SKI JUMPING SIMULATION

COMP PHYSICS

Presented by : Alex & Pandya

What is Ski Jumping?

Ski jumping is one of the winter sports that known to be dangerous but also very exciting. It involves a skier descending a steep slope before propelling themself off an incline in an effort to cover the most ground possible before landing.

Is it extreme and dangerous?

Athletes that ski jump are susceptible to both severe injuries and overuse injuries. It requires skills, execution, and a lot of bravery.

Around

24

FATAL ACCIDENTS

According to The International Ski Federation (FIS) Database, there have been 24 fatalities in ski jumping since 1961.



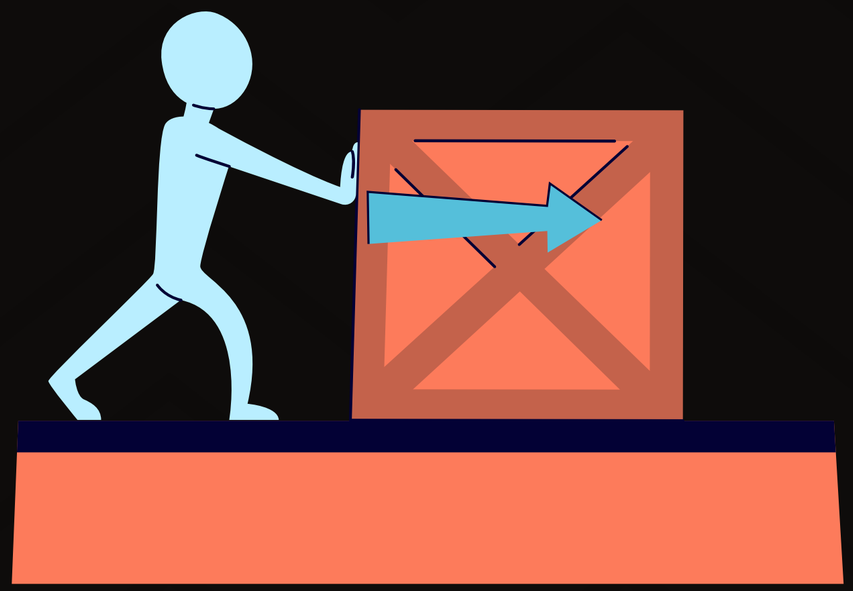
Proposed Solution

Creating an simulator that provides tools to show the design and analysis of ski jumps in order to calculating landing surface shape and ensure that the jumper always impacts the landing surface safely

LITERATURE REVIEWS

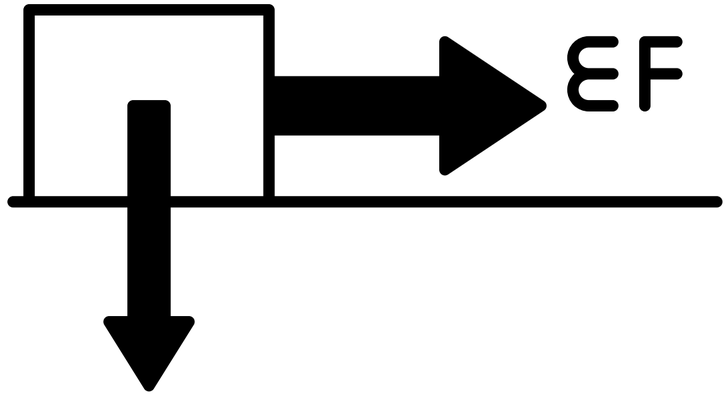
1	2	3
JUMPICUS - COMPUTER SIMULATION IN SKI JUMPING	**CREATING AN IMMERSIVE VIRTUAL REALITY APPLICATION FOR SKI JUMPING**	**DYNAMICS OF THE IN-RUN IN SKI JUMPING: A SIMULATION STUDY**
This study create a simulation in a computer called JUMPICUS. JUMPICUS can analyze real jumps by using digital cameras to capture the motion and evaluate joint angles, global position, and orientation. Ski jumping specific parameters can be evaluated and compared to ideal parameters.	This study create a prototype of a VR application for ski jumping. The goal for this project is to explore if such an application can be used to teach the basics of ski jumping, and if it can be used as a training tool for athletes.	This study simulates the in-run in ski jumping with a 4-segment forward dynamic model (foot, leg, thigh, and upper body) in order to analyze the mechanical demands on a ski jumper taking the in-run in a static position

PHYSICS FORMULA



NEWTON SECOND LAW

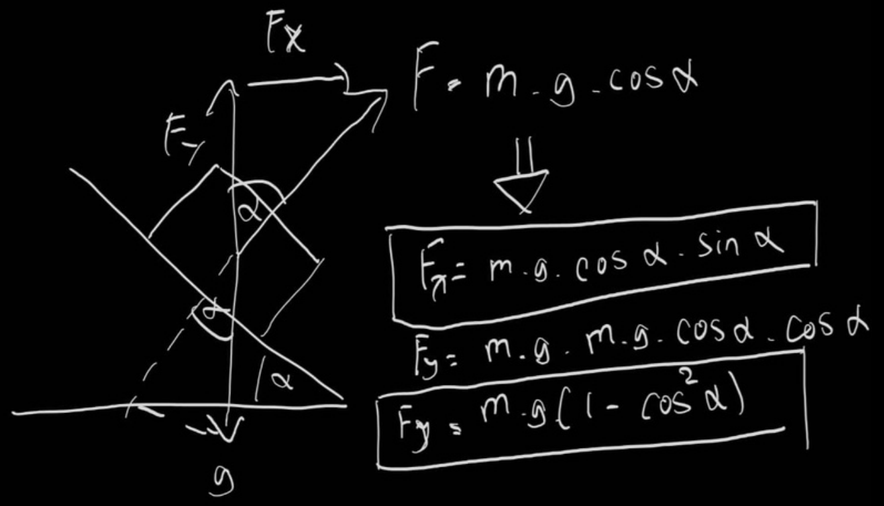
$$\Sigma F = ma$$



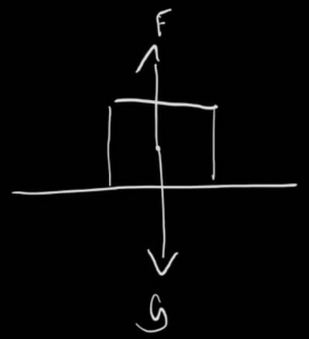
IMPLEMENTATION

PHYSICS FORMULA WITH PYTHON

```
# on the in-run
if posx < distance_ramp:
    gx = mass * -g * sin_inrun * cos_inrun * (1-friction)
    gy = mass * g * (1 - cos_inrun * cos_inrun) * (1-friction)
```



```
# on the ramp
        elif posx < distance_ramp + length_ramp:
            gx = 0
            gy = 0
            if vy != 0:
                vx = (vx ** 2 + vy ** 2) ** 0.5
            vy = 0
```



$F_x = 0$

$F_y = m \cdot g - F_{support}$ (Normal Force)

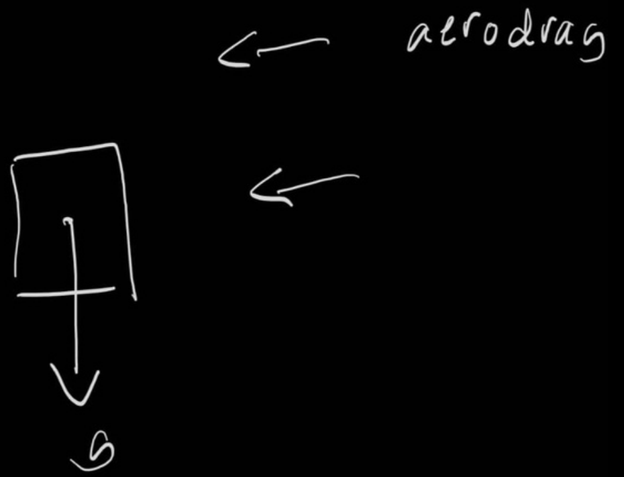
$= 0$

approximation

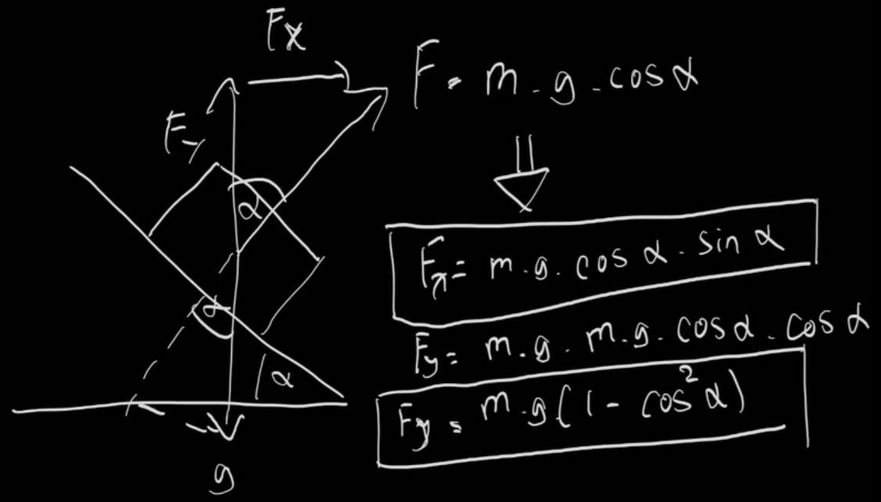
$\Rightarrow V_x = V \times y$

Ramp : Before Ramp

```
# flying and landing
        elif posx < distance_slope:
            slope_height_posx = (distance_slope - posx)*tan_slope+2.50
            # flying
            if posy >= slope_height_posx - 5.5:
                gx = mass * g * aerodrag
                gy = mass * g * (1 - aerodrag)
```



```
# landing
Else:
gx = mass * -g * sin_slope * cos_slope * (1 - friction)
gy = mass * g * (1 - cos_slope * cos_slope) * (1 - friction)
angle_landing = math.atan(-vy / vx) - angle_slope
jumped=((posx-(distance_ramp+length_ramp))**2+(posy-height_ramp**2)**0.5
v_new = (vx ** 2 + vy ** 2) ** 0.5 * math.cos(angle_landing)
vx = v_new * cos_slope
vy = -v_new * sin_slope
```



Simulator Demo

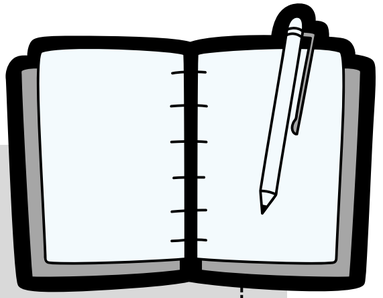
WITH VSCODE

EVALUATION

THINGS THAT WE EVALUATE

- Visualization
- Physics Simulation
- Bugs and Issues
- Future Enhancement

Conclusion



As a conclusion to this presentation, we have successfully completed this simulator project, which is a solution to the problem we addressed. Various evaluations have been carried out such as ensuring that the physics simulation can run properly, making the visualization and features of the simulator look refreshing and easy to understand, and fixing the problems in the program. This aims to make the simulation run well. It is hoped that this project can be useful as an educational material so that people understand how ski jumping is done.

Thankyou for listening!