Qbes Voxel Engine

A short presentation with info for potential developers

Architecture

- The solution consists of 3 main projects
  - Common.Logic
  - Server.Logic
  - Client.Logic
- The other libraries and executables are only helpers

IDE and .NET Framework

- One of these IDEs should work just fine:
  - Visual Studio 2010 (not EXPRESS edition)
  - SharpDevelop 4.3 (free)
  - MonoDevelop 4
- For use with Microsoft.NET make sure you have .NET 4.0
- For use with Mono make sure you have Mono.NET 2.11 or newer (should work)

Build

- In source directory open Qbes.sln
- All source, dependencies and most resources are listed in the solution's projects
- Select either DEBUG or RELEASEDIAG build configuration
- Build the solution
  - Note that when compiling with Mono.NET you may get some errors that can be solved by disabling **treat warnings as errors** build option

Run

- When compiled start the Qbes.Server project
  - The server will load the game from source\Qbes.Server\Worlds\Hybrid\
  - Note that the world name to load is hard coded (feel free to make it configurable as this was low priority on my TODO list :-))
- Once server loads start the Qbes.Client project
  - After a while the client should receive all terrain data from the server
- Note that DEBUG is REALLY SLOW!

First Look

About the Test World

- This world was used for testing throughout the development
- It is a hybrid of initial plain terrain modified by me and testers
  - So ignore all the silly stuff :-)
- When I added a simple procedural terrain generator I merged portion of the test world with newly generated terrain
  - That's why it looks so weird on the edges

Controls

- Use mouse to look around
- Typical WSAD movement
  - Note that there's a movement bug when strafing and moving either forward or backward
- Use SPACE to go up and LCTRL to go down
  - No gravity is in place
- F1 hides HUD, F3 toggles diag info (Windows only), F11 toggles fullscreen
- G toggles mousegrab, F toggles fog

Controls

- Use mouse buttons to interact with the world (with like 3 meter range limitation)
  - Left mouse button places block
  - Right mouse button removes block
- Use mousewheel to change materials
  - R and T keys work too
- Press ENTER for chatbox
  - Only limited set of characters is allowed
- ESCAPE to exit or cancel chatbox

Server Commands

- broadcast: sends a chat message to all clients
- exit: cooks a dinner (seriously ;-))
  - Also saves the world before it start cooking
- kick: kicks a player by name
- players: lists connected clients
- save: saves changed areas and entities
- save-all: saves all data

Configuration

- For server configuration see XML file located in source\Qbes.Server\Config\
  - In code look for Qbes.Server.Logic.Configuration namespace
- For client configuration see XML file located in source\Qbes.Client\Config\
  - In code look for Qbes.Client.Logic.Configuration namespace
- The structure should be fairly self-explanatory

How it Works

- The terrain data is divided into the following hierarchy (starting with the highest level)
  - Area (64x64 blocks)
  - Segment (8x8 blocks)
  - Box (ranges from 1 to 512 blocks)
  - Cube (a single block)

Why a Box?

- Boxes are merged blocks of the same material so that less disk space, memory and processing power is used
- When placing and removing cubes boxes are automatically rearranged
- A single segment can have 0 (most optimal) to 512 boxes (least optimal)

Rendering

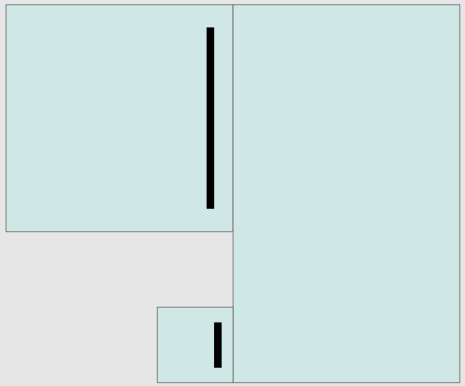
- On client there are a few basic optimizations in place
  - View distance
  - Viewport elimination
  - Hidden faces elimination
- However these are only in a very basic form and sometimes faces are eliminated when they shouldn't be :-)

More on Hidden Faces

- A box on the client contains info about its faces visibility
- These are assembled when terrain is loaded or changed and apply only to immediate adjacency
- See image on next slide

More on Hidden Faces

Thick lines indicate hidden faces



Terrain Data Files

- Each area is stored in its own file
  - I planned to change this to a sort of DB like storage with few files for terrain and index helper files
- Area definition has a header of 9 bytes
  - 0-3: length of the area data
  - 4-5: X coordinate divided by 64
  - 6: Y coordinate divided by 64
  - 7-8: Z coordinate divided by 64

Terrain Data Files

- After the area header segments are stored
- Each segments has the following 6 byte header
  - 0-1: segment data length
  - 2-5: segment version
- Note that coordinates are not stored
  - The segments need to follow in a precise order which is controlled by area serialization and deserialization

Terrain Data Files

- Segment's boxes are stored after a segment header (if there are any)
- Each box needs 6 bytes
  - 0: X1 and X2 offsets merged into a single byte
  - 1: Y1 and Y2 offsets merged into a single byte
  - 2: Z1 and Z2 offsets merged into a single byte
  - 3: Flags
  - 4-5: Material ID

Client/Server Communication

- Both UDP and TCP is utilizied
  - For UDP Lidgren library is being used
  - For TCP NetworkComms library is being used
- By default all messages are sent over UDP unless the message size (without headers) breached a set max message size
  - When breached the message is compressed, split into chunks defined by max message size and sent over TCP

Client/Server Communication

- Both client and server implement the following interfaces:
  - Qbes.Common.Logic.Networking.IClientToServer
  - Qbes.Common.Logic.Networking.IServerToClient
- UDP messages are split into various channels and have different settings regarding reliability
- There's an issue when a player taking over an old and not yet discarded connection has problems

Client/Server Communication

- Currently there is no authentication on the server
- The server also has a self-hosted HTTP server using System.Net.HttpListener
  - Very early implementation but it could be used both as a management console or serving terrain and player info, social functions...

Did I Miss Something?

- Let me know if you have some questions
- I don't really plan to interfere with this project because of time issues so as long as the following is preserved I'm OK with changes:
  - Keep it all opensource under LGPL 3
  - Check here and there if it runs under .Net 4.0 equivalent Mono as I'd like it to run on Linux as well

HAPPY CODING

AND

HAVE FUN!