DOCUMENTATION FLATTIVERSE

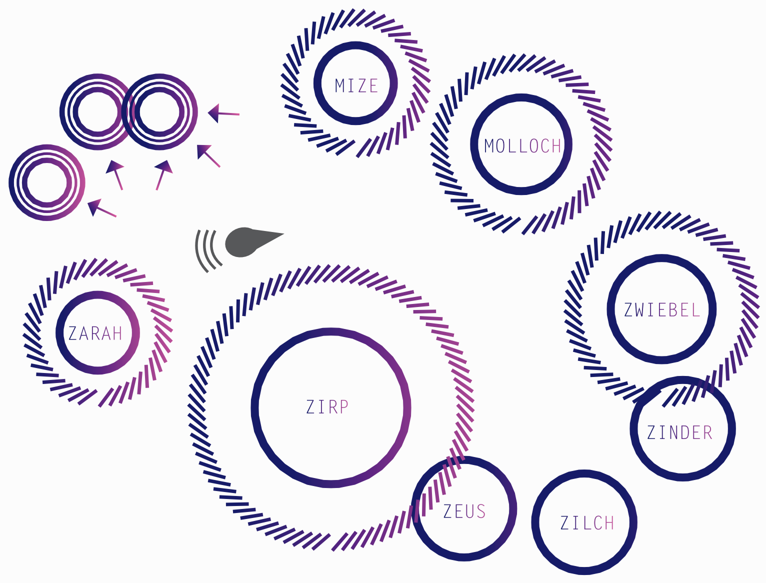
2024

OVERVIEW

WELCOME

Hello, pilots! I‘m Commander Ed Grayson, or just Commander Ed, leading this mission in the FLATTIVERSE universe.

I‘ve seen many generations of spaceship captains exploring Flattiverse, some peacefully and others not so much. There have been battles for mission objectives, and even the earliest spacemen shared stories of alien planets and suns:



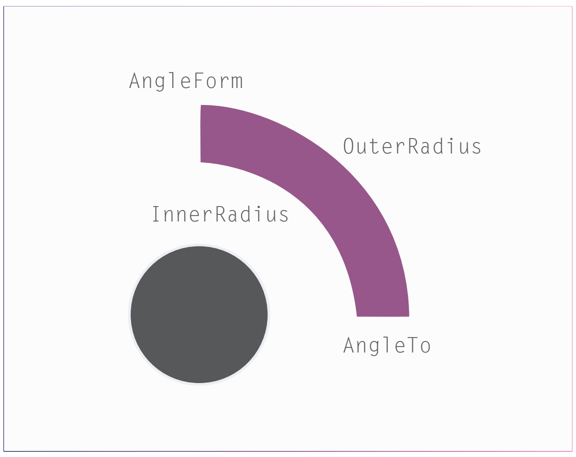
Be careful, as heading out without a well-prepared ship is not adviced. There are stories of ships without good visibility and navigation crashing into planets due to gravitational forces. Don‘t underestimate more experienced ships; they might see smaller ones as easy targets.

But don’t worry, I will be your guide. I have read all the old records and I am ready to share all my knowledge!

SUNS AND BLACK HOLES

Getting too close to a sun always results in total destruction, just like the deathly danger from planets, moons, meteroids or black holes. Suns emits energy and ions on specific sections, while black holes exert additional gravity at specific sections.

These specific Sections consists of two angles (from ... to) and the distance from the sun‘s or black hole‘s center.



A 90° section in front of a sun.

Sun Energy

Spacecrafts can use the energy from suns to recharge their power and is necessary for the ship‘s propulsion and functions. It provides essential power to all ship systems.

Ions

Ions are the primary energy source for the ship‘s protective shield.

WORMHOLES

Wormholes are connections between universes that can be traversed.

PLANETS, METEOROIDS AND MOONS

Planets, Meteoroids and Moons are usually safe and, like suns and black holes, pull things toward them with gravity. If a spaceship isn‘t good at navigating, it might still get pulled in and crash into them. There are resources to be found on specific Sections on these three types of celestial bodies, which are explained in the following section. These specific Sections work in the same way as they do on suns and black holes.

PLANETS, METEOROIDS AND MOONS

Planets, Meteoroids and Moons are usually safe and, like suns and black holes, pull things toward them with gravity. If a spaceship isn‘t good at navigating, it might still get pulled in and crash into them. There are resources to be found on specific Sections on these three types of celestial bodies, which are explained in the following section. These specific Sections work in the same way as they do on suns and black holes.

RESOURCES

Compared to older Flattiverse versions, this version of the game is more resource-driven. Resources are crucial for building a stronger ship as you progress in the game.

Tungsten (Wolfram)

is rare and valuable. It‘s used for strong materials and high-tech upgrades.

Iron (Eisen)

is common and essential. It‘s the main material for building and fixing our ships.

Tritium

is rare and powerful, key for fast engines that let us travel far

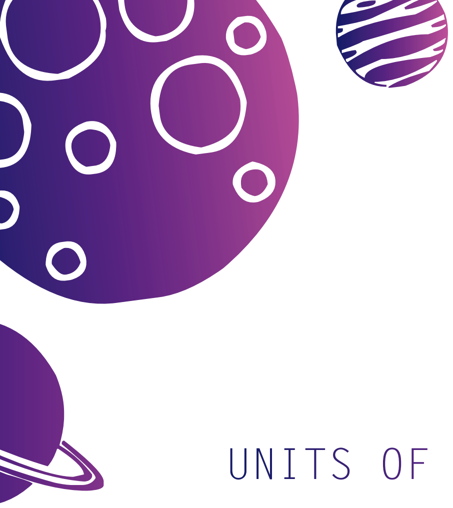
Silicon

is plentiful and important for our ship‘s computers and systems, helping with direction, talking, and staying safe.

These resources help us upgrade our ships and explore further. They‘re not just materials; they tell stories of the universe and our place in it.

As we explore the FLATTIVERSE, remember these resources are crucial. Finding them is part of being a pilot and helps us understand the universe.

Let‘s use these resources smartly as we adventure into space. They help us survive, grow, and leave our mark among the stars.



UNITS OF THE UNIVERSE

All units of the Universe are represented by a Unit-Class. The Unit class is the superclass of all objects in a universe. Every object has fundamental properties

Name

Unique name of the Unit.

IsVisible

Describes if other units can see this unit. That’s important for scanning and spotting the unit.

Movement

The speed of the Unit relative to the scanning ship. Can change for mobile Units.

Gravity

Gravitational pull on other Units

Mobility

Stationary, on a fixed orbit, or mobile.

UnitKind

Type of a Unit, like a sun, black Hole, …

Cluster

The cluster is a part of the galaxy.

Direction

The way the Unit is facing

Position

Position relative to the player‘s ship.

Radius

Size of the Unit.

Team

Name of the team, which you belongs to (if you belong to one)

DISTANCE AND SPEED MEASUREMENTS

Flattiverse is based on an absolute coordinate system. This is a change compared to previous game versions, which were based on a relative coordinate system.

Flattiverse represents all information related to positions and movements using its own **Vector** class. A **Vector** consists of X and Y components and includes a variety of useful vector operations, such as rotation or changing the vector‘s length. Additionally, a vector provides simple information about its length and angle.

The **Vector** class also overloads some operators like + or -, leading to relatively concise formulas (compared to Java). Multiplication with and division by scalars alters the vector‘s length, and comparison operators like < or > compare their lengths.

```
 public class Vector
{
    public double X; // X-compomemt
    public double Y; // Y-component
    public double Length; // vector length
    public double Angle; // vector angle

    public Vector();
    public Vector(double x, double y);
    public static Vector FromAngleLength(double angle, double length);
}
```

GRAVITATION

Gravity is crucial in Flattiverse. All units that exert gravity `(unit.Gravity != 0)` attract all units that are susceptible to gravity `(unit.Mobility == Mobility.Mobile)`.

In theory, one can calculate the exact gravity only if one knows all units with `Gravity != 0`, which is practically impossible since even enemy or unknown ships have gravity. However, it is still possible to locally estimate gravity approximately because it decreases with greater distance.

The Gravity algorithm calculates the distance between the two units for which the gravitational effect is to be determined:

```
Vector diff = Gravitated.Position - gravee.Position;
```

It then trims this distance vector to a minimum distance of 100 (to prevent excessive gravity if you get even closer)

```
diff.Length = gravee.Gravity * -100.0 / (diff.LengthSquared > 10000.0 ?
diff.Length : 100.0);
```

Then adds this to the movement of the attracted unit:

```
Gravitated.Movement += diff;
```

HOMEPAGE

REGISTRATION

To become part of the FLATTIVERSE crew as a pilot, registration with your Email-Address on the FLATTIVERSE portal is required: http://www.flattiverse.com

After conforming your Email-Address (Double-Opt-In) you will be able to logon to the page.

Please go to your profile to copy your Api-Key. The Api-Key is necessary to logon to the GameServer with your space ship.

Be especially careful with your API-Key! You need it to identify yourself and communicate with the server!

In your profile, you can upload an Avatar, change your Username, your Password and generate a new Api-Key, if necessary.

UNIVERSES

This is a beginner-friendly universe designed for pilots undergoing their first flight lessons.

PLATZHALTER

UNIVERSES

The first significant flight mission that will put your ship to the test.

PLATZHALTER

DOWNLOADS

On the Flattiverse homepage, you can download some useful stuff.

PLATZHALTER

SELECT YOUR SPACESHIP

Now it is time to select your spaceship to embark on your adventure in one of the universes. While you can choose a fully hardware equipped spaceship here, your reponisbility is to programm an environment to ensure that navigation and defense mechanisms are operational.

Some of these parameters might vary in their maximum value depending on which ship you choose. Resources are needed to upgrade the ship, and each ship has various levels.

Engerie

Necessary for the ship's propulsion and functions. It provides essential power to all ship systems.

Ion

The primary energy source for the ship's protective shield, shielding it from external attacks and dangers.

Hull

The structural outer layer of the ship that surrounds and protects all components from damage.

Shield

An energy-based defense system that surrounds the ship, shielding it from enemy fire and space hazards.

Nozzle

Steering nozzle that controls the direction of the thrust, allowing precise movements and changes in direction.

Extractor

A system used for resource extraction, essential for supplying and maintaining ship functions.

Thruster

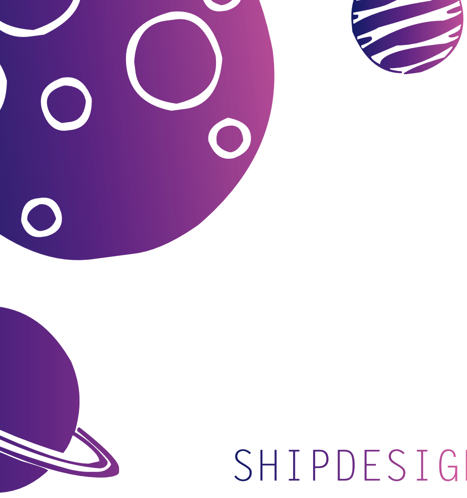
Propulsion system that moves the ship through space, ensuring necessary speed and maneuverability.

Weapon

Armament of the ship used for both attack and defense against threats.

Cargo

Space for storing collected resources.



SHIPDESIGN

The ShipDesign class holds all values of that will be relevant when you construct a ship.

You can access all values via the ShipDesignConfig-Class which you can get from ship.Config.

Those are the values that your ship will have right from the start.

Of course, these values will not suffice for a seasoned player.

This is why you can strengthen your ship with upgrades - also contained in the ShipDesign class!

UPGRADE

To improve your ship’s stats, you can build upgrades. An Upgrade costs specific ressources.

Some upgrades may require other upgrades to be build first - you can check the PreviousUpgrade value in the upgrade.Config.

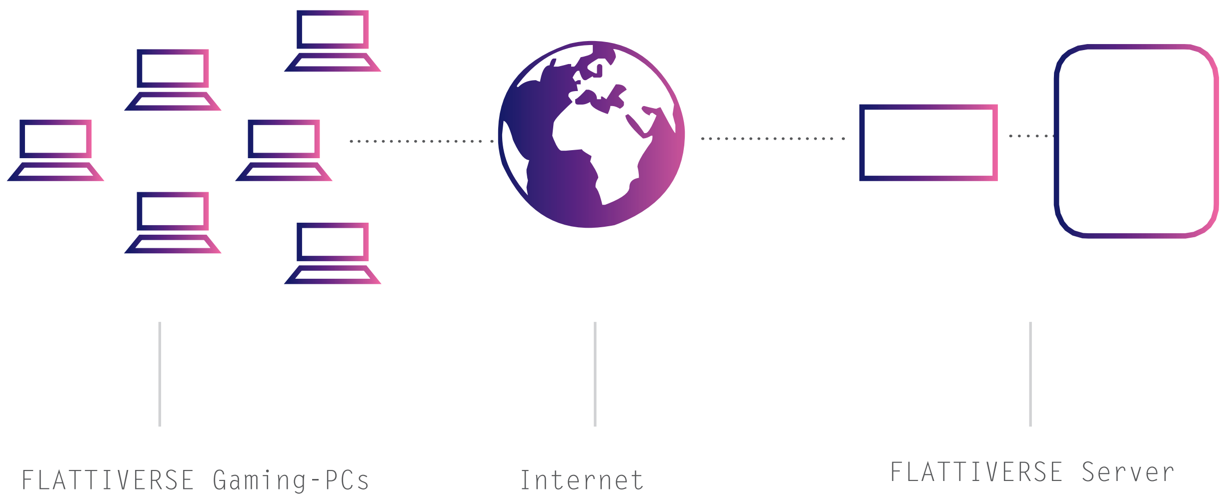
You can’t build right away - you will need to collect resources first.

Which and how many of them is also listed inside the respective upgrade.

CONNECTION TO FLATTIVERSE

CONNECTION TO FLATTIVERSE

The Flattiverse simulator runs on a game server on the internet, and the spaceship programs access it through interfaces in a local DLL. This ensures that all participants adhere to the same rules of the game. Communication between the individual computer and the server takes place through the Flattiverse.Connector.dll, which must be included in the project.



In comparison to previous versions, the Flattiverse game server now has a fixed tick rate, which enhances the real-time gaming experience. This also means that players with laggy ships can‘t significantly impact the fluidity of the entire game.

FETCH AND INTEGRATE CONNECTOR WITH NUGET

The Flattiverse.Connector.dll is available as NuGet package. The Connector establishes the connection to the Flattiverse server.

Flattiverse.Connector
.NET 8.0

To get a NuGet package using the .NET CLI (Command-Line Interface) in the console, you can use the dotnet add package command. Here are the steps:

1. Open a command prompt or terminal window.
2. Use the following command to add a NuGet package to your .NET project:

```
dotnet add package Flattiverse.Connector
```

If you are working with a specific project, make sure to navigate to the project‘s directory using the cd command before running the dotnet add package command.

CONNECTING TO THE GAME-SERVER

The Connector.dll comes with the **Flattiverse** namespace.

```
using Flattiverse.Connector;
```

To connect to a **Universe** it needs to be instantiated first:

```
Universe universe = new Universe();
```

You can overload this constructor to connect to a different server instead of the course-server. After establishing the connection, you can query some information by using universe.Galaxies, universe.Teams and universe.Players.

If you know the name of a **Team** or a **Galaxy**, you can also get information about those:

```
GalaxyInfo galaxyInfo = universe.Galaxies[“Beginners Course“];

TeamInfo teamInfo = universe.Teams[“Red“];
```

The Universe can be joined by using the Api-Key you can find in your account on the Homepage:

```
Galaxy galaxy = await universe.Join(“gameserver-address“, “API-Key“,
teamId);
```

During connection establishment, the Connector can throw exceptions for various reasons:

- IOExceptions if there‘s an issue with the connection.
- GameExceptions if the server reports errors, such as an incorrect password.

In order to connect with your Ship you need to register it to the **Universe**:

```
galaxy.RegisterShip(string Name, ShipDesign design)
```

Name	Name of the ship
ShipDesign	The current ship configuration / design

Here are some additional important Connector classes that store information from the game-server:

Galaxy	A Galaxy contains one or more universes and defines the game rules for them, such as the maximum number of Players, eligible Teams, and the Game Type (Mission, Shoot-the-flag, etc). Players in the same Galaxy can send each other messages (FlattiverseMessages).
Cluster	A Cluster is the playing field for spaceships. All objects in the Cluster are derived from Unit. The names of objects within a Cluster are unique.
UniversalHolder	UniversalHolder stores objects in a way that the server can update them through the Connector. This property is only lost when reading a list. However, UniversalHolder itself supports iterations, making listener creation unnecessary.
Player	All current online users.
PlayerKind	Whether the player is a Player, Spectator or Admin
Team	It can be relevant which Team a SpaceShip or MissionTarget belongs to
Controllable and ControllableInfo	Objects that the Player can control are Controllable, such as Ships, Probes, and Starbases. ControllableInfo objects contain publicly visible information about one‘s own or someone else‘s Controllable. The type of Controllable or what the ControllableInfo refers to is determined by the derived class.
Upgrade	Upgrades to Ships can be made when enough Resources are available.
Account	One‘s own Account instance is found in connector.Account. Galaxies contain a list of the accounts of the participants playing there. Self-scanned Ships carry the Account of their owner.

SENDING AND RECEIVING MESSAGES

RECEIVING MESSAGES

Messages come from the server as notifications, for example, when a player has entered a UniverseGroup (GameMessage). Additionally, players can send messages to each other.

```
[13:16:32] -SYSTEM- Flattiverse-Engine #0.9.6.1 on ftv://galaxy.flatti-
verse.com:22 bids you welcome, Cliff McLane.
[13:16:32] -MOTD- Hello Cliff McLane,
-MOTD-
MOTD- Thank you for connecting to Flattiverse. We wish you a pleasant
stay. For a quick start,
-MOTD- try to avoid unnecessary dialogs, such as a login dialog or a
choose universe dialog.
-MOTD-
[13:16:32] -SYSTEM- You are currently on Rank #10 with a PVP-ELO-Rating
of 0,00 and no Tournament-ELO-Rating.
[13:16:32] -SYSTEM- You are currently on Rank #10 with a PVP-ELO-Rating
of 0,00 and no Tournament-ELO-Rating.
[13:16:32] -SYSTEM- You haven‘t executed Connector.DoBenchmark() or
Connector.LoadBenchmark(...). We strongly encourage you to do so or you
won‘t be able to enter some of the UniverseGroups.
[13:16:32] Player Cliff McLane from Team None joined the game.
[13:16:32] Ship DasBoot of Cliff McLane continued game.
[13:16:41] Ship DasBoot of Cliff McLane collided with Sun Out-Star.
```

GLOBAL MESSAGES

Connect to a universe group List of all available universe groups on the websi-te MessageArrived Event“

```
await galaxy.SendMessage(1, “This is a message.“);
```

MESSAGES TO INDIVIDUAL PLAYERS

THE SPACESHIP

Celestial objects might be pretty to look at, but Flattiverse would be a pretty boring with just looking and no flying (and shooting).

That’s where Ships come into play (ha, see what i did there?) - Ships you can control!

FLATTIVERSE 2024 || THE SPACESHIP

There are three ways in which you will encounter Ships:

- Scanned Ships, which you encounter, well, when you scan them.
- As a Controllable, the way you control your own ships.
- Openly as ControllableInfo, something everyone who listens to events can see, containing some general infos about the Ship.

See the pages below for some addional information and stay safe in the Flatti-verse!

SCAN

NEW: Scanning works automatically; you just need to specify the direction.

In order to scan units scanning needs to be switched on. Scanning consumes Energy.

```
ship.Scan = true;
```

A scanner detects all units that fall within its scanning beam, even if only partially. For instance, it can determine the center of suns based on their surface characteristics, even if it's outside the scanned area.

The scanner identifies all units, including their name, direction, distance, and speed. This allows for the clear identification of players and their movements.

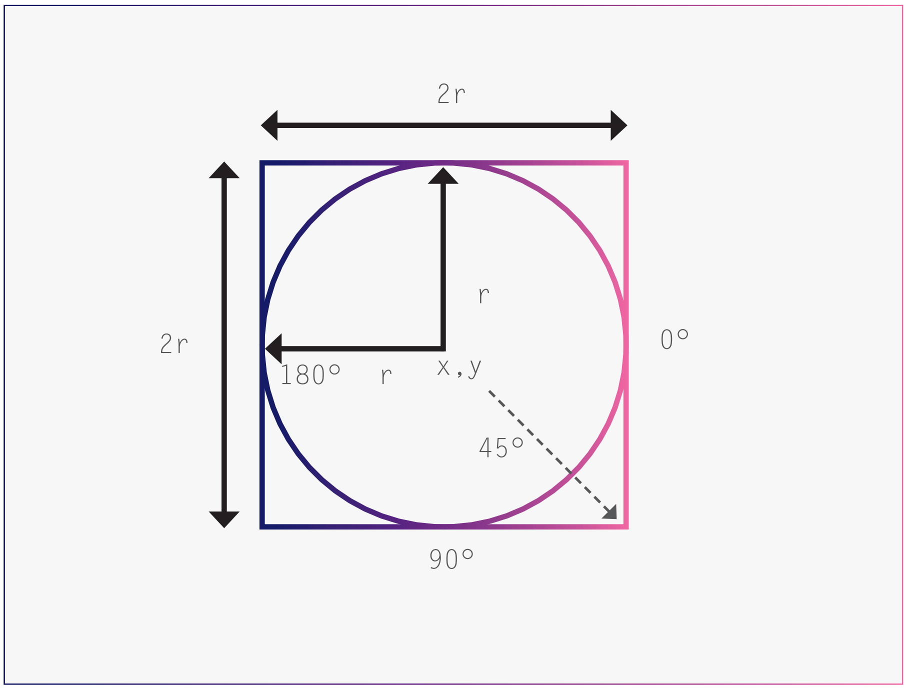
The result of a scan is returned to the connected player via events.

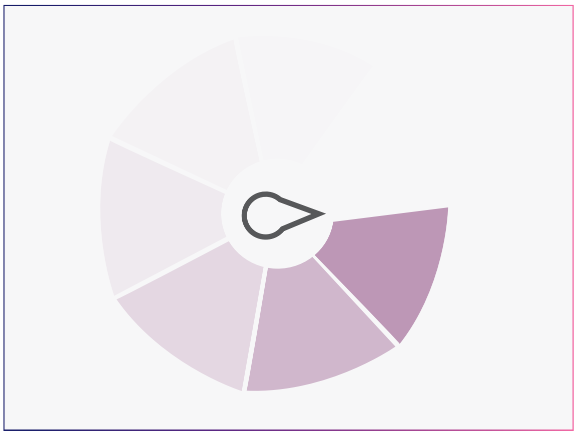
As already mentioned in the chapter **Suns | Wormholes | Planets | Resources** , the server in Flattiverse maps all objects and processes to so-called Units and derived classes. The most important fields or properties of a Unit are:

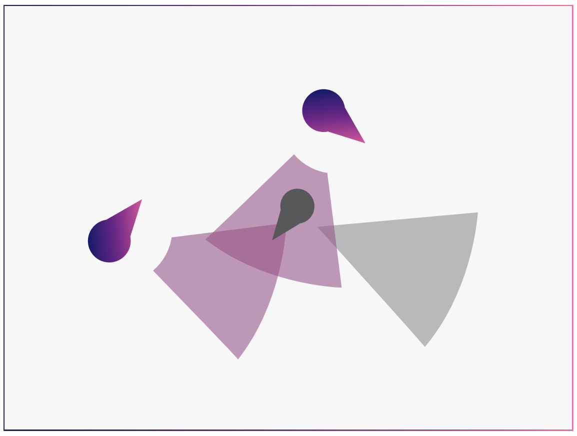
```
 public class Unit
{
    //Position of the object(relative coordinates)
    public Vector Position;
    //Movement Vector (relative coordinates)
    public Vector Movement;
    //Radius of the object
    public int Radius;
    //name of the object
    public readonly string Name;
    //Gravitational pull on other Units
    public readonly Vector Gravity;
    //Stationary, on a fixed orbit, or mobile
    public readonly bool Mobility;
}
```

Derived classes from this include, for example: `Sun`, `Shoot`, `Trash`, `BlackHole`, `Explosion`, `PlayerShip`, and `Buoy`. Some of these may have additional parameters; for instance, suns have a radius of the solar corona.

In order to display a radar screen to navigate more easily objects need to be displayed according to their coordinates and their radius:







NAVIGATION

The art of flight lies in jumping and missing the ground. Similarly, to be a good Flattiverse pilot, you need to avoid crashing into celestial objects and (optional) other ships.

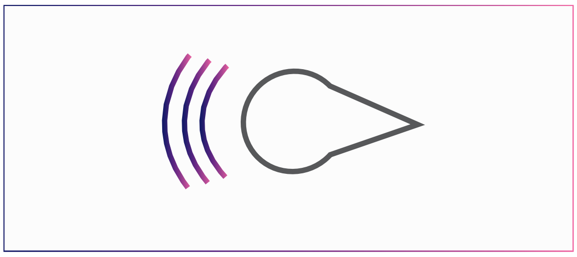
In order to do this, you have two tools: Thruster and Nozzle. The value of thruster controls how much you accellerate towards where your ship’s nose is pointing. Nozzle controls how fast said nose is changing its orientation.

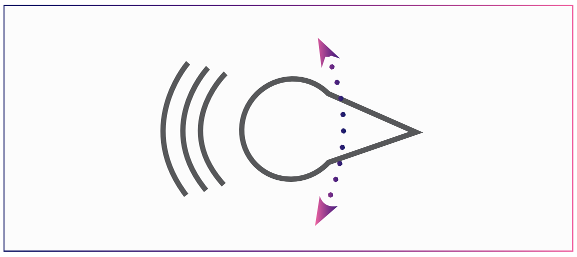
The maximum speed of a ship is limited. Smaller ships fly faster than larger ones, because they are lighter. Acceleration consumes energy, and Flattiverse initiates acceleration only when the spacecraft has sufficient energy reserves. If the energy is not enough, the ship breaks apart.

Taking control of your ship could look something like this:

```
ship.SetNozzle(-0.05d);

ship.SetThruster(2d);
```





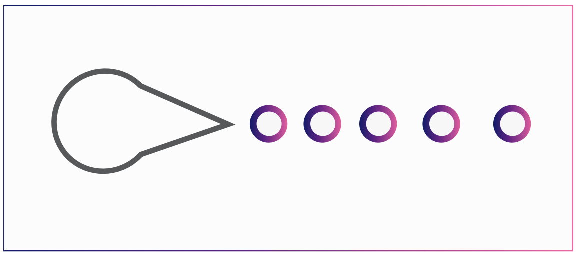
SCHIESSEN

Ships can deploy photon torpedoes that cause damage to other ships.

Aside from the velocity and ticks, all other proerties of the torpedo are defined by the ship's systems. The load value, for example, determines the size of the explosion radius upon detonation. If ships are within the detonation circle (ship's radius overlaps with the explosion radius), they suffer damage that can lead to damage or the destruction of the ship.

velocity	velocity of photon torpedo
ticks	number of ticks the photon torpedo is flying until it explodes (it will also explode when it hits a target)

NEW: You can only shoot in the direction the ship is pointing with it's nozzle - and in the opposite direction by providing a negative velocity.



```
ship.Shoot(double velocity, int ticks);
```

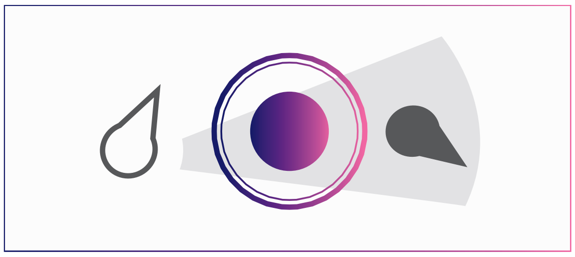
Depending on the torpedos properties, its path is affected by the gravity of planets, black holes, suns or other ships.

KOLLISION

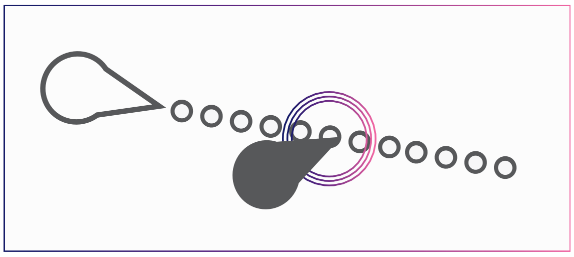
MASKING

Ships can deploy photon torpedoes that cause damage to other ships.

Aside from the velocity and ticks, all other proerties of the torpedo are defined by the ship’s systems. The load value, for example, determines the size of the explosion radius upon detonation. If ships are within the detonation circle (ship‘s radius overlaps with the explosion radius), they suffer damage that can lead to damage or the destruction of the ship.



KOLLISION



POINTS AND RANKING

Kills, deaths, and “neutral deaths” (e.g. collisions with celestial bodies) are counted towards your score. Flattiverse keeps track of your score throughout your session as well as your total score over time.

Not only players (local) have scores - accounts (global) and teams are also rated.