

SpaceJump:  
A Jump'n'Run game in space

*Maik Thielen and Michael Zimmermann*

# Index

- [1\) Project goals](#)
- [2\) Story](#)
- [3\) HUD](#)
  - [3.1\) lives](#)
  - [3.2\) health](#)
  - [3.3\) collectable count](#)
- [4\) Player](#)
  - [4.1\) melee attack](#)
  - [4.2\) laser attack](#)
  - [4.3\) implementation](#)
- [5\) Level: Introduction](#)
- [6\) Level: Ice](#)
  - [6.1\) concept](#)
  - [6.2\) enemies](#)
    - [6.3.1\) seeker](#)
    - [6.3.2\) Wolf](#)
    - [6.3.3 Ice Mage](#)
  - [6.4\) objects](#)
    - [6.4.1\) energy package](#)
    - [6.4.2\) ice spikes](#)
    - [6.4.3\) crystals](#)
    - [6.4.4\) destroyable ice blocks](#)
    - [6.4.5\) health kit](#)
- [7\) Level: space](#)
  - [7.1\) concept](#)
  - [7.2\) asteroids](#)
  - [7.3\) enemies](#)
    - [7.3.1\) Seeker](#)
    - [7.3.2\) Golem](#)
  - [7.4\) objects](#)
    - [7.4.1\) replacement parts](#)
    - [7.4.2\) space mines](#)
    - [7.4.3\) health kit](#)
- [8\) cinematic scenes](#)
  - [8.1\) Intro](#)
  - [8.2\) rocket crash](#)
  - [8.3\) final](#)

## 9) Problems we've encountered

9.1) Unity editor

9.2) git commits

9.3) Ground detection

## 10) Teamwork

10.1) Source Code Management (SCM)

10.2) Conflict resolution

10.3) organization related documents

10.4) communication

## 11) References

11.1) assets

11.2) Unity information

# 1) Project goals

The goal of this project is to develop a 2D Jump'n'Run game using the Unity game engine.

# 2) Story

The player's name is from a more advanced civilization. He's a scientist and explorer and he build a rocket with a hyperdriver so he can visit very distant galaxies.

The only problem is that his home planet doesn't have enough energy resources to power the hyperdrive.

That's why he has to visit one of the nearby planets first to collect a lot of energy crystals which he can use to power the hyperdrive. This part of the story plays in the ice level.

It's a cold world full of ice, monsters and energy crystals.

Once he has enough energy he starts his rocket to get off the planet and prepares the hyperdrive for the journey.

Unfortunately, in the exact same moment where he wants to start the engine, a asteroid hits the rocket. Most things are still working but the hyperdrive got damaged and he doesn't have the replacement parts to repair it.

Luckily, there's a nearby asteroid belt with tons of space debris where he'll probably find everything he need to repair his rocket.

This part of the story plays in the space level. It's very silent and there are mostly asteroids flying around. There's also a huge golem made of stone waiting for him at the end of the level.

Once he defeated him, he can finally get back to his rocket, repair the hyperdrive and start his journey to explore worlds in distant galaxies.

### 3) HUD

The HUD is the interface that's rendered on top of the camera and thus always visible.

#### 3.1) lives

This is a small heart icon with a number in it and represents the number of lives the player has left until the game is over.

#### 3.2) health

This is the health bar which represents the health per life. Once it's empty, once live will be subtracted (if greater than 0) and the health bar will be refilled. If there is no live and health left, the player dies and the game is over.

This count can get increased by collecting a health pack.

#### 3.3) collectable count

The player must collect energy in the ice level and replacement parts in the space level.

The collectable information tells the player how many he collected from these and how many are left. It's just a small icon and a text in the form '1/10'.

## 4) Player

The player looks like a trooper which basically is a black space suit. He can walk, jump and has two two different attacks

### 4.1) melee attack

This is just a simple punch and makes minor damage only. It's enough to kill seekers and to destroy ice blocks.

### 4.2) laser attack

This is a powerful weapon. He can fire it once per second only so it can't be used too often. It makes much more damage than a simple punch and can be used to kill big enemies like the golem.

### 4.3) implementation

The player has 3 circle colliders, for his head, body and feet to make collisions as realistic as possible without too much computation overhead.

There's also a small box collider with the trigger-property enabled right in front of the player which is used to detect if a enemy got hit during a melee attack.

Jumping is realized by adding a small force up. This force will automatically decrease shortly because the player is controlled by the physics engine which pulls him down because of gravity.

The player object is also closely connected to the HUD so it can be updated when his health changes.

The ground detection works by using a small circle to ask the physics engine if any object that's part of the ground layer intersects with this circle. If so, the player's probably standing on solid ground.

## 5) Level: Introduction

This level takes place after a short cinematic scene and before the player starts flying to the ice planet. He's on his green home planet and has to walk to his rocket.

It's a good chance for the player to get familiar with the controls.

Once he reaches the rocket he goes inside, the rocket starts and automatically flies to the ice planet where he can find the energy crystals he needs.

## 6) Level: Ice

### 6.1) concept

The first level has a winter / ice setting. The mission of this level is to collect the energy the player needs to get the hyper drive working. There are 6 energy blocks which need to be collected in total.

To collect them, the player has to avoid falling ice crystals / ice spikes, survive lakes by jumping from tile to tile and survive the ice cave inside the level. Some enemies are included and try to preclude the player from collecting the energy.

### 6.2) enemies

#### 6.3.1) seeker

The seeker probably is the simplest enemy of the whole game. It's just a small monster which dies after one hit and also gives the player damage on contact.

It is able to either walk or just stand in one place. He's also able to walk between two points like the golem so he doesn't fall off the ground and die.

#### 6.3.2) Wolf

A very aggressive opponent always moving around and looking for some prey. The wolf has a high running speed which makes him very dangerous. As soon as it detects something, it will start to chase it immediately by running even faster. If the target is in attack rage, the wolf starts to attack it by using its big claws.

### 6.3.3 Ice Mage

The ice mage is the most interesting enemy inside the ice level. He does not move around the world and just stands still and waits for the player to enter his attack range.

If the player is inside the attack range of the ice mage, he uses the power of ice magic to attack the player by casting ice projectiles in the direction of the player. These projectiles can be destroyed with the player's laser weapon.

## 6.4) objects

### 6.4.1) energy package

Energy is used to get the hyper drive working. The energy package is a collectable object which looks like a small coin with a small thunderbolt image inside. Once collected, the counter inside the HUD will be increased by 1.

### 6.4.2) ice spikes

Small hurting ice spike which will inflict damage to the player while he is in contact with it. Used to limitate the moving space of the player in some situations so the player has to avoid those ice spikes to prevent receiving damage.

### 6.4.3) crystals

Spiky ice crystal which is supposed to surprise the player by falling down the sky. This happens as soon as the player reaches the event hitbox. If the ice block hit the player, the player receives some damage and the crystal object will be destroyed.

### 6.4.4) destroyable ice blocks

Small ice blocks, which can be destroyed by the player with his laser attack.

### 6.4.5) health kit

A first aid object which will increase the current health points of the player by a given amount. If the player collects it, the heal will take effect and the health kit object will be destroyed.



## 7) Level: space

### 7.1) concept

This level has a space setting. That means that there is no large, solid ground. Instead you have to walk and jump between floating asteroids.

Some asteroids are standing still, some are moving and some are rotating.

This also means that the player will die when he'll fall too far down where he'd never be able to get back up to another asteroid.

### 7.2) asteroids

Since the asteroids are weightless but still controlled by the physics engine, there are many ways to use them. We've added many configurable properties to them so they can be easily used in the unity editor.

They can be given an initial rotation speed and movement direction. They also can be configured to pause until they become visible for the first time. Also, they can be made reset themselves to the start position when they leave the screen. This is useful for making asteroids continuously come in from the top and leave the screen at the bottom.

This can be used as a moving platform target which is needed to get to the next safe and static asteroid.

Another thing worth mentioning is that the asteroids need a lot of mass so they can't be pushed away by the player.

### 7.3) enemies

#### 7.3.1) Seeker

This is the same enemy as in the ice level.

#### 7.3.2) Golem

The Golem is the biggest enemy of this level and waits at the end of the level. He's very big, strong, and gives the player a lot of damage.

His only attack is punching the player with his giant stone hands.

He walks back and forth on between two distant points on a range of close asteroids which form a ground. This works by placing trigger-collider objects at these two points and as soon as the

golem collides with them, he just turns around and continues walking in the opposite direction. He also turns around when the player touches his back.

By using a raycast he can detect if the player is range and once he is he starts his attack.

## 7.4) objects

### 7.4.1) replacement parts

These represent the replacement parts the player has to collect according to the story. To simplify things, they're just simple boxes which can be collected and which increase a counter in the HUD.

### 7.4.2) space mines

These are mines which explode when they get in contact with any other object. This includes the player, enemies and asteroids.

Both the player and enemies can get damage from a space mine explosion.

### 7.4.3) health kit

This is the same object as in the ice level.

## 8) cinematic scenes

### 8.1) Intro

This this only gives a short introduction into the game's story using cinematic text effects. The 'Introduction' scene comes right after this one.

### 8.2) rocket crash

This scene comes after the ice level. When the player may expect that the game has already ended, all out of the sudden an asteroid hits the rocket and the player gets a new mission.

### 8.3) final

This is the final scene after the the player has successfully completed the space level. You're flying through hyperspace with your rocket.

## 9) Problems we've encountered

### 9.1) Unity editor

The Unity editor can be very buggy and especially when files changed outside (using git), you sometimes have to remove all build files and reopen the project.

There also are some problems with case-sensitivity when using both Windows and Unix machines.

### 9.2) git commits

Unity likes to change some files seemingly random. Together with the fact that even small changes can produce huge changes in the project files it's sometimes hard to produce a clean git history with 'per-feature'- commits.

### 9.3) Ground detection

It's sometimes hard to do ground detection properly - especially when you want to be able to walk on enemies etc.

When not using tiled maps it can also happen that the player gets stuck in between two platforms, so you either have to write complicated (and buggy) ground-detection code you u just have to design your level more carefully so such things just can't happen anymore.

## 10) Teamwork

To work on such a project as a team there are a few things which need to be taken care of.

### 10.1) Source Code Management (SCM)

We used the SCM tool git to be able to work on the same project and at the same files at the same time. For this to work properly with unity, the project must be configured to store all project files in text form instead of the (default) binary form.

### 10.2) Conflict resolution

For resolving merge conflicts on the same files we used the tool 'SourceGear DiffMerge' which makes it very easy to handle these.

### 10.3) organization related documents

We've used Google Drive to store TODO-lists, documentations, and collection of assets which we may want to use in future.

### 10.4) communication

It's not always easy to be at the same place at the same time with computers capable of developing games. So the most used communication tools while developing this game was Google Hangouts in both text and voice form.

# 11) References

## 11.1) assets

<http://www.flashkit.com/loops/Ambient/Ambient/space-drifter-150305121007.html>

<http://www.1001fonts.com/sf-zero-gravity-font.html>

<http://www.1001fonts.com/insane-hours-2-font.html>

<http://clipartist.net/svg/moon-in-comic-style-super-duper-svg/>

<http://clipartist.net/svg/rocket-blue-and-red-black-white-line-art-svg/>

<http://answers.unity3d.com/questions/29183/2d-camera-smooth-follow.html>

<https://www.freesound.org/people/Arctura/sounds/39829/>

<http://answers.unity3d.com/questions/677914/cant-remove-gameobject-from-scene-in-editor-not-ap.html>

<http://opengameart.org/content/complete-spaceship-game-art-pack>

<https://www.freesound.org/people/rhodesmas/sounds/321723/>

<https://www.freesound.org/people/Apenguin73/sounds/335152/>

<https://www.freesound.org/people/jobro/sounds/35680/>

<https://www.freesound.org/people/qubodup/sounds/157609/>

<https://www.freesound.org/people/Greencouch/sounds/124905/>

<https://www.freesound.org/people/thatjeffcarter/sounds/165436/>

<https://openclipart.org/detail/17031/iconpuzzleyellow>

<https://openclipart.org/detail/228673/hyperspace>

## 11.2) Unity information

<https://docs.unity3d.com>

<https://unity3d.com/community>