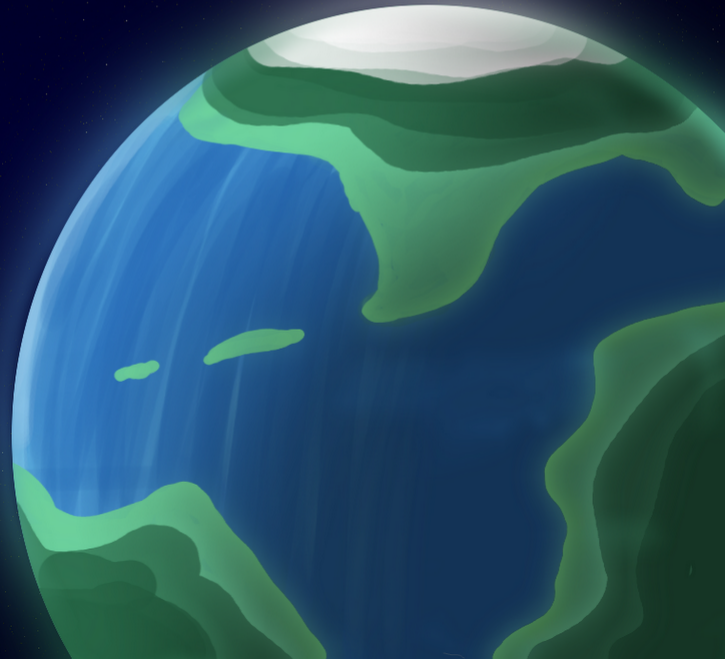


SPACE TROUBLE!



Action-packed multiplayer
space battles...

... with just
the right amount
of chaos!



Development Notebook

ETHZ Game Programming Lab 2012

Team: Random String Games

Team Members: Daniel Zimmermann

Rudolf M. Schreier

Adrian Blumer

1 Formal Game Proposal

1.1 Game Description

Overview

“Space Trouble!” is an action-packed space shooter where up to four players navigate their ships through the gravitation fields of a 2.5D solar system, battling each other using various weapons and tactics. The whole thing is spiced up with destructible planets and special items providing interesting abilities, as well as deadly weaponry.

General Requirements and Design Decisions

In this section, we first state a set of goals and expectations for our game. The following points should then serve as justification for the gameplay description that follows in the next section.

- As a party game, every action shall be **easy to understand**. Ideally, our game should be easy to learn and hard to master, although with a clear main focus on the former. An important point towards reaching this goal will be to find a proper balance between chaotic and strategic gameplay elements.
- We state that the games you play with your friends are most fun if the **challenge comes from the person next to you**. Of course we want the players to play the game again. This is well accomplished by generating a “next time I will beat you” or “let’s gang up on him next time” atmosphere.

This requirement also facilitates the development process: Challenging the players through well-balanced level designs or complicated Artificial Intelligence might easily go beyond our abilities and time budget, and may be disappointing for skilled players while being too hard for people only playing occasionally. In our game, we can easily meet this requirement by design.

- The game shall be **fast, chaotic, colorful and exciting**. The individual play rounds shall not take too long. For a party game, where many people might wait for their turn to play, it is crucial to define a round-based gameplay which allows for frequent switching. The players should not need a long introduction of how to

control or win the game.

- The game shall be **fun for both the winner and the losers**, or in other words: Playing the game shall be fun enough that winning becomes secondary.
- We define a **simple ground idea which is fun by itself**, mixed with high speeds and chaotic settings as well as interesting and challenging additions like gravity effects and extras. With this, we can precisely control the challenge the game poses to the players, and with it, the players' experience.
This makes our development clearer and more modular, in the end helping us to successfully finish this project.
- The **influence of gravity, nice graphical effects, destructible environments** and many **easy-to-use items** shall set our game apart from usual space shooters and multiplayer last-one-standing games. By foregoing the additional degree of freedom a 3D game would give, we open the door for casual players. By carefully designing visual effects, we aim to reach a wide range of people having fun with our game.

Gameplay Elements

The World

Each of up to four players controls his space ship in a 2.5-dimensional world containing multiple randomly placed planets and similar celestial bodies. Each of those exposes gravity effects to its surrounding, manipulating the movement of other objects. Planets may be damaged and destroyed by player weapons such as rockets. Bombs attracted by a planet can easily blow a huge hole in its surface. Explosions can break off debris from the planet, which in turn can damage space ships or planets they hit and possess their own centres of gravity. The whole setting is constructed such that some planets are not much bigger than the space ships themselves.

According to this physics model, planets are solid bodies (at least most of them) that exert a force onto the player spaceships. Therefore it is possible for players to both crash or land (if the speed of the player is low enough) on the planet surface. Landing however is rather a consequence of the physics simulation than a central gameplay element. Moreover, debris moving with high speeds may damage the player's ships.

To keep players from exiting the playing field, we implement some boundaries. Because intentionally leaving the area might be a tactical choice by the players, it is an issue of balancing if these boundaries must be hard (i.e. players are never

allowed to leave the screen, and instead collide with an invisible barrier) or soft (i.e. players are warned and must re-enter the area or their ships will be destroyed after 5 seconds).

For this reason, we shall implement multiple such boundary handling methods, and decide between them based on feedback from playtesting. Amongst the different methods are:

- Players must remain within the boundary through hard constraints
- Players can leave the area, but are pushed back by a strong force
- Players can leave the area, but take constant damage outside
 - The player is visually altered to make this rule clear
- Players can leave the area, but explode after a certain time outside
 - We might show the “outside” with special colors (toxin / military exclusion zone?) to make clear where the playing field ends

Players & Ships

Each player can control his/her ship freely, as long as the influence of gravity, damage on your ship and your opponents permit it. This means that fuel is unlimited, as we want to guarantee that players can always escape all planets' gravity.

For more variety, players will be able to select different ship types, which differ in properties such as their turning circle, speed and acceleration.

The players' ships have a limited base arsenal of weapons and abilities which can be extended by collecting items on the map.

This base arsenal may include:

- A simple laser front gun the shots of which are not affected by gravity. To better distinguish players, their shots should also be distinguishable, e.g. by their color.
- A shield for active defense: It is only active while the player presses a button. Activating the shield will protect the ship against almost everything, but at the same time it has a limited charge which depletes very fast and recharges only slowly. This requires the player to use the shield strategically.
- Simple rockets that do more damage than the laser gun but are affected by gravity.

Player Controls

Each player navigates his spaceship using the Xbox Controller. Navigating the space ship in this gravity influenced environment poses a non-negligible challenge to the player. Therefore we propose different control schemes that will be implemented and evaluated during early playtesting:

- The orientation of the first control stick defines the direction of acceleration, its tilt defines the strength of acceleration and maximal velocity. Weapons are fired in the direction the ship faces.
- The orientation of the first control stick defines the direction of acceleration, its tilt defines the strength of acceleration and maximal velocity. The second control stick defines the aiming direction of the weapons.
- The first control stick only defines the orientation of the space ship (and thus the direction of acceleration and aiming). A button is used to accelerate the ship.

In all cases, an evaluation whether the ship's rotation happens directly, or becomes too sluggish, will be necessary. Although the setting is in space, rotation and speed are restricted through friction forces to add intuitiveness to the gameplay.

To use items, switch items or trigger events, several buttons on the game pad are used in an intuitive way.

Game Modes

A game round starts with all participating players placed on the battlefield. Every player has a base arsenal of weapons to his disposal. Multiple planets, items and possibly other elements are placed on the field. Each player is required to navigate through the world, collect items and shoot his opponents. There may be several game modes which define different victory conditions. Once a round is finished, the players can change their settings and/or spaceships, and can start a new round.

For all modes listed below, there is a general distinction to be made beforehand:

One idea is that players only get one life and each round lasts only for a certain period (e.g. 60 seconds) - thus, players who have died will never have to wait for their next go for more than 60 seconds.

Alternatively, a round lasts much longer (e.g. five minutes) and players get multiple lives. This means that all players will be able to play for longer periods at a time, but also have to put down the controller for a longer duration after losing.

We don't see this as a core design decision, but rather an easily remediable balancing issue, which shall be addressed during playtesting.

The possible game modes include:

- Last Man Standing: Send your opponents to space-ship heaven while they try to do the same thing to you. There is a fixed number of respawns (possibly zero), which can be changed in the settings.
- Combine the relic: Collect all shards of a relic to win the game. Destroying a player will free all shards collected by him and the destroyed player will respawn after a short time.

Items & Weapons

Labeled boxes randomly appear on the playing field. Some of them will be affected by the global gravity field, and will eventually fall onto a planet's surface. At any time they can be collected by the players' ships, providing new weapons and abilities.

Players will be able to select which items can appear during the game. This way, restricting the set of items makes it easier for new players to jump into the game, and selecting a larger set of items gives advanced players more variety. There could also be presets, e.g. "Basic", "Normal" and "Chaotic" to choose from.

To allow for advanced balancing, some item types might be unaffected by gravitation, and thus easier to pick up than others. Playtesting will show if this gets too confusing or unfair.

A selection of possible items could be:

- Dimensional jump: The player performs a sudden jump to a position further ahead, ignoring any obstacles in its way.
- Big, Bad Laser: A weapon that needs to be charged. When released, a bright ray destroys everything in its way. For tactical reasons and improved realism, this (and possibly other "heavy" weapons) might cause a significant amount of recoil to the player's ship.
- Homing rockets: Follow the nearest opponent but have limited fuel.
- Bombs / Mines: Affected by gravity, cause large scale explosions that also affect planets.
- Cluster rockets: First press fires a rocket, second press splits them into many clusters, third press detonates all the clusters at once.
- Black Holes: Artificial centers of strong gravity. Destroy everything that reaches their center.
- Repair packs: Refill the player's health.
- Gun Turrets: Are dropped like bombs and fall towards the nearest planet. There, they assemble themselves and begin supporting the player by firing upon his enemies.
- Main gun upgrade: Temporarily improves your simple laser gun. This may increase projectile speed, rate of fire, spread or other properties.

Various

- It might be necessary / useful to have fine particles (star dust) that indicate the local gravity fields.

Game settings & Balancing

To ensure both maximum fairness in all matches, starting from early playtesting, and maximum variety for players in later stages, matches have various properties, which

can be balanced on a purely mathematical level. These settings include

- which item types appear
- the frequency at which items appear
- the maximum number of hit points per player and
- the maximum number of lives per player

Not all of the mentioned features might be implemented as stated, or make it to the final release, and we will choose an appropriate order of implementation.

1.2 “Big Idea” Bullseye

see title page

1.3 Sketches & Artwork

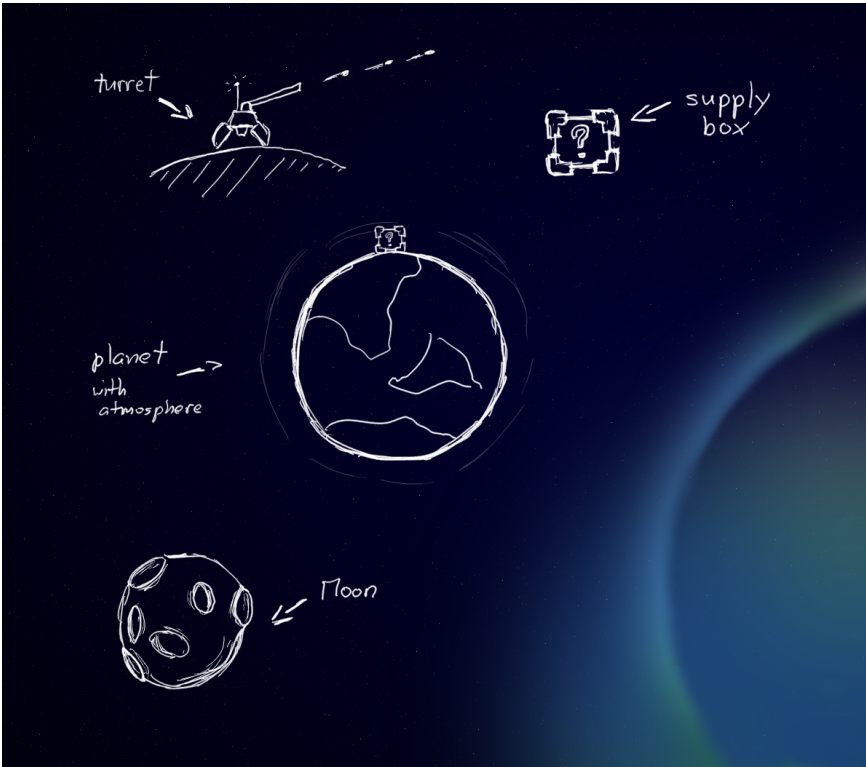


Fig. 1.1: Two example planets, supply box prototype and an autonomous turret on a planet's surface.

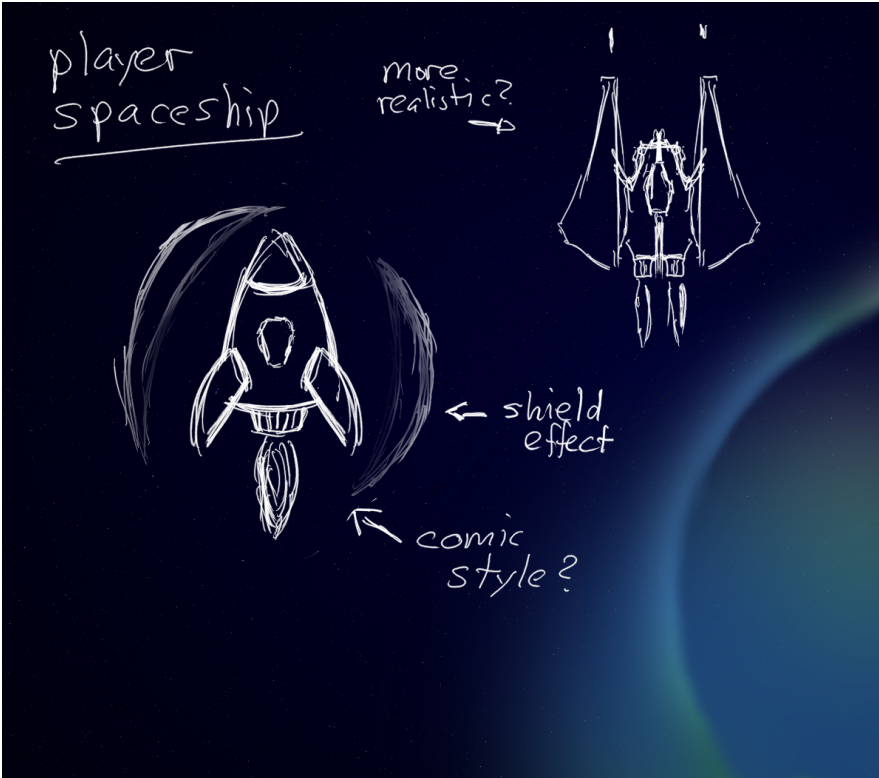


Fig 1.2: Two completely different space ship designs. One more realistic, the other comic-style

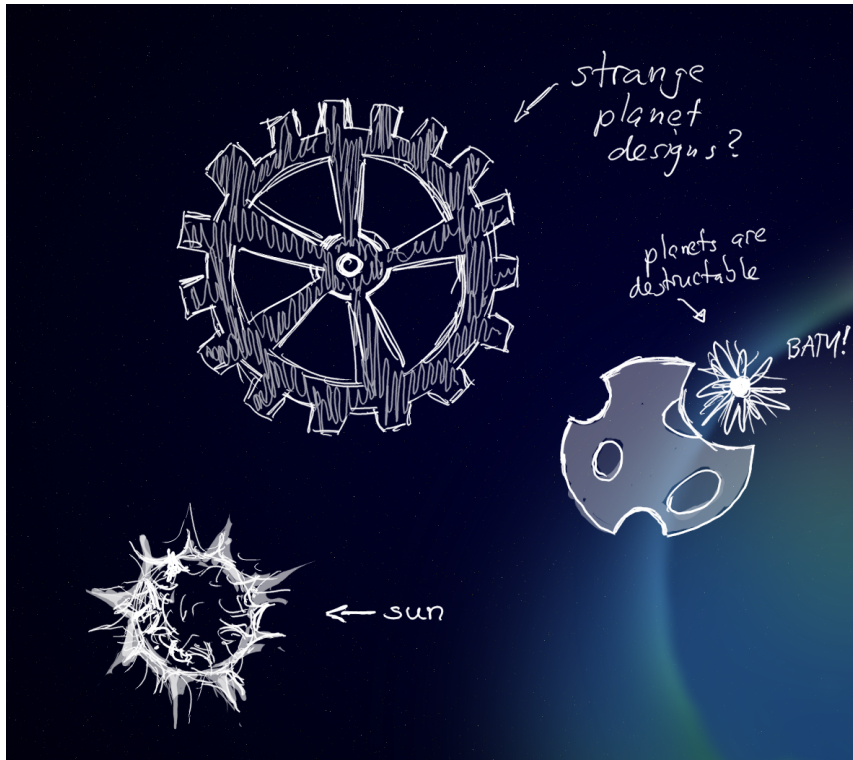


Fig 1.3: Variants for planet design and planet destructibility explained.

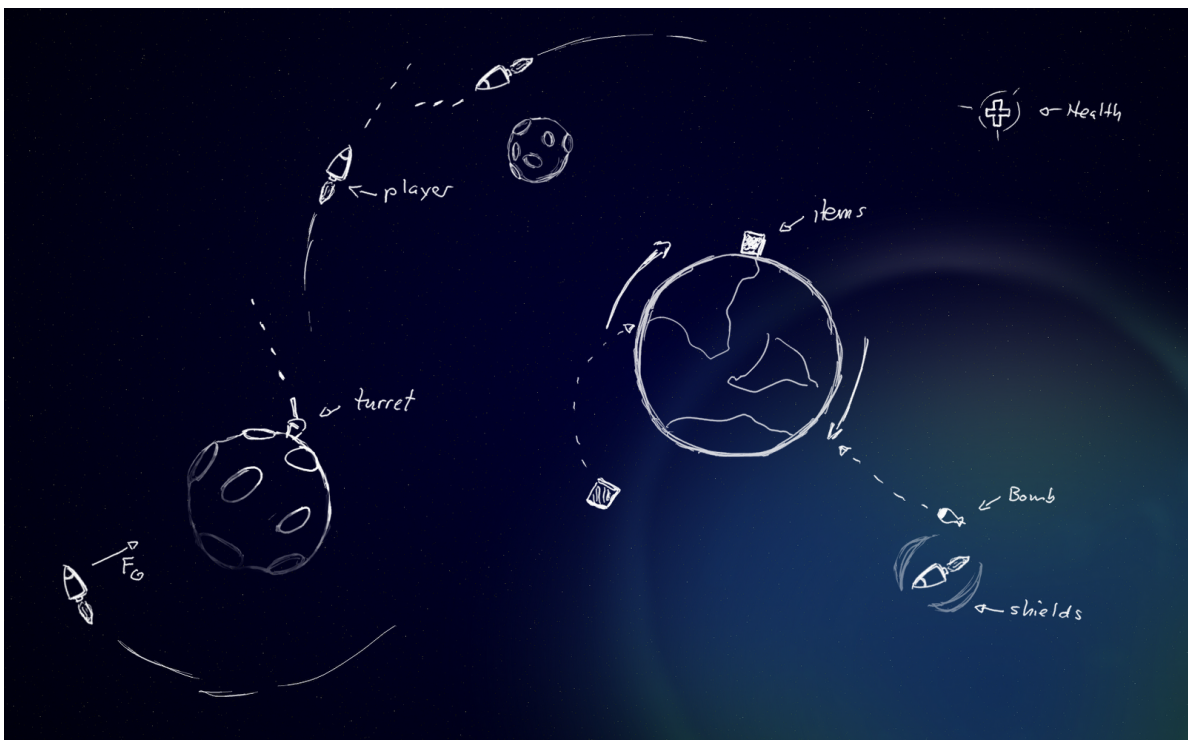


Fig 1.4: Sketch of a gameplay situation with various mechanics annotated.

1.4 Development Schedule

Total time effort is given in man hours.

Meta

Task	associated people	time effort
Idea Brainstorming	all	20.0h
Proposal draft meetings	all	12.0h
Proposal draft document	all	6.0h
Idea pitch presentation	all	4.0h
Final proposal meetings	all	12.0h
Final proposal document	all	3.0h
Physical prototype	all	15.0h
Proposal & prototype presentation	all	4.0h
Demo presentation	all	4.0h
Interim report document	all	9.0h
Interim report demo/presentation	all	4.0h
Alpha release document	all	6.0h
Alpha release demo/presentation	all	4.0h
Playtest document	all	6.0h
Playtest presentations	all	4.0h
Conclusion document	all	6.0h
Final public presentation	all	12.0h
Basic trailer & demo video editing	rms	6.0h

Preparation tasks

Task	associated people	time effort
Project Setup	dz	0.5h

Screen framework setup	dz	8.0h
Study of existing physics engines	rms	2.0h
Study of particle engines	dz	1.0h
Study of XNA examples	all	8.0h
Xbox360 Setup	rms	1.0h
Visual Studio & XNA Setup	all	4.5h
Sketches & Artwork	ab	4.0h

Functional minimum

Task	associated people	time effort
Gravity engine	dz	2.0h
Spacial data structures	dz	2.0h
User Controls	ab	3.0h
Static camera	ab	1.0h
Collision detection & response in 2D, planets as polygons	rms	4.0h
Laser Gun & Rockets	ab	2.0h
Placeholder graphics	dz	3.0h
Hard-coded map	rms	2.0h
Early playtesting	all	2.0h

Low target

Task	associated people	time effort
Moving camera	rms	3.0h
Visual backdrop	dz	3.0h
Active Shield System	ab	3.0h
Nice game menu	dz	5.0h
Item system & few items	rms, ab	7.0h
UI with health bars etc.	ab	3.0h

Desirable target

Task	associated people	time effort
Basic particle effects	rms	6.0h
Advanced shaders/lighting	ab	12.0h
Basic sound effects & music	rms	6.0h
Improved user controls	ab	2.0h
More items	all	8.0h
Random map generation	rms	6.0h
Destructible planets	dz	15.0h

High target

Task	associated people	time effort
High quality 3D models and 2D sprites	ab, dz	10.0h
Fancy shaders	ab, dz	12.0h
Advanced particle effects	dz	5.0h
Large set of items	all	12.0h
Advanced sound effects	rms	4.0h
Advanced background music (multiple tracks)	rms	4.0h
Consistent art style	ab	3.0h

Extras

Task	associated people	time effort
Multiple (n) themed sets of planet art	all	$n \times 5.0h$
Non-player AI ships	all	20.0h
Map editor	all	15.0h

1.6 Timeline

Week 08		Idea brainstorming
Week 09		Concept art creation
Week 10	Game Idea Pitch Proposal Draft	
Week 11		
Week 12	Formal Proposal Physical Prototype	Preparation tasks
Week 13		
Week 14	First playable demo	Functional minimum
Week 15		
Week 16	Interim demos	Low target
Week 17		
Week 18		
Week 19	Alpha release	Desirable target
Week 20		
Week 21	Playtest student presentation	High target
Week 22	Final public presentations	
and then...		Extras

1.7 Tools and Technologies

- XNA 4.0
- Microsoft Visual Studio 2010
- C# .NET
- Xbox 360
- Mercurial

- Mercury particle engine
- Kdenlive
- Hydrogen / Ardour / Audacity / MuseScore

- GIMP
- Blender
- Pixologic Sculptris

- LaTeX/PGF/TikZ
- Google Docs