

5.3

SR-7.2

Space Rat: Edition 7, Test 2

Zuza Petrushka

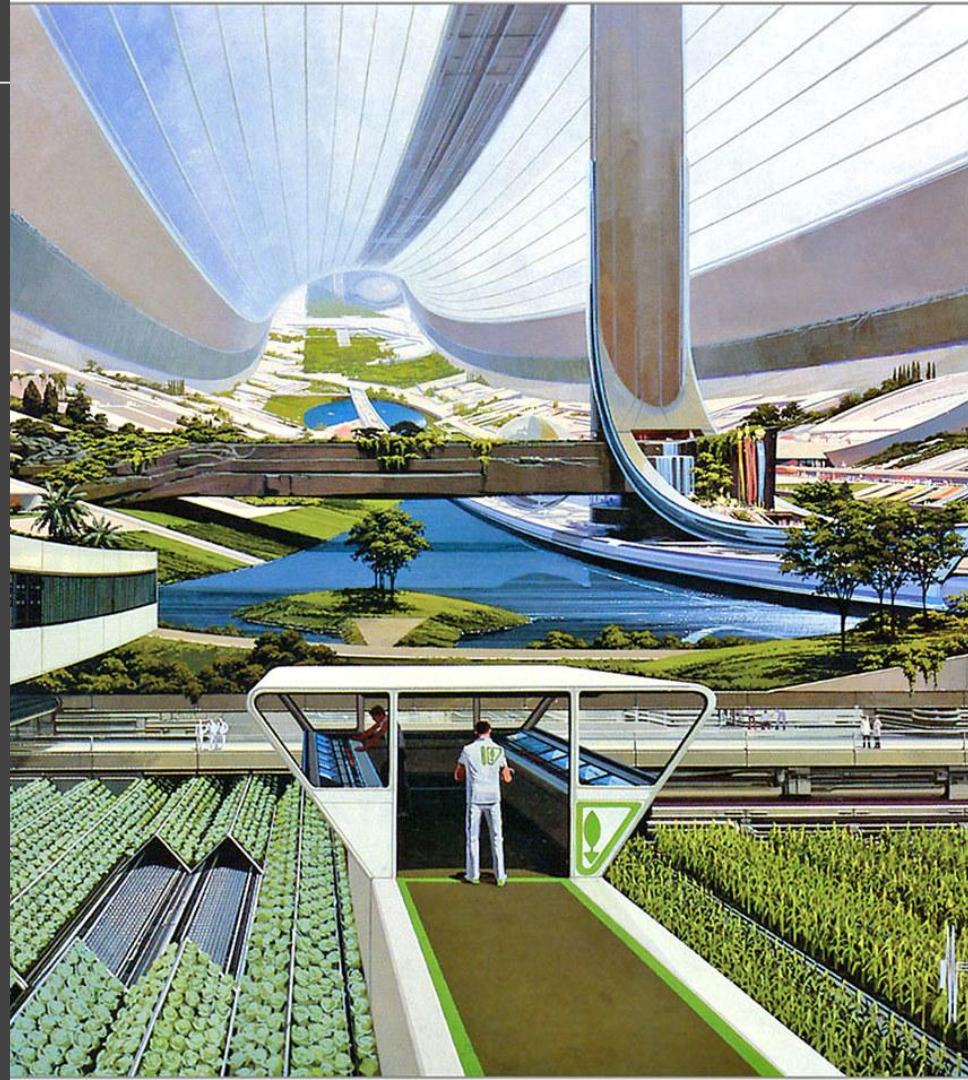


2050

Spaceports, Hotels, Warehouses, and Other Off-World Facilities

- ◆ Moving cargo safely in space is very different than here.
- ◆ Different industries have to adapt, while still providing jobs for their employees.
- ◆ AI can allow for accurate predictions, however a human is in charge of these operations.

Syd Mead, Orbiting Space Colony, 1976





Deep Space
Storage and
Warehousing_



Fast-paced Space
Shipping and
Delivery_

Maintenance,
Service,
Agriculture_

1. To ensure the **safety** of the human life.

Toyota Brand Beliefs_



Safety

Works in tandem with humans. Voice activated, manual feature, and a fueling system powered by water.



Reliable

3D printed sand manufacturing allows titanium frame to have a cradle life cycle; nothing is wasted.



Adjustable

Arm and tail structure allows unit to grab multiple structures and materials within space.



Sustainable

Reused titanium frame, with water fueling system, ensures that the environment back home is safe.

Why Rats?

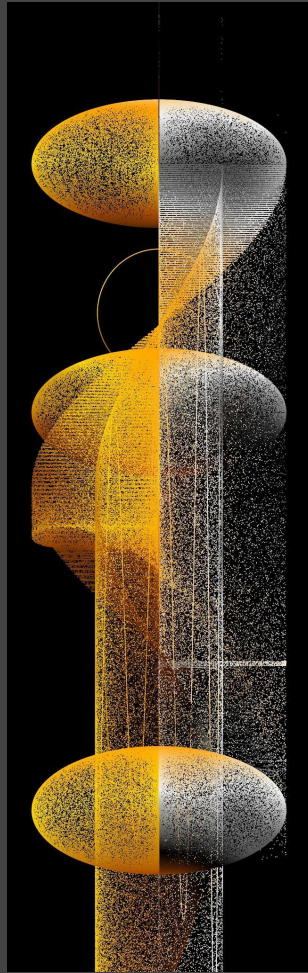
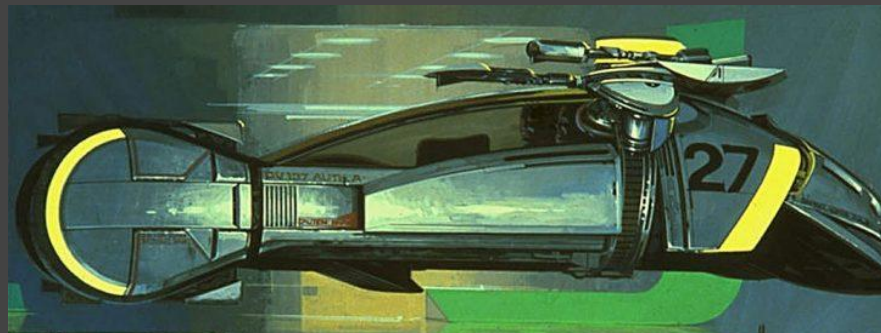
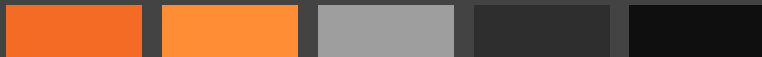
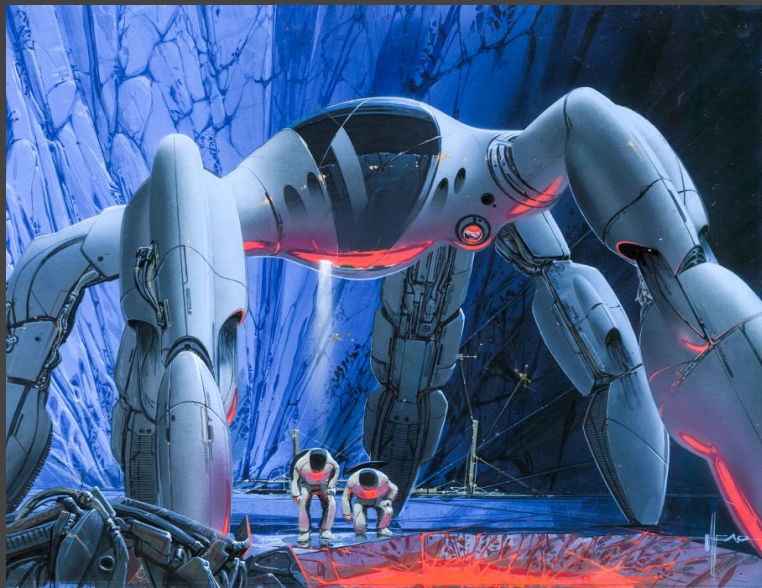
Rats have a unique sense of gravity in space, using their tail as balance.

Quickly adapting to zero gravity, they were the most stable species.

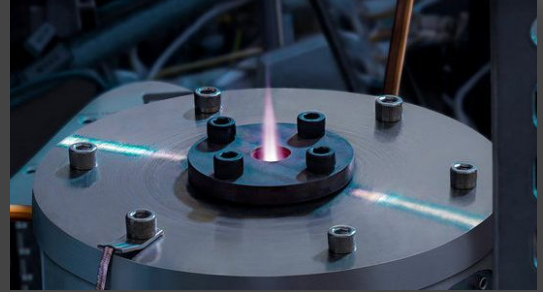
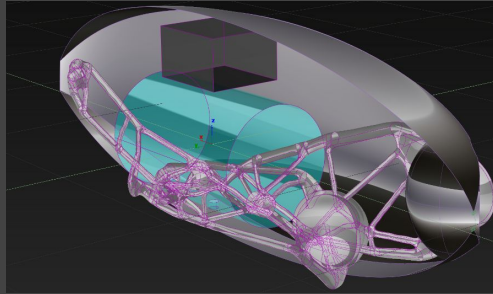
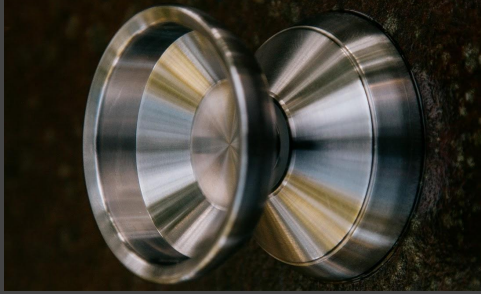
As their rear legs lifted up, their tail went down, making them have more control with minimal touchpoints.

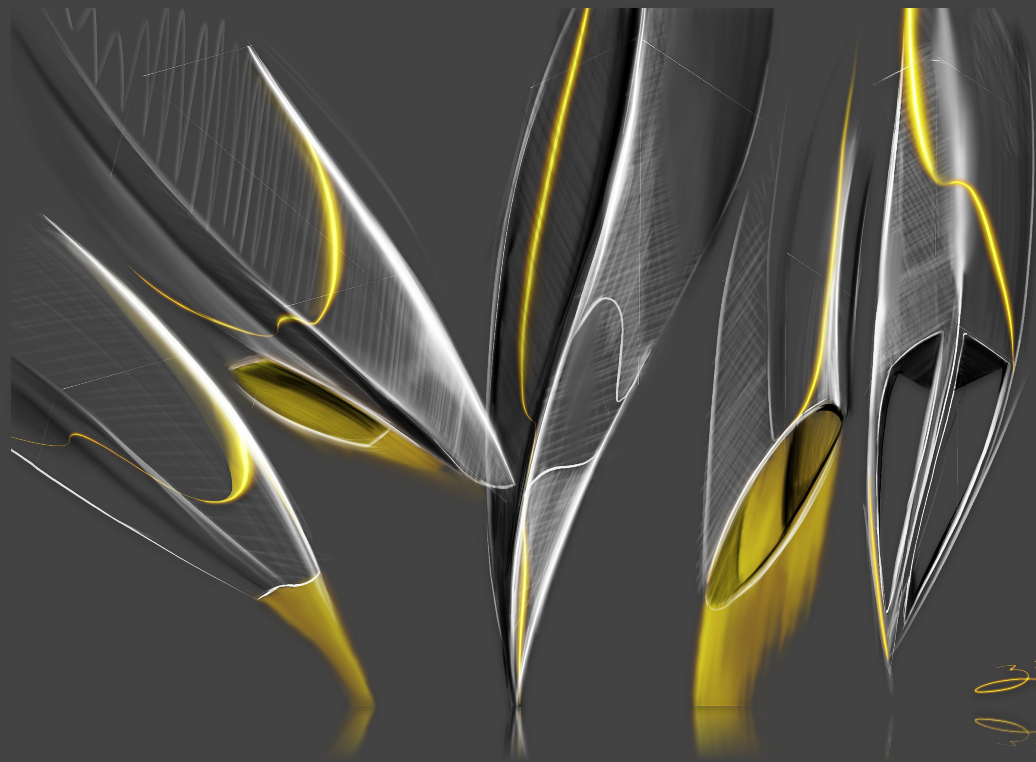
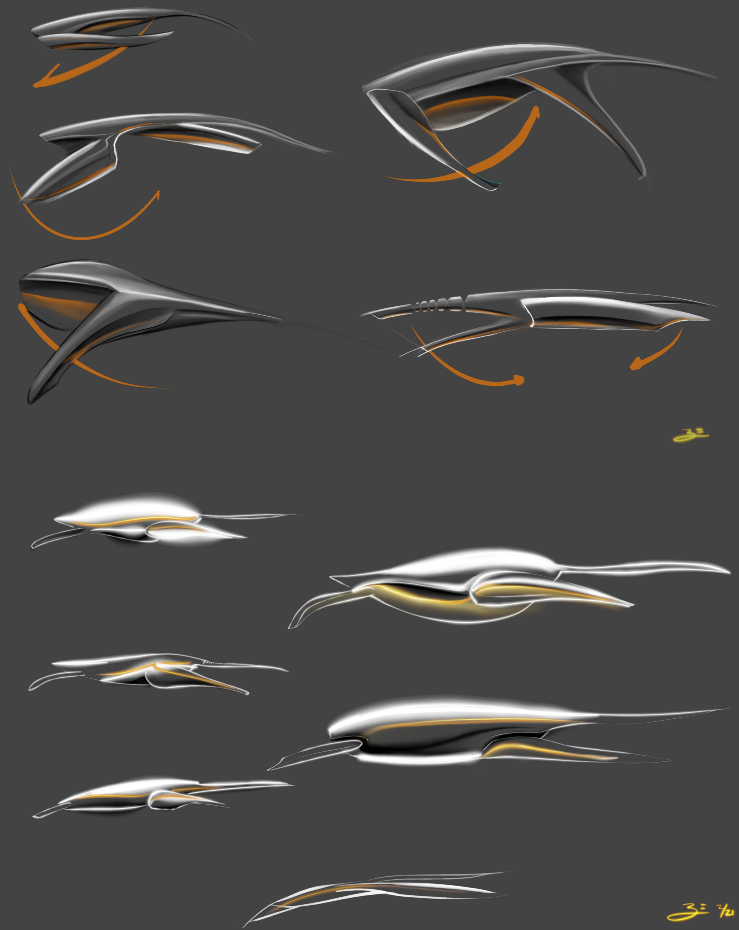


Mood

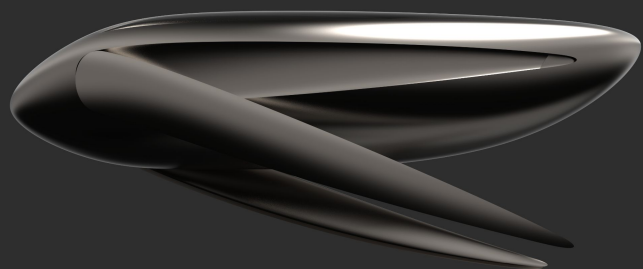
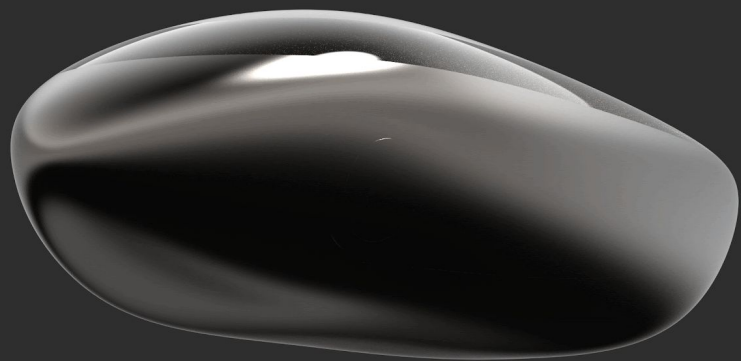


Materials and Tech

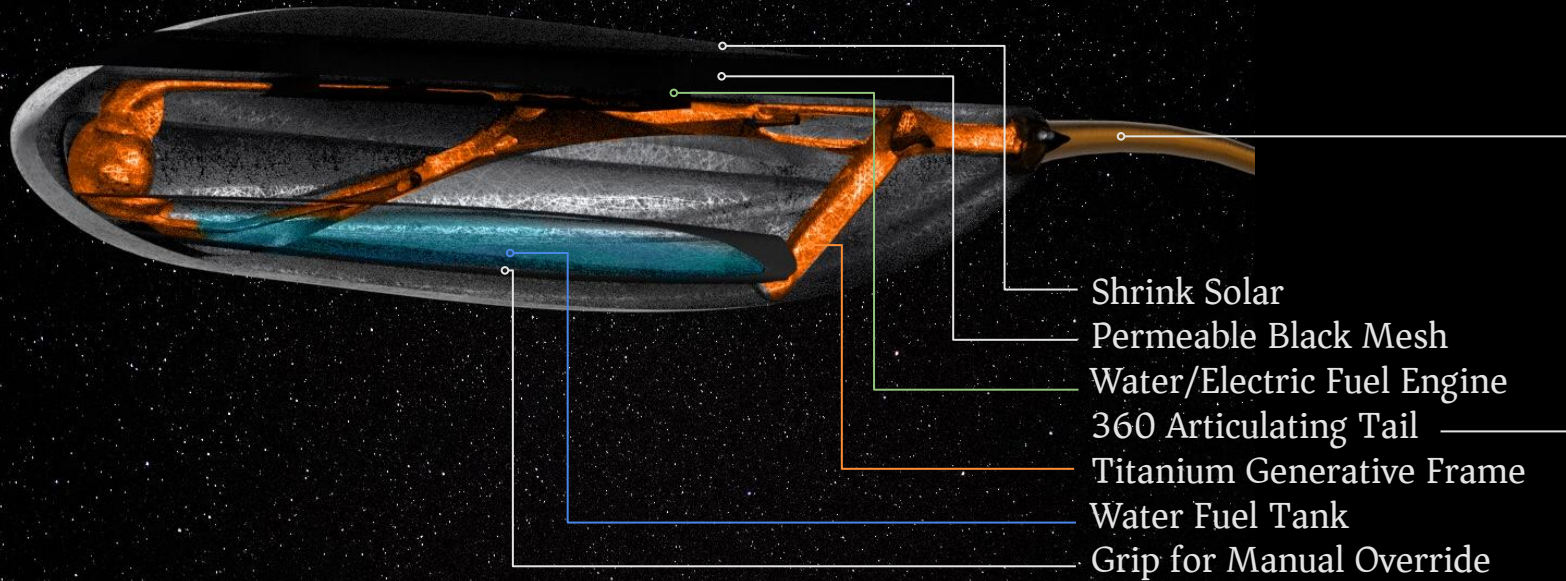


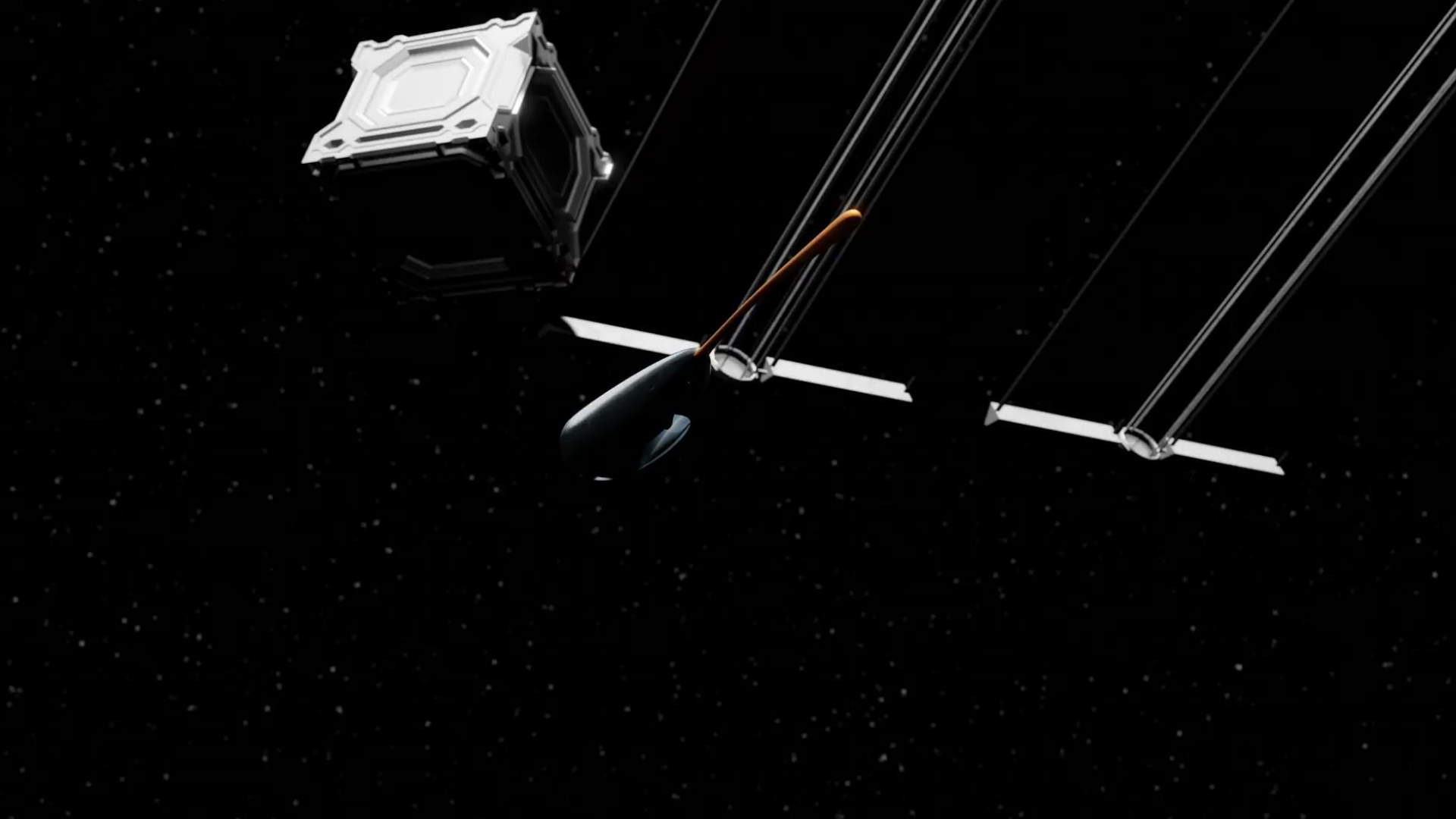


3.2 7.2
3.2 8.2



Interior





Thank you

Zuza Petrushka
May 5th, 2022, SP2022
ID5 with Matt Grigsby





S R - 7 . 2

Z U Z A P E T R U S H K A

```
private void RunTasks()
{
    Dictionary<string, string> dict = new Dictionary<string, string>();

    List<Task> tasks = new List<Task>();
    foreach (KeyValuePair<string, string> kvp in dict)
    {
        Console.WriteLine("Executing task " + kvp.Key + " ...");
        Task t = new Task(() => MyMethod(kvp.Key, kvp.Value));
        tasks.Add(t);
        t.Start();
        t.ContinueWith(task => Console.WriteLine(kvp.Key + " completed"));
    }

    Console.WriteLine("Waiting tasks to complete...");
    Task.WaitAll(tasks.ToArray());
    Console.WriteLine("All tasks completed...");
}

private void MyMethod(string arg1, string arg2)
{
}
}
```

await new orders: (...)

