:
  - The rocket will slowly go off path, and start curving.

- We want to avoid this, but how can you do it?
  - Using Aerodynamics
  - Using Thrust

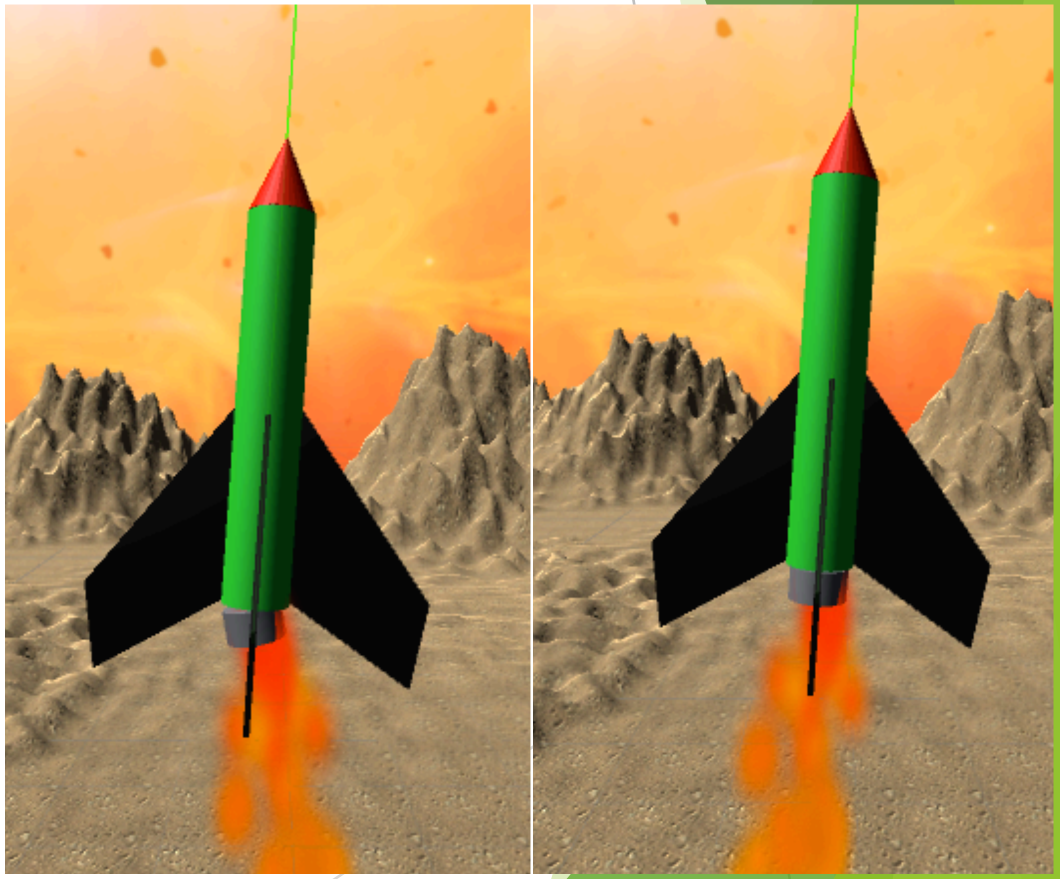
Stabilization

- Using Aerodynamics
- The rocket has fins which help stabilize the rocket.
  - We can unbalance the rocket in a desired direction by rotating them a little
- By rotating the wings on the opposite side of where the rocket is tilting, we can straighten it up.
- There is however a downside
  - When there is no air, we can create no drag



Stabilization

- Using Thrust
- If the rocket starts tilting, we can rotate the thruster, adding thrust in a different direction
  - This will add torque, rotating the rocket in the desired direction .
- Does work Both in space, and near the earth but it as very large forces is used, it is hard to control



Issues in the implementation

- The implementation however does not work for these solutions.
- We still need to work on it to make it work.
- Aerodynamics:
  - We need to calculate which wing must rotate, and by how much
- Thrust:
  - We need to balance the amount of force we add, such that the rocket does not rotate too much.