|  |
| --- |
| Jesse Kaukonen |
| Captain! |
| Design document for "Captain!" |
|  |
| **Jesse Kaukonen** |
| **4/7/2011** |

|  |
| --- |
| This is the design document for the game project "Captain!" |

# Contents

[1 Contents 2](#_Toc293773776)

[2 About this document 3](#_Toc293773777)

[3 Battle mode 3](#_Toc293773778)

[3.1 The ship 3](#_Toc293773779)

[3.2 The point of the battle mode 3](#_Toc293773780)

[3.2.1 Combat 3](#_Toc293773781)

[3.3 Hull 3](#_Toc293773782)

[3.4 Rooms and components 3](#_Toc293773783)

[3.4.1 List of rooms with their components 4](#_Toc293773784)

[3.5 Damage 5](#_Toc293773785)

[3.6 Air 6](#_Toc293773786)

[3.7 Pressure 6](#_Toc293773787)

[4 Exploration / campaign mode 6](#_Toc293773788)

[4.1 Generation 6](#_Toc293773789)

[4.1.1 Pre-defined 6](#_Toc293773790)

[4.1.2 Random 7](#_Toc293773791)

[4.2 Story 7](#_Toc293773792)

[4.2.1 Protagonist 7](#_Toc293773793)

[5 Character mode 7](#_Toc293773794)

[6 On implementation 7](#_Toc293773795)

# About this document

This document explains the features and possible contents of the game "Captain!". It is aimed for only the developers.

# Battle mode

## The ship

The ship is the main focus of the game. Everything is based on the ship as a complex collection of components, personnel, weaponry and various variables. The ship doesn't have "hitpoints", nor will the game be considered lost when some set of modules is destroyed. The engines may be destroyed, but there are hard-working mechanics onboard who are able to fix things even during a battle, even if it is only duct tape. The only condition for losing is when either everyone is dead, every belligerent has retreated from the battlefield or all hostiles have come to a ceasefire / surrendered.

## The point of the battle mode

Ultimately, there could possibly be a campaign mode (see section 4, campaign), but in the battle mode the goal is usually to destroy the enemy ships, perform the activity in the area (colonize a planet, deliver a package, whatever) or engage some ship in diplomatic relations.

### Combat

The combat begins the moment a ship goes to "Red Alert". The ship personnel performs some set list of activities if the captain isn't on the bridge such as activates shields and returns fire. If the captain is on the bridge, then the captain can change any otherwise automatic orders. During combat, the ships fire at each other, people die and components break. The combat ends if one of the parties surrenders or retreats, if the ship is damaged so much that it can't operate or if one ship is boarded and taken over.

## Hull

The ship has a hull: A preset container for rooms and components. The hull has a material out of which it has been constructed, some variables such as weight and various numbers of resistance (against radiation, type-C beams etc). The hull is a set of squares that can't have anything inside them. They are solid metal, or some other substance, in which nothing can be built. The hull can't be multiple layers - those are armor.

## Rooms and components

Once a hull has been selected, rooms are placed on it. Room is at minimum 1x1 squares, but usually larger. Corridors are rooms, and so are the bridge, the engine room and the mess hall. Each room is defined by an area of squares which is designated during the planning of the ship. Once rooms are set, they are static. They can't change afterwards unless re-designed at the shipyard. Each room has some variables in them, such as air inside the room, pressure, crew inside the room, list of personnel inside, list of components inside, and air & energy consumption.

A component is a device that provides something to the ship. They must be placed inside a room. Some components must be adjacent to the hull (such as engines and torpedo launchers), while some can be located anywhere inside the ship. Each of the components has a set of functions, and some have requirements. An engine naturally gives some thrust to the ship, but also requires energy. A bed increases the amount of personnel you can have onboard, and sick bay makes it possible to heal personnel. A laser turret does what expected if it has enough energy.

### List of rooms with their components

#### Reactor core

Reactor: Provides energy to the ship. The energy does not require wiring inside the ship as it is assumed that all the walls have a jungle of cables going inside them. So if there is a reactor inside the ship, then all the squares are receiving power. Each square, however, has a variable defining how much energy is consumed inside that square each second. Size varies from 2x2 to 10x10

Coolant: If coolant is damaged, then the reactor will begin heating up, eventually either blowing up or shutting down. Size varies from 1x1 to 5x5.

Reactor control: If external communications are damaged, the reactor control can be used to shut down the reactor manually. Size 1x1.

#### Bridge

Captain's chair: The currently highest ranking officer is sitting on this chair and issuing orders. If the captain is elsewhere, then the ship is piloted by an "autopilot" according to the instructions set by the player. This means that if the captain is wounded in the sickbay, then the first officer will carry on with the mission. The chair will always be occupied by someone. Size 1x1.

Communications: Manages the communications. Size 1x1.

Weapons: Manages the weaponry. Size 1x1.

Management: Handles sealing areas, opening and locking doors, shutting down generators and other matters of this nature. Size 1x1.

Helm: Manages the engines and warp drive. Size 1x1.

#### Crew quarters

Bed: Increases the max size of personnel the ship can have. Size 1x1.

Pool table / video games / luxuries: Add morale. Size varies.

#### Sickbay

Medical cabinet: Stores medical stockpiles. Size 1x1

Freezer: Stores bodies for medical study / other purposes. Size 1x1.

Medical bed: Allows wounded personnel to heal. Size 1x1.

Operating table: Allows surgical operations. Size 1x1.

Medical scanner: Allows scanning. Size 1x1.

#### Engine room

Power relay: The engines are special components in the way they need an external device for routing the power, mostly because of the vast amounts of energy required for moving an object such massive as the ship. There is a requirement of one per engine. Size 1x1.

Engine: Needs to be placed adjacent to the hull. If there are two engines, then the ship will usually find a way to get anywhere, albeit slowly if the engines are located in a difficult fashion. Size and shape varies.

Warp drive: Enables warping. Size 3x3.

#### Shield room

Shield generator: Provides a shield to the ship. The larger the ship, the more power is required from the shield generator. The ship can only have one type of shield generator, but there may be many of the same type. Size varies.

Shield control: Can be used to manually manage the shield. Size 1x1.

#### Communications

Communications table: Enables communications to space. Size 1x1.

Internal communications: Handles the communications occurring inside the ship. Size 1x1.

Deep-space communications: Enables sending transmission to longer distances than the nearby space. Size 1x1.

#### Transporter room

Transfer control: Required for using the transporters. Size 1x1.

Transporter: Teleports personnel to locations inside or outside the ship. Size varies, and defines the amount of people it can transfer.

#### Weapons room

Torpedo tube: Allows launching a torpedo. Must be placed next to the ship's hull. Size 1x1.

Phaser battery: Allows firing a phaser beam. Must be placed next to the ship's hull. Size 1x1.

Mass driver: Allows firing projectiles. Must be placed next to the ship's hull. Size 1x1.

#### Storage

Ammunition box: Can contain torpedos and other main gun ammunition. Size 1x1.

Freeze box: Can contain food. Size 1x1.

Chest: Can contain various items. Size 1x1.

#### Corridor

Empty: Simply a connector between squares. By default, the entire ship is a big corridor. Size 1x1.

Door: Can be closed to shut down areas. Size 1x1.

## Damage

As the ship gets hit, damage happens. Damage can affect one or multiple rooms. As damage happens, the components inside the room lose health based on the proximity of the hit. As the component loses health, its probability for operating correctly changes. If a communications table is damaged to 70 points of health from 100, then it has a 70% chance of working.

Damage can usually be fixed by mechanics if there are materials onboard. Repairing a communications table from 70 hp to 100 hp takes some 30 minutes, while repairing an engine from 50 hp to 1000 hp takes days.

Damage is always added to squares. If the square has flammable material and the room has air inside, then a fire may occur depending on the type of damage. A fire can spread, and creates smoke. Fires can be put down by using fire extinguishers, or sealing the room and thus removing the air.

## Air

Each square has an air variable. This is means how much air there currently is at that square. There can be from 0 to 10 air. If the square has more than 0 air, then personnel can operate there and fires may occur. Each person consumes air in the square they are, but if the air production is operating, this isn't changed. Each person consumes 0.01 air per second. This means two people in 1 square in a sealed room of size 4 will consume the 40 available air in the room in 2000 seconds, which is 33.3 minutes. May need changing if the game is made more realistic.

## Pressure

Each square has a pressure variable. If the square is pressurized, personnel can operate there. Non-pressurized squares add a large amount of damage to the personnel inside.

# Exploration / campaign mode

During this mode the player generally controls the space ship. He can travel between stars, enter into battles, send messages to other ships and planets, collect resources from asteroids and do other more strategically important things. While in this mode he can also order changes to be made to the ship.

## Generation

### Pre-defined

The galaxy will be generated with a pre-defined seed. This means that all the games will generate the same galaxy, but it will happen as the game progresses. This means the hard drive requirements should go down quite a bit.

#### Galaxy generation process

* The random seed is read from the configuration file
* Using it, stars are created on the galaxy map. They all get name, mass, location, temperature and size. If the mass / size ratio passes Schwarzschild radius, then the star is made into a black hole. There should be only one or two of these.
* If there are black holes, modify the stars to twirl towards it
* Load the starting point from the configuration file
* Create the solar system at the starting point

#### Solar system generation process

* Check the amount of stars
* Create the stars and save their vertex, normal and texture data into memory. This data is not saved on hard drive - it is generated when the solar system is loaded
* Depending on the number of stars, calculate orbits around them
* Create the planets around the stars, from the closest to the sun to the furthest
* As each planet is generated, they all receive mass, type, size, gravity and number
* For each planet, generate a landscape bump map. As the game is all in 2D, it can simply be a bump map without anything fine
* For each planet, generate a colour map. This is based on the planet type and temperature
* For each planet, possibly add moons
* For each planet, add a possible civilization'

### Random

Don't use a pre-defined seed, but generate a new one. This will always produce a random galaxy.

## Story

### Protagonist

The player character is a lab tube born genius. At the beginning of the game there is a tutorial section where the player wakes up from the tube and fights his way out from the laboratory facility, possibly a space station. He is instructed by some mastermind professor that wishes to release him to the wild galaxy for some ultimate reason, but the player simply uses this open door for escaping, without knowing who he is or where he is. He has received training in weapons, strategy and various technologies in the facility. He escapes in a one person fighter, and the introduction ends.

Years later he is a newly arriving captain on a Coalition vessel. Once again, in this character-mode he moves through a space station into a shuttle in the orbit of some planet and gets taken to his ship-to-be-commanded. There he has open hands to meet the crew and do whatever he wants. At this point the player is completely clueless on what he is supposed to be doing, and can simply order the ship to jump out from the star system.

Some minutes later the plot begins and the ship begins its long journey, whatever that may be.

# Character mode

This is the default mode. While in this mode, the player can move his actual character wherever he wishes, usually somewhere on the ship. He can also chat with other characters and interact with the surroundings. In battle situations he can participate on defending the ship from boarding parties / participate in occupying enemy vessels / space stations and in general play as 2D top-down shooters work. Some planets and space stations can also be visited.

# On implementation

The graphics are only 2D to eliminate one dimension of complexity. The best way is probably to go with OpenGL (+SDL?) for the ease of programming and multiplatform. No pre-made engines unless required for maximum learning experience. Physics should prove to be a fun puzzle, and the design might change depending on the outcome of the battle with the physics.

The battle arena is probably filled with empty space. There are some 1000 - 10 000 squares inside the ship, and inside each square there are components, humans and other things of note. This means that there will be quite a load of variables changing for each frame, so perhaps some kind of list for "updated objects" is required. Graphics will probably be simplified first with the ship showing the floor plan with nothing moving inside. The humans and other stuff can be added later.

The squares inside the ship could probably be defined relative to the hull, not screen borders. Each square could either have the X and Y coordinates on the screen, or they might simply be calculated by a simple algorithm, or they might be in a 2-dimensional array. Anyway, they need to be set in such fashion that they are always static in relation to the ship, but so that the squares can change their appearance (damage makes them black or something). It might be reasonable to build the ship from a set of flat 2D planes with a texture on them as this way rendering the geometry is fast. In future, the third dimension could be added for the look (2.5D), and the ships could simply move in XY-coordinate system.

For the starters it might be good to ignore any thoughts on the campaign / exploration mode. First there should simply be the battle mode where the ship can move around, then maybe try some collision tests, then try getting some of the module code in.