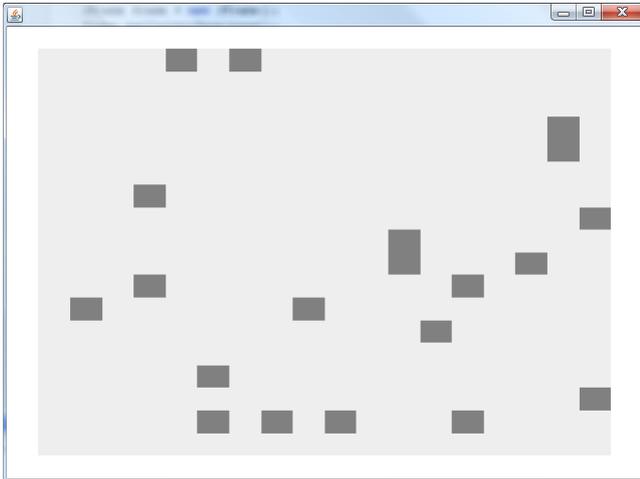


# Ufohippa 3 Developers Diary

---

See Post Mortem in the end of this document

**21.8.2009**

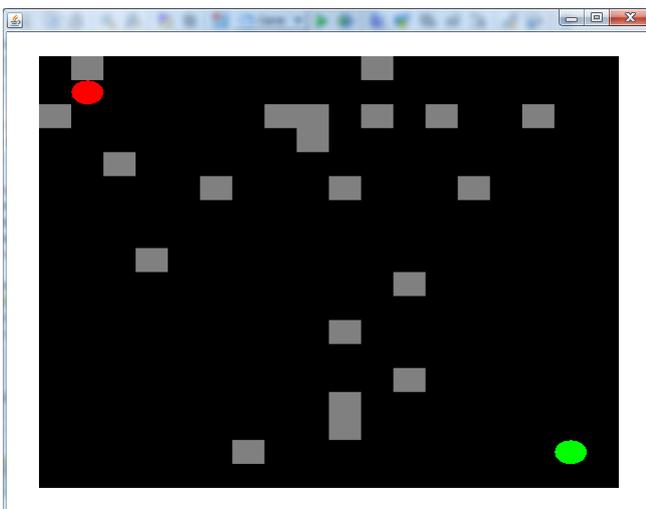


Started implementation

- Initial IntelliJ IDEA project set-up and general configuration
- Implemented basic classes for levels, players and game
- Created a very simple reference renderer with Java2D

The game is now able to render a randomly generated simple level on the screen with different tile types.

**29.8.2009**

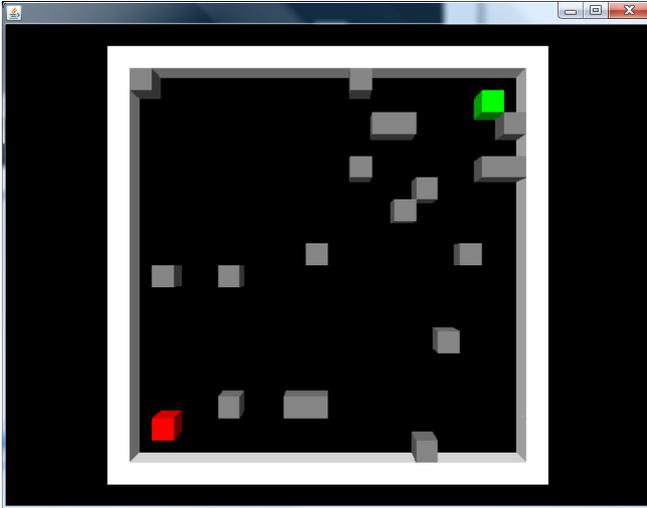


Added OpenGL rendering

- Installed JOGL libraries

- Implemented OpenGLRenderer in addition to Java2DRenderer.
- Started working with meshes and vertex buffer objects

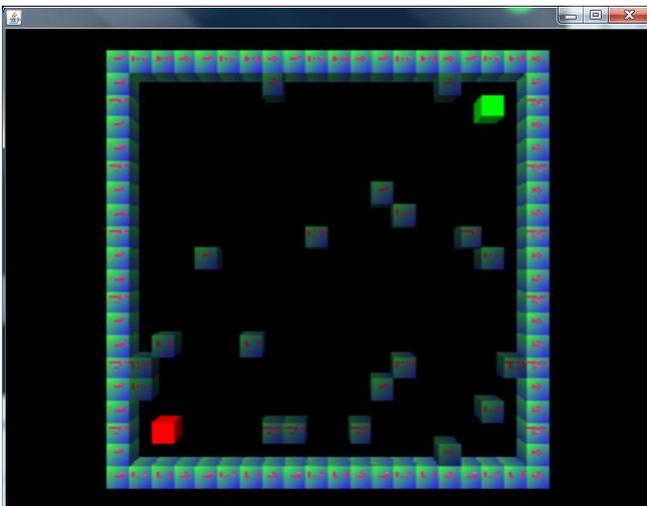
30.8.2009



Improved OpenGL rendering

- Render tiles as cubes
- Added lighting
- VBOs not yet working

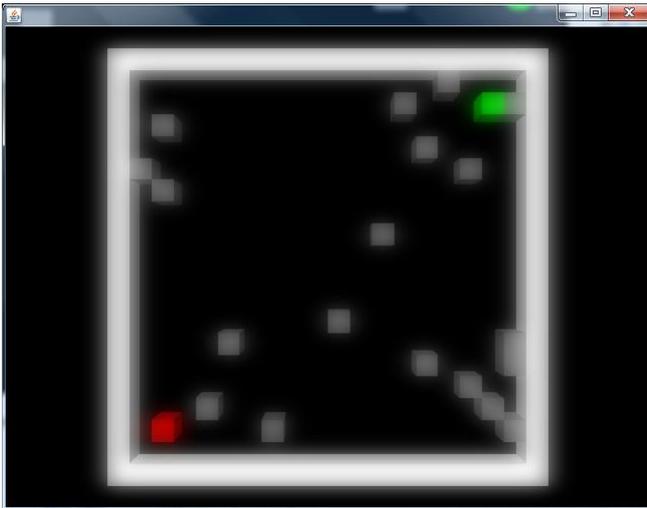
6.9.2009



Texturing support and off-screen rendering

- VBOs are now completely working
- Improved VBO data model to support normals and texture coordinates
- Implemented texturing
- Implemented multiple render passes and off-screen rendering with FBOs

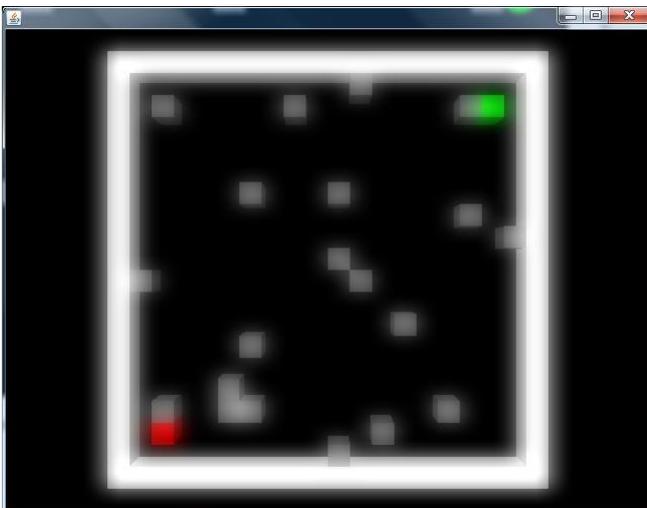
10.9.2009



Implemented shader support and bloom shaders

- GLSL shader support done
- Implemented the following shaders: Horizontal blur, vertical blur and blend. These three are the basis for Bloom effect

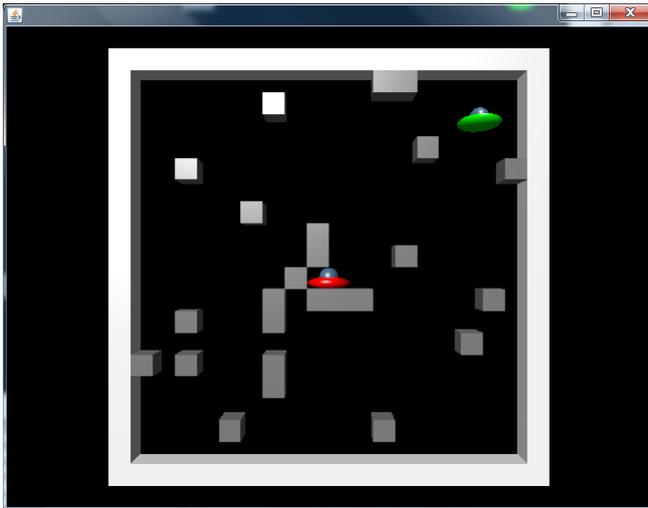
13.9.2009



Improved bloom shader

- Blur shader is now generated with configurable kernel size
- Support for Gaussian and box blur
- Proper weighting to normalize result

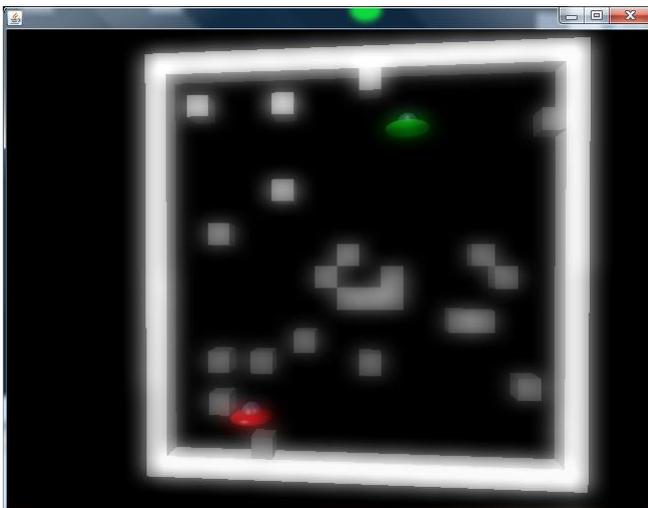
20.9.2009



Ufo model

- Created ufo generator consisting of one ellipsoid for hull and two hemispheres for dome
- Per pixel phong shader

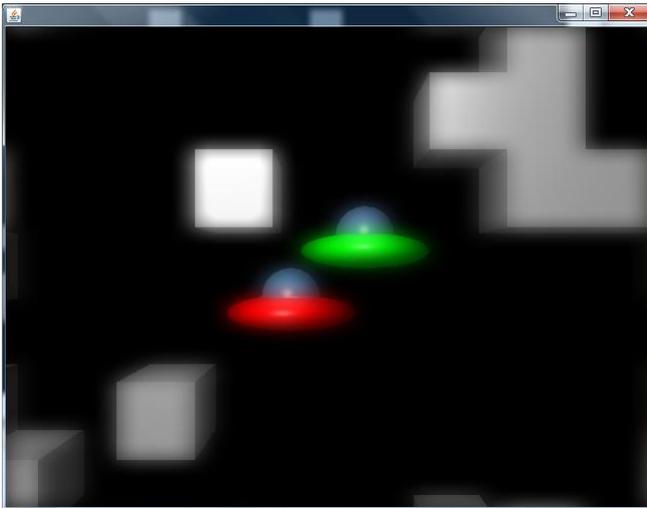
24.9.2009



Game loop

- Started creating client-server separation
- Implemented game loop with frame skipping
- Implemented keyboard handling for two players

**1.10.2009**



Collision detection

- Working collision detecting with tiles
- Working collision detecting between ufos

**8.10.2009**

```
Started game server  
SERVER: Client Superkupla:60690(0) state changed to IDLE  
CLIENT: State changed to IDLE
```

Network support

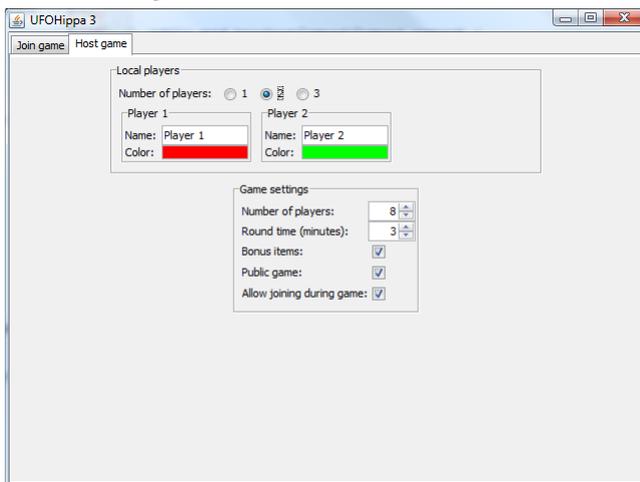
- Created initial support for network communications
- Implemented two-way HELLO-message

**15.10.2009**

Improved network architecture

- Implemented hierarchical callback-based state machine
- Re-implemented two-way HELLO-message on top of new architecture
- Created thread-aware logger

21.10.2009



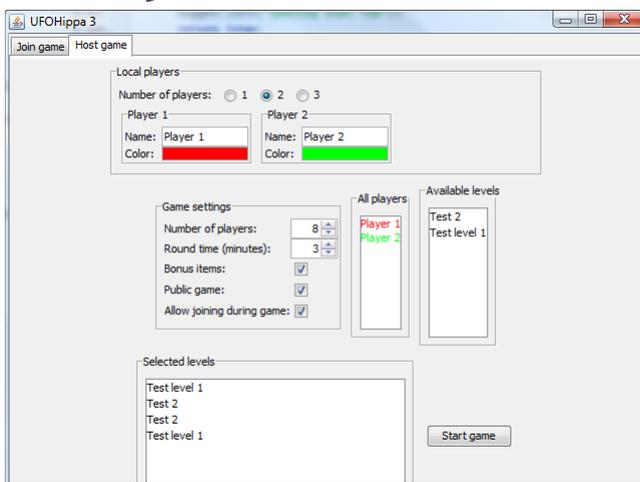
## Lobby

- Graphical user interface for lobby
- Network support improvements
- New network commands for player settings

5.11.2009

Improved network architecture, no visible progress

12.11.2009



## FMOD integration with JNA

- Integrated FMOD sound library with JNA
- Created high-level sound manager class for playing music and sound effects
- Added simple level selection

### 19.11.2009

Added support for sound effects and copied some of the sound effects from old UFOHippa.

### 3.12.2009

Game settings synchronization

- All relevant game settings such as playing time, players and levels are now synchronized to clients
- Start game button tells server and all clients that a game is about to start.

### 11.12.2009

State synchronization, no visible progress

- Server side client handler states can now be synchronized to wait other client handlers

### 17.12.2009

Sending levels and starting game

- Levels are now sent over network and received by client
- Client enters full screen game mode before starting to receive levels

### 7.1.2010



Loading screen

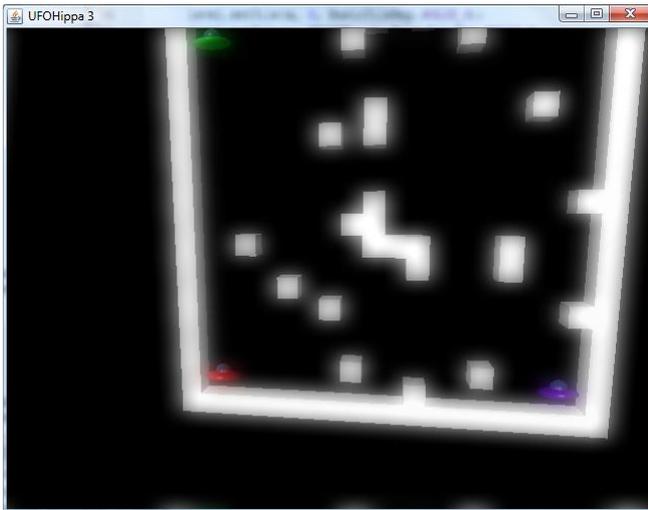
- Level sending progress is now indicated with an OpenGL progress bar
- All texts are rendered with JOGL TextRenderer

## 14.1.2010

Improved match starting

- Simplified architecture on match starting
- Implemented network protocol and client side handling for starting rounds

## 21.1.2010



Round starting with proper level setup

- Renderer is now able to render the level started by server
- Added starting positions to network protocol

## 28.1.2010

Keyboard input

- Keyboard events are now converted to player input actions and sent to the server
- Separated shared server context for all server states to use

## 4.2.2010

Game state propagating to clients

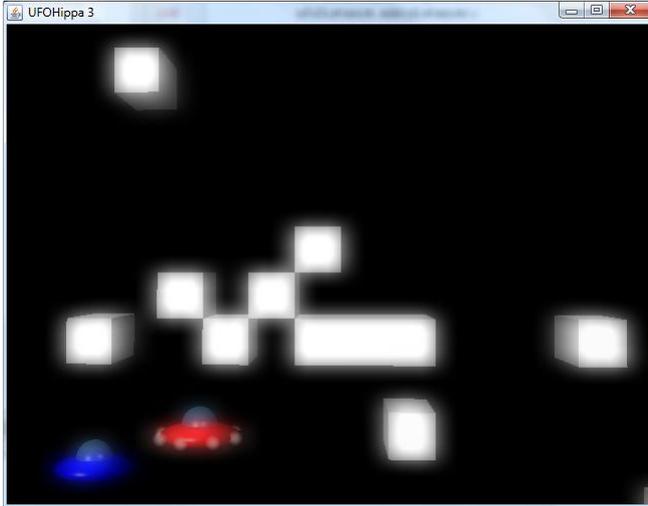
- Game is now playable through network
- All UFO locations are transmitted through network in real-time
- Clients are also updating the locations to predict the server state

## 11.2.2010

### Improved realtime network updating

- Fixed a bug which caused update packets to be sent very late
- Added logic for keeping track of current tag

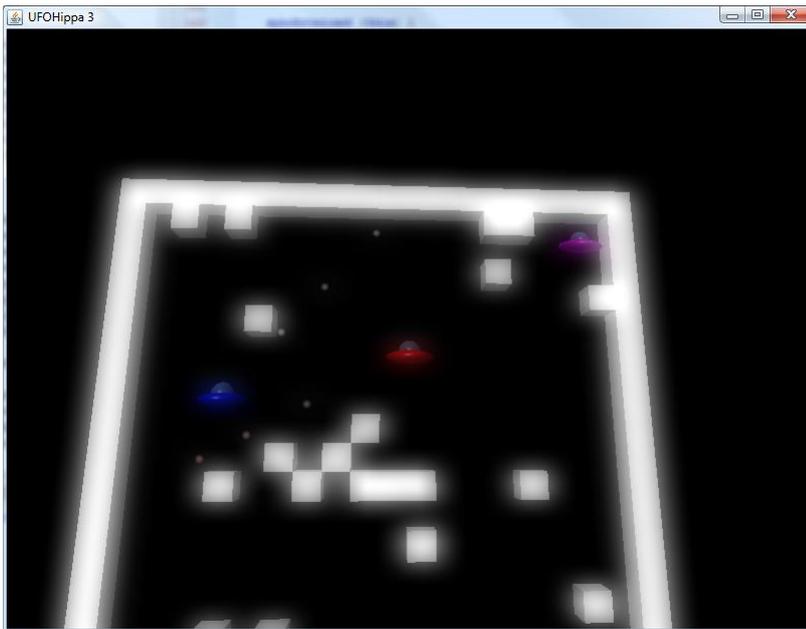
## 15.2.2010



### Tag visualization

- Implemented GeneratedTexture and FlareTexture
- Created blinking and rotating lights for tag ufo
- Added safety time for tag

**25.2.2010**



Shooting

- Implemented shooting for both server and client side
- Renders bullets as flares with the same color than the originating ufo
- Does not handle bullet collisions yet
- Integrated sound system to the game
- Improved keyboard handling to support non-player actions and work without focus

**4.3.2010**

Bullet bouncing

- Implemented bullets collision checking against tiles with probability based bouncing
- Debugged FModEx sound system. Sounds fail to play after hardware channels run out, fix by calling System::update

**14.3.2010**

Fixed FModEx problems which were located last time, no visible progress

- All sound system commands are now synchronized
- Sound system is updated regularly in its own thread

**19.3.2010**

Server to client synchronization debugging and fixing, no visible progress

- Merged all real-time update commands to single UpdateCommand
- Fixed some issues with frame skipping, but some are still left

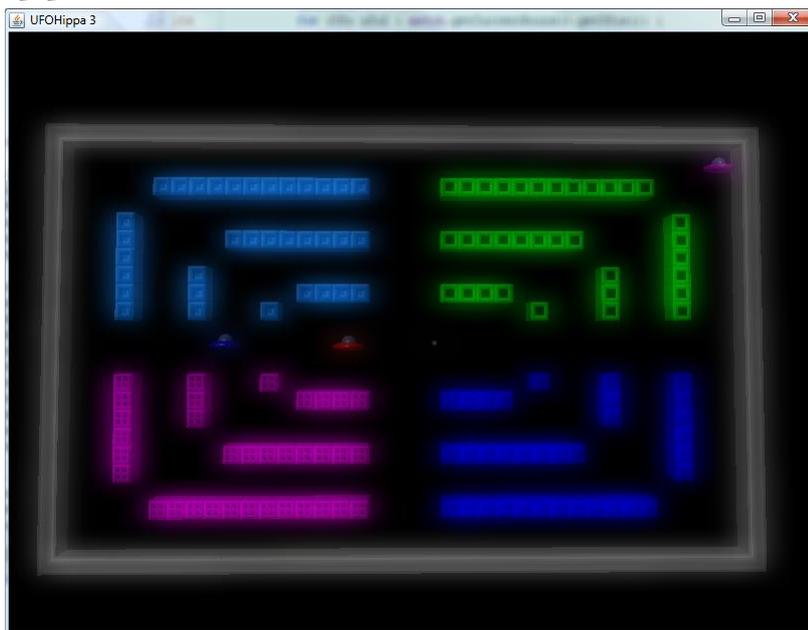
20.3.2010



#### Levels

- Created simple specification for XML based level format
- Implemented loading of levels from disk
- Implemented loading of tile sets from resources
- Converted old tile images from UFOHippa 2 to new format (not yet used)
- Created a level converter for converting levels from UFOHippa 2, and converted one level

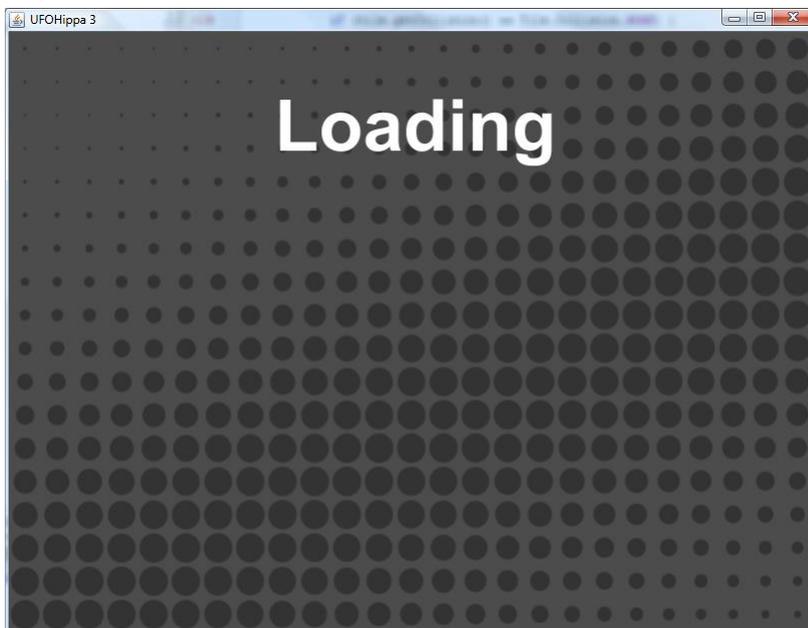
25.3.2010



Levels

- Converted all old levels
- Added tile texturing using old graphics

2.4.2010



Resource loading

- Implemented resource loader, which loads resources on the background and finalizes them with GL context
- Created new shader for rendering loading screen background animation

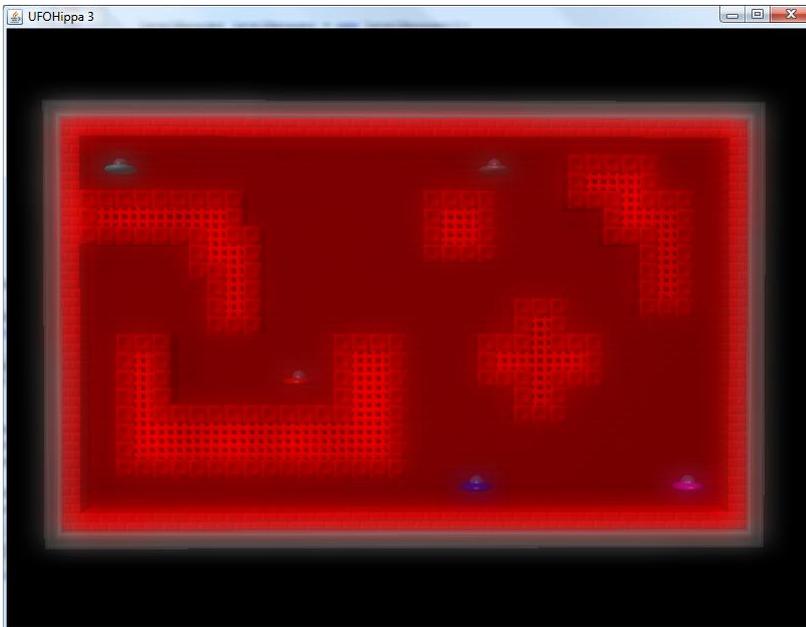
8.4.2010



Background tiles

- Implemented proper loading of tiles in background
- Converted old background tiles in addition to foreground tiles

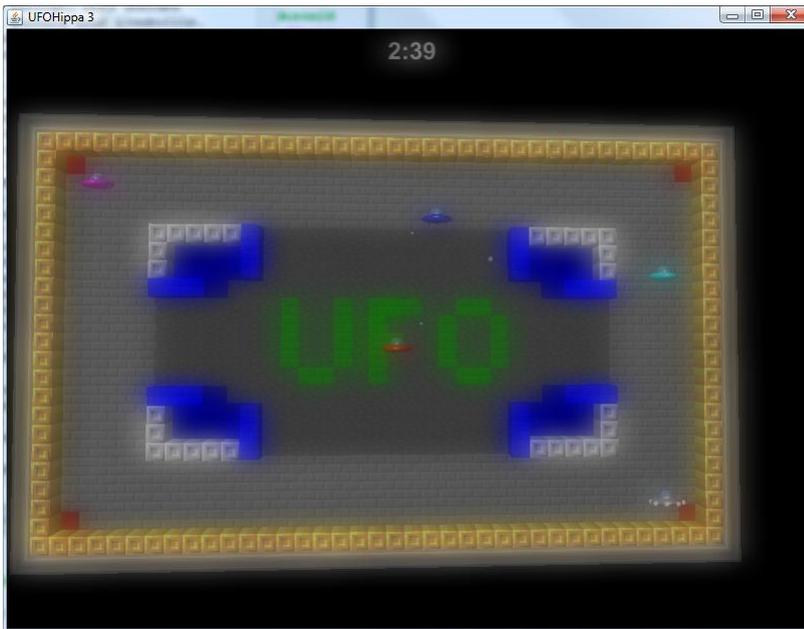
13.4.2010



Logging and starting positions

- Added configurable logging handlers
- Command-line parsing
- Added file based logger, which is enabled with command-line arguments
- Implemented starting position finder (not optimal yet)

22.4.2010



Crosshair and round timer

- Implemented same crosshair mechanism than in UFOHippa 2
- Implemented round timer

29.4.2010



Artificial Intelligence

- Started creating AI framework
- Implemented RandomAI with random movement
- Implemented SeekerAI who directly either targets the nearest opponent or away from the tag

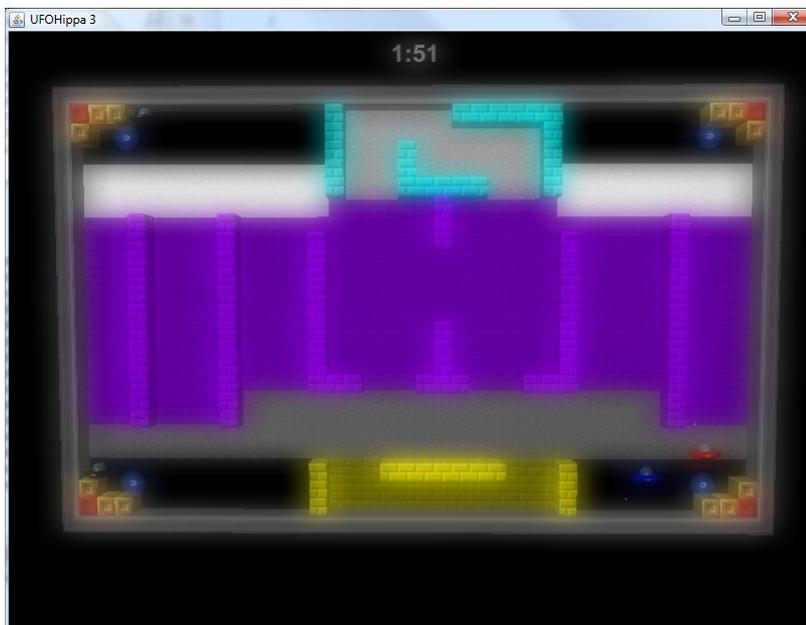
7.5.2010



Round ending

- Implementation for round ending
- Round result score calculation
- Logic for changing to next round

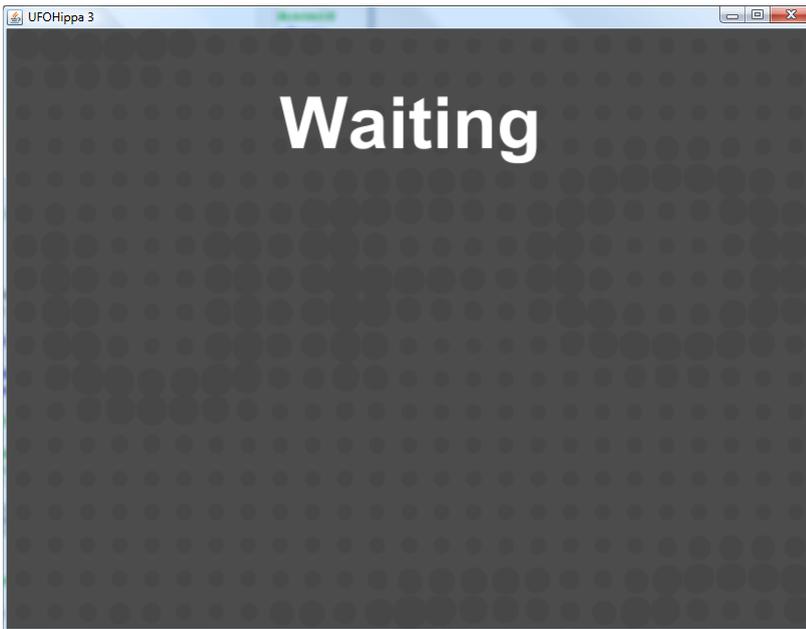
16.5.2010



Teleports

- Implemented logic and visualization for blue teleport

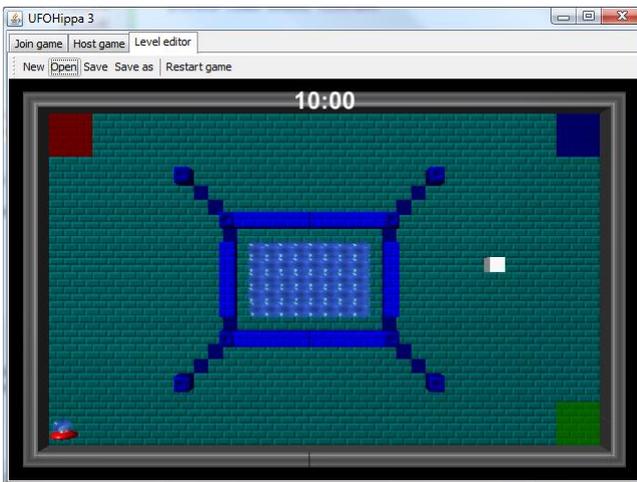
20.5.2010



Miscellaneous improvements, waiting screen

- Integrated new musics by JDruid
- Fixed some bugs related to debug client preventing main menu usage
- Added new sound effects
- Added waiting screen with rotozoomer effect for waiting other players

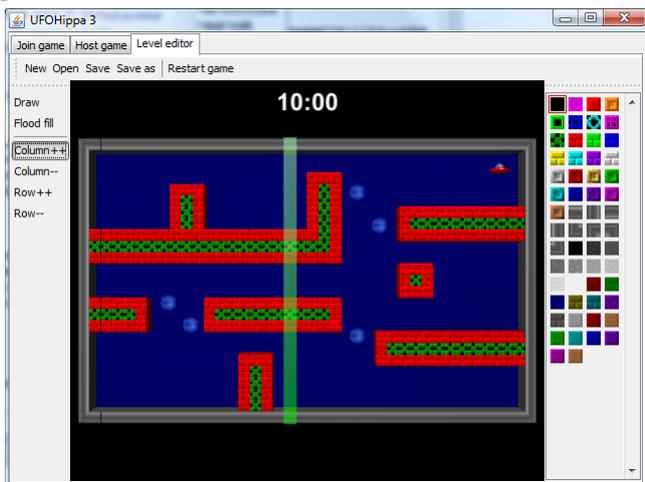
28.5.2010



Level editor

- Created a new tab for level editor
- Level editor can load levels and render them in full size
- Mouse cursor is rendered in 3D space as a white tile

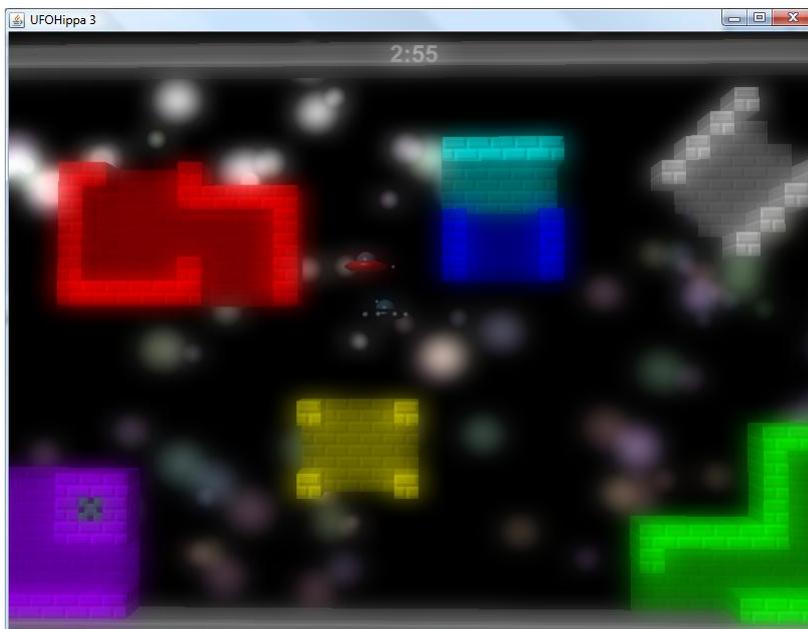
3.6.2010



### Level editor

- Tile editing
- Level resizing by inserting / deleting rows and columns
- Level saving

11.6.2010



### Background animation

- Added possibility to create new levels in level editor
- Created star system like background animation

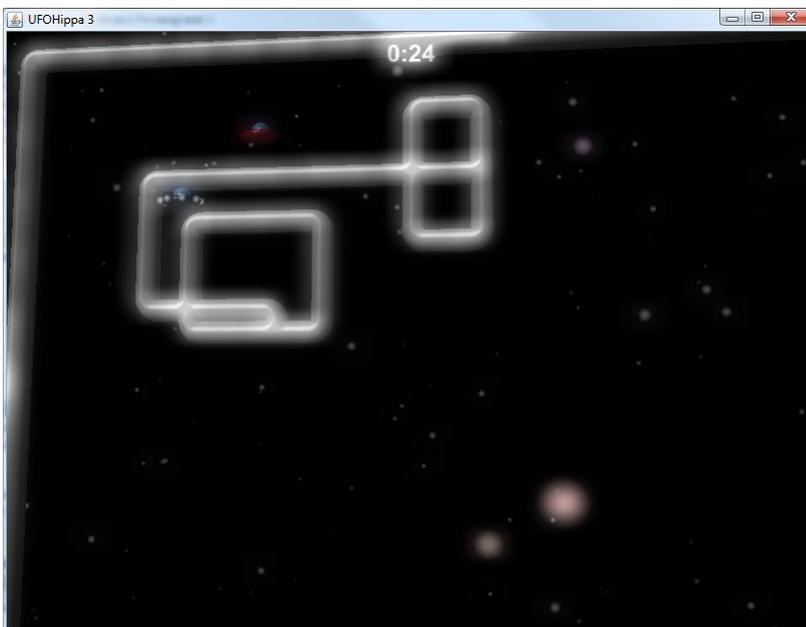
18.6.2010



Improved HDR rendering

- Added teleports to level editor
- Improved HDR rendering by blurring only bright values
- Created starfield background animation

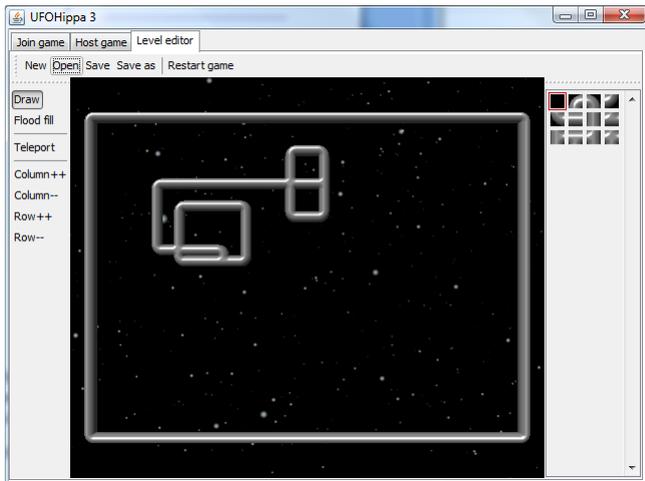
24.6.2010



Pipes

- Support for 3D models as tiles
- Created code generated pipe models

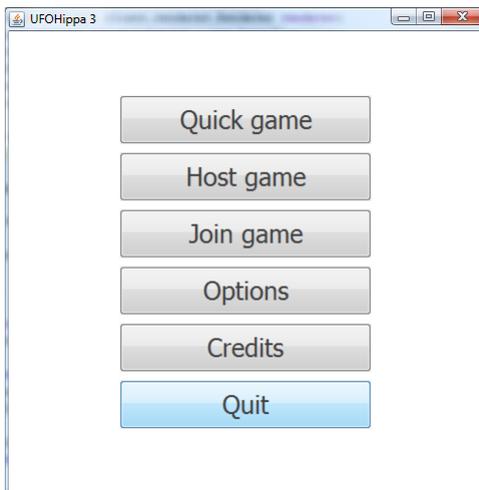
## 1.7.2010



### Support for multiple tile sets

- Level editor can now handle multiple tile sets properly
- Started new level selector tree

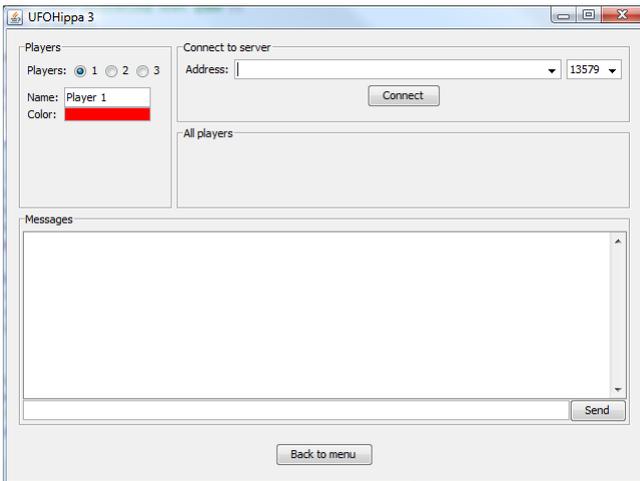
## 4.7.2010



### New menus

- Implemented new JTree based level selector
- Started implementing new menus
- Created Credits screen

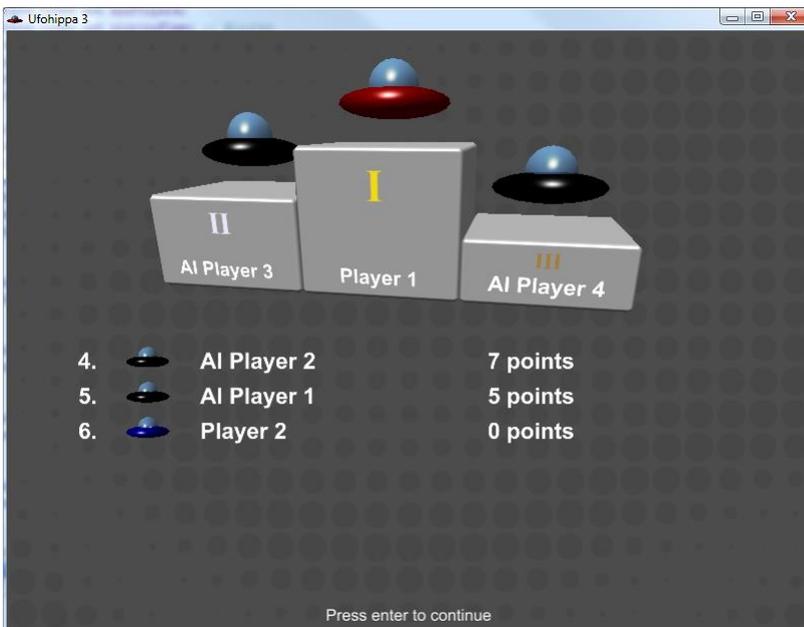
9.7.2010



### Joining

- Implemented menu for joining to a game
- Bug fixing

22.7.2010



### Polishing for release

- Mac OS X support
- Linux support
- New tiles
- Results screen

25.7.2010



First release for Assembly

- New levels
- Bug fixing
- Title screen
- Help and Readme.txt

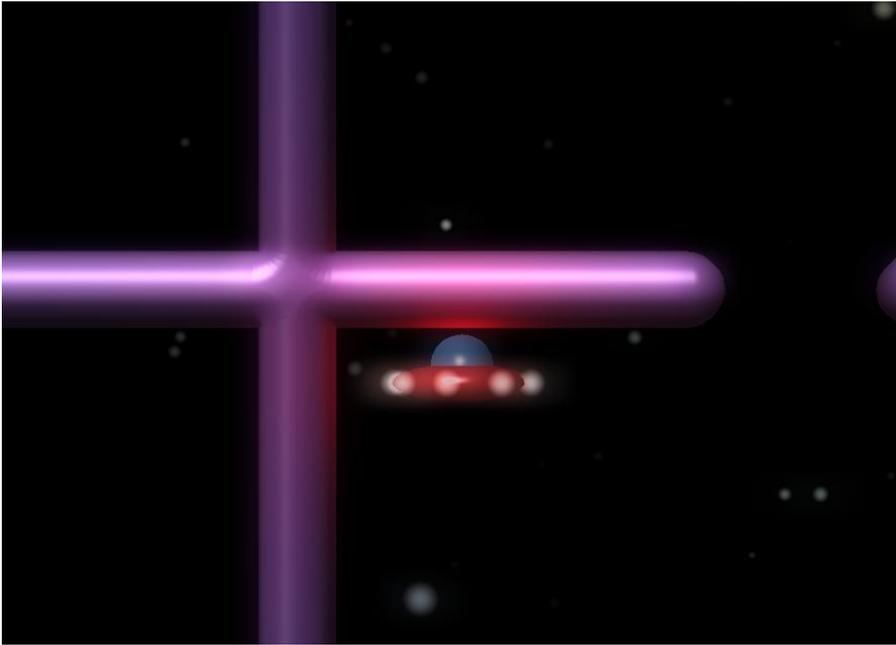
1.8.2010



Second release for Assembly

- Possibility to choose tile color
- New AI
- Bug fixing

4.8.2010



Final release

- Added point light source to tag ufo
- Bug fixing
- Wrote post mortem

## Post Mortem

The project started right after Assembly 2009 when I decided to create a sequel for the outdated game UFOHippa 2. First few weeks went for research and design and after that I started programming.

Development time took a bit over 200 working hours not including music. In the end time run out, thus I couldn't introduce even nearly all the features I wanted. The end result was still satisfying however.

The code base is 252 Java files consisting of 19000 lines of code in total (about 90 lines of code per development hour).

### What went right?

- Java was an excellent choice for OpenGL rendering and game programming  
→ Development is fast and debugging is easy
- Creating level converter for old levels early made testing easier
- Writing Development diary and keeping hour tracking was valuable with very little effort
- Working several hours per week regularly was interesting

### What went wrong?

- Network state machine was too dominative and too complex  
→ Writing new message types was easy, everything else was hard
- Creating own thread for each task made development easier, but introduces nasty bugs
- Design document written before implementation was useless
- Mixing Swing and OpenGL made full screen support impossible without sacrificing performance
- Sound effects should have been remade instead of using directly the old bad ones

### What should be done differently?

- Have separate programmers for graphics and game play
- Have an artist and support for 3D models
- Add support for dedicated server and world-wide lobby early on
- Create support for shared state objects, which are automatically synchronized among all clients and server