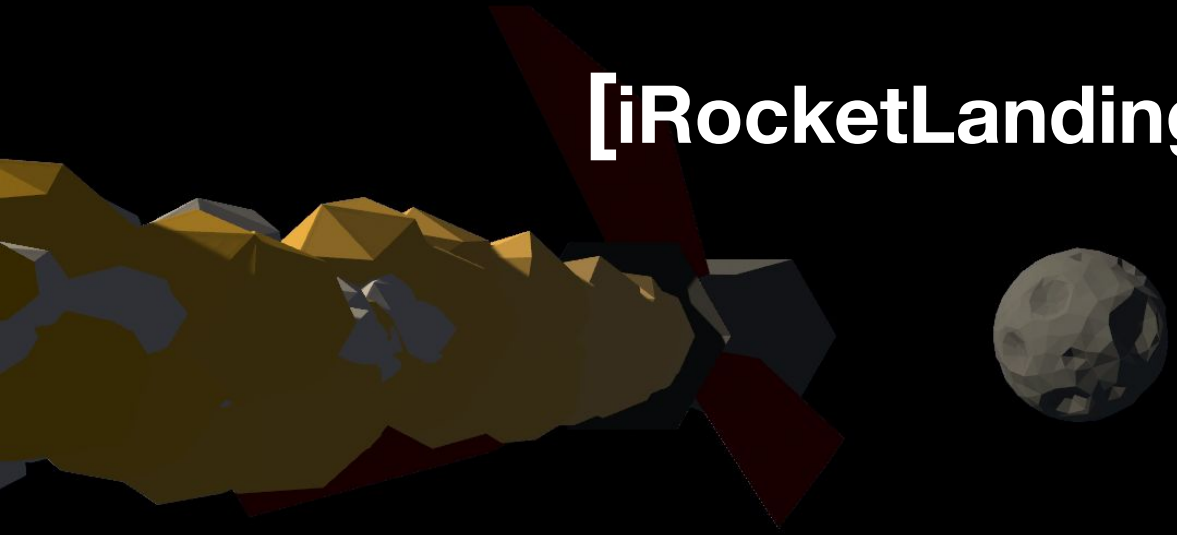
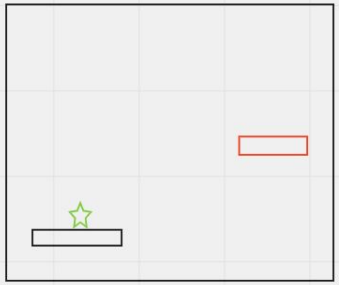
[iRocketLanding24] PEM-KI 2021



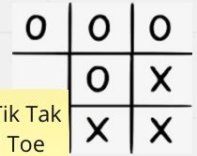
Andreas Ellwanger
Andreas Griesbeck
Thanh Le
David Schmider

Project idea

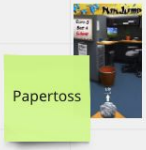
Ideation

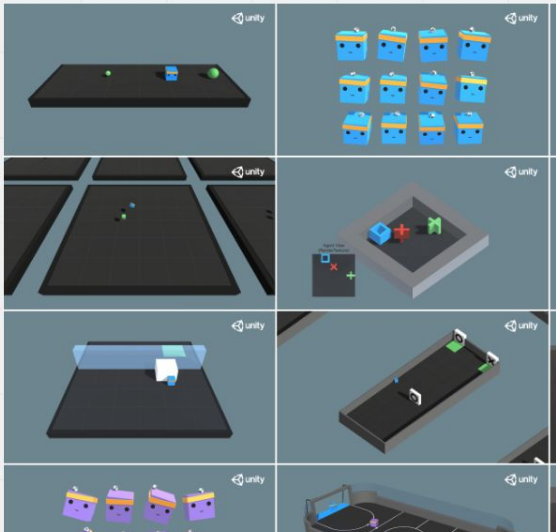



Raketenlandung
imgflip.com


Tik Tak Toe


Pong
Ocatagonal Pong


Papertoss




Darts


Moorhuhn


Würfel Fußball


Tactical round-based
Tank Game
Worms


Flipper

Idea

- simulate a rocket flight to the moon
- inspired by SpaceX
- allow rocket to launch from anywhere on earth
- land rocket bottom first
- include gravity
- make big impact on mankind

Approach

Organization

- create a simple proof of concept
- model world as first step
- improve iteratively
- use versioning
- add complexity stepwise
- pair-programming
- adapt parameters and run in parallel on different machines

Timetable

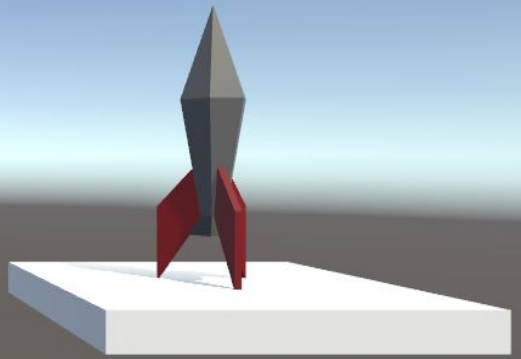
- Day 1 - World model
- Day 2 - Run first training, simple gravity, hit moon
- Day 3 - Refactor project, resize models, adapt parameters
- Day 4 - Add proper gravity, create different runs, create final pitch
- Day 5 - Fix time scaling, training

Technology stack

- Unity 2019.4.20f
- mlagents
- tensorboard
- Python 3.7.9
- collab
- Assets in use
  - Low Poly Rocket
    https://assetstore.unity.com/packages/vfx/particles/low-poly-rocket-trail-75911
  - Low Poly Planet Pack
    https://assetstore.unity.com/packages/3d/planets-pack-72089
  - Starfield Skybox
    https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717

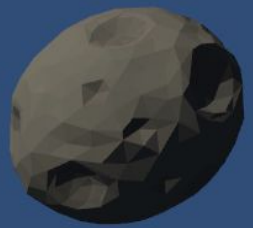
Iteration 1

- 2 platforms
- Pseudo 3D world
- Rocket should touch the target pad



Iteration 2

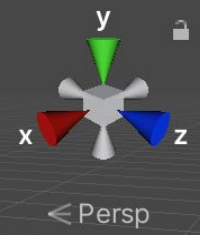
- 2 spheres with gravity
- continuous input parameters
- fixed launch, starting and end position



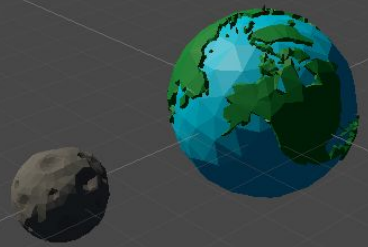
Iteration 3

- 2 spheres with gravity
- **random launch and moon position**

Configuration



- **Observations** (17 + 80)
  - 2x Ray Perception Sensor (3D) as child sensors
  - distance to moon / earth
  - direction to moon / earth
  - angle to moon / earth
  - own angle / position
  - rotation / x / y velocity

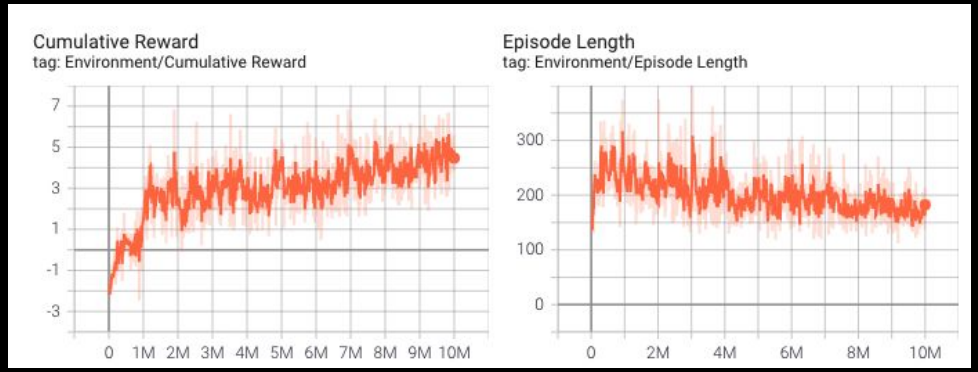


- **Actions** (3)
  - forward
  - left
  - right

- **Rewards**
  - lift off earth
  - decreasing distance to moon
  - touching moon
  - landing angle to moon near 0
    - low landing speed on moon
    - low angular velocity during landing
- **Punishments**
  - no lift off
  - fast rotation
  - earth collision
  - exiting bounds
  - per step / total steps reached

Training

- increased hidden units for higher complexity
- greater gamma for possible future rewards



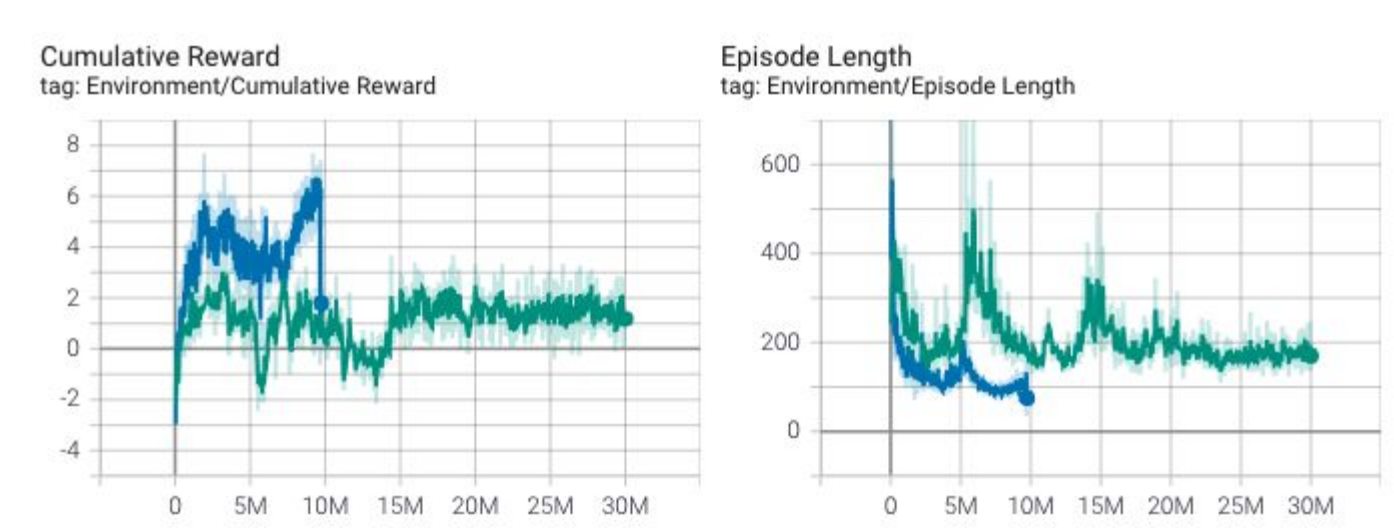
rocket_002.yaml

```yaml
behaviors:
  RocketBehaviour:
    trainer_type: ppo
    hyperparameters:
      batch_size: 128
      buffer_size: 1280
      learning_rate: 3.0e-4
      beta: 5.0e-4
      epsilon: 0.2
      lambd: 0.9
      num_epoch: 5
      learning_rate_schedule: linear
    network_settings:
      normalize: true
      hidden_units: 256
      num_layers: 3
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
    max_steps: 10000000
    time_horizon: 256
    summary_freq: 5000
```

Insights

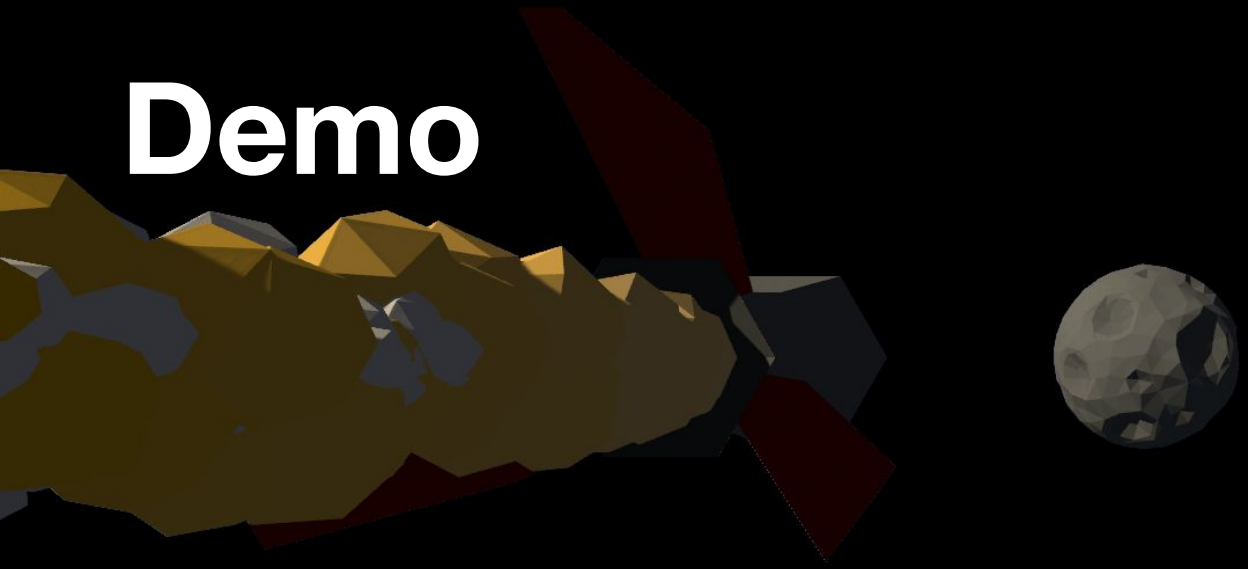
- Global vs local position
- Tried imitation learning, but dropped it (task too complex)
- Ray Perception Sensors
  - has to be 3D, 2D does not work in 3D Scenes
  - less is more
  - no 360° ray detection but rather some to the front and some to the back
  - can easily become the majority of inputs (if stacked, too many rays or tags)
- Parameters
  - at first: more input than hidden units -> widened model
  - deeper model
  - optimized gamma, epsilon & max steps
- `Update( )` / `FixedUpdate( )` on gravity -> sensitive to time

Insights (training)




STEPS(7)

Demo



Vision - Environment

- Real 3D world simulation (gravity, sizes, distances, orbit)
- Self-rotating earth and moon orbiting earth
- Solar System simulation with rotating planets, sun etc
- Simulate air(-resistance), pressure, atmosphere, temperature
- Obstacles like satellites, ISS and asteroids

Vision - Rocket

- Add realistic thrusters to rocket
- Simulate material & all relevant forces
- Booster rockets and capsule separation in orbit (multiple stages)
- Simulate fuel

Vision - Objective

- Land on a predefined target area on the moon
- Land smoothly
- Optimized fuel consumption
- Fly in a way the rocket would not break & not kill passengers
- Return to earth

Thank you!