

Multi-player Level Design

Who am I ?

- Thomas Buijtenweg
- Year 3 D&P
- Played **A LOT** of shooters (10000+ hours)
- Played at a semi-professional level for 6 years (Team-fortress-classic, Quake 3)
- Designer of Dragon Monastery
<http://youtu.be/ElaAVQF1z2k>

Multi-player Level Design

Topics I will cover:

- Basic Level Design examples
- FPS Player Motivation
- Designing Effective Spaces
- Flow
- Visibility
- UDK

Basic Level Design

Meaningful Choice!

A game without choice is boring (depending on the target audience)

Basic Level Design

The Basics - Chess

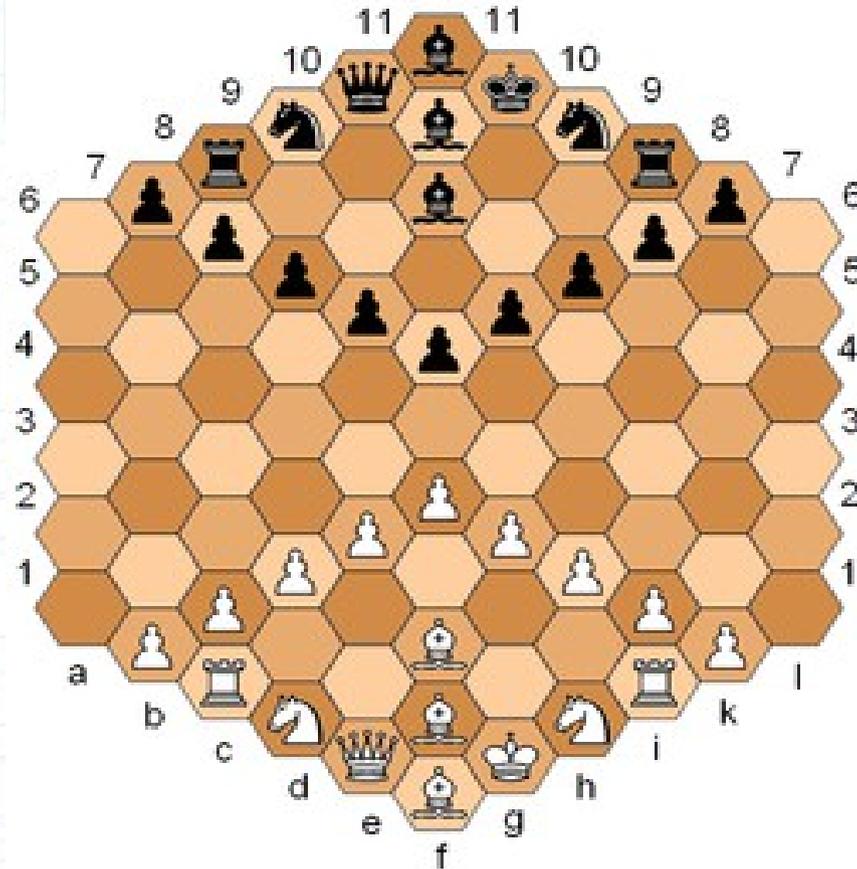
- Mirror map
- Balanced ?
- Choice ?



Basic Level Design

The Basics - Chess

- Mirror map
- Balanced ?
- Choice ?
- Is this still chess ?



Basic Level Design

DE_Dust2

- Mirror map ?
- Balanced ?
- Choice ?



Basic Level Design

Multi-player level design is about creating choices, not good versus bad (although possible), but meaningful game-changing choice.

The harder these choices are to make, the more interesting the level becomes.

Most of these choices involve movement through your level

FPS player motivation

“Surviving long enough to shoot, kill (annihilate) and preferably humiliate your opponents.”

http://youtu.be/1U2dMum_U_s

FPS player motivation

“Surviving long enough to shoot, kill (annihilate) and preferably humiliate your opponents.”

This requires:

- an opponent
- a weapon (preferably a big one)
- any kind of buff I can find to increase my chances of winning.

FPS player motivation

“Surviving long enough to shoot, kill (annihilate) and preferably humiliate your opponents.”

This requires:

Fragging them a lot more than they frag me, preferably without taking damage and a big flashing scoreboard that tells me how great I'm doing.

<http://youtu.be/LuOOvQvIAGY?t=2m31s>

FPS player motivation

Players will do anything to get that winning edge.

Players will break your game, either the map or the game mechanics, or both in order to get the advantage they need to win.

<http://youtu.be/7rFDo7nUraY>

FPS player motivation

Know your game:

If you know about the exploits in your game you can design around them, or simply accept them and include them into the design.

<http://youtu.be/IOCWr8gU64k?t=49s>

Designing Effective Spaces

As a personal guide I use the following:

- Each area must have a **function**
- Each area must lead to **at least 2 other functions**
- After completing its function, an area should (almost) **never be useful to stay in**
- Try to stick with **triangle item placement**

Designing Effective Spaces

Each area must have a function

Blue-Red-Yellow-Green pick-up

(Keg o' Health, Shield-belt, Body-armor, Thigh-pads)



Designing Effective Spaces

Each area must have a function

Weapon pick-up

(Pulse Rifle, Shock Rifle, Rocket Launcher)



Designing Effective Spaces

Each area must have a function

Booster pick-up

(U-Damage, Berserker, Jump Boots)



Designing Effective Spaces

Each area must have a function

- Resupply
Allow the player to calm down and pick up some health/ammo to get back in the fight
- Overwatch
Let the player view the battlefield, scout where his opponents are and give him a chance to plan his next move

Designing Effective Spaces

Each area must lead to at least 2 other functions

A One-way area is boring for the player (no choice!), but also highly predictable for his opponent.

An area that leads to a similar function is very unrewarding for the player.

Designing Effective Spaces

After completing its function an area should (almost) never be useful to stay in.

If an area has a large benefit and has all resources nearby, players will remain within the area for extended periods.

If multiple players do this in different areas, nobody will encounter an enemy.

Designing Effective Spaces

Try to stick with triangle item placement

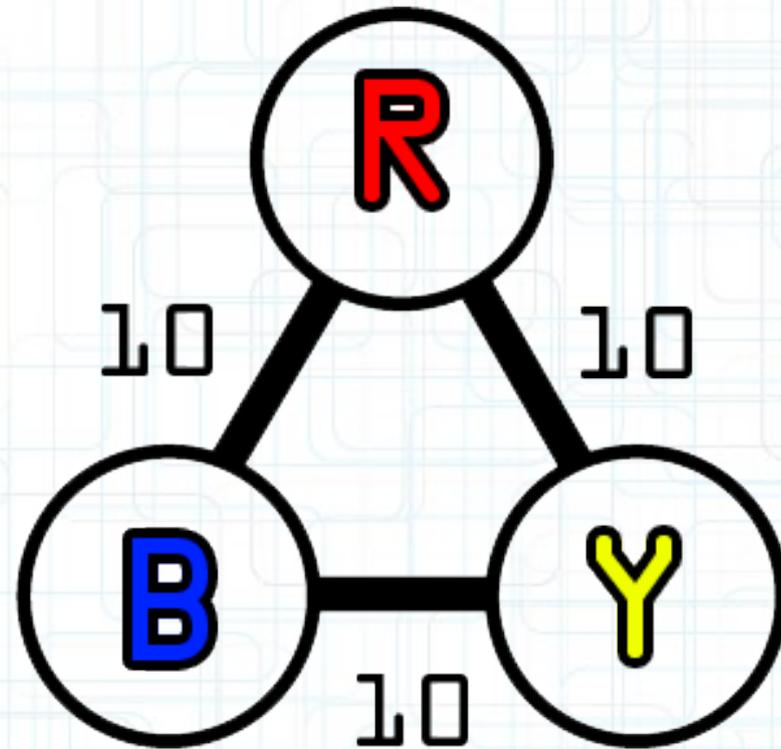
This requires some explanation

In essence it is a theoretical way to balance your level. This obviously does not remove the requirement of play-testing.

Designing Effective Spaces

Example 1:

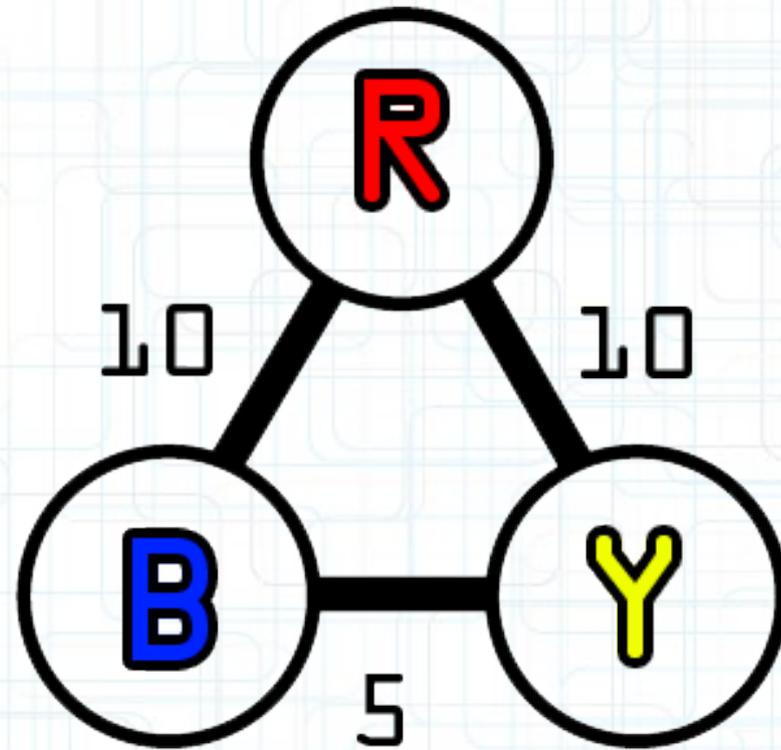
- All distances are equal
- In a 1-on-1 each player will have a major buff (B or R)



Designing Effective Spaces

Example 2:

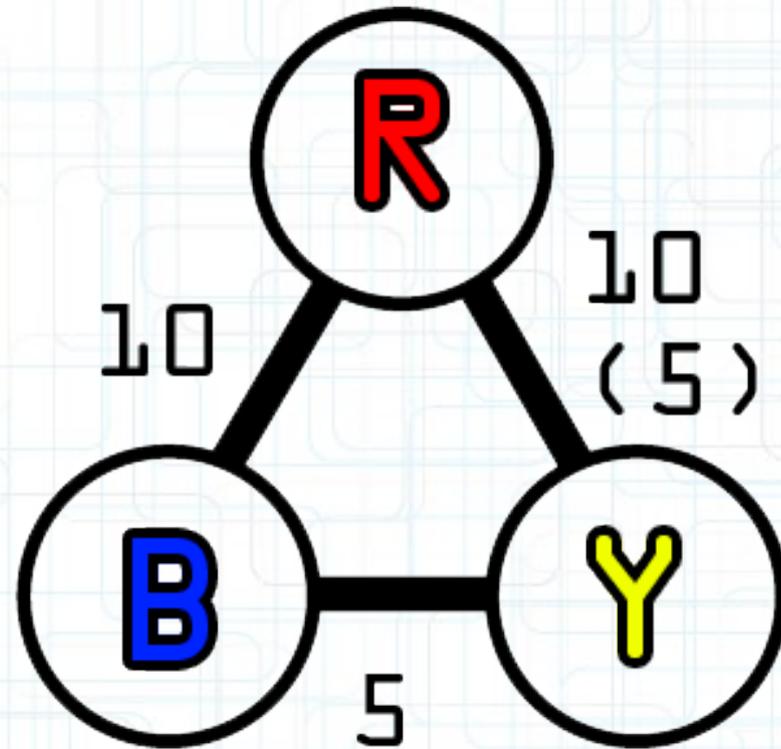
- Route B,Y is shorter
- Player that goes for B will also have Y



Designing Effective Spaces

Example 3:

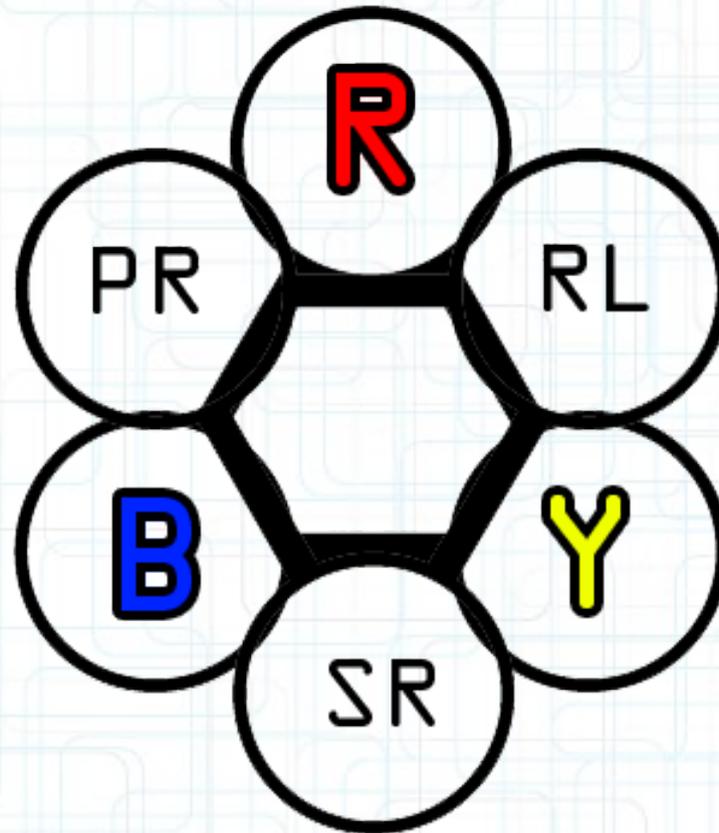
- Route R,Y has a Ninja-path
- Players will fight at Y if player at R succeeds at moving over the Ninja-path



Designing Effective Spaces

Example 4:

- Add in another triangle with weapons
- Where would you go ?



Designing Effective Spaces

Other things to consider:

Add a Udamage to the center room ?

One-way or Two-way ninja-path ?

Punish the player if he fails at making a ninja-path ?

How exposed is the player when collecting a pick-up ?

Designing Effective Spaces

Know the answers to these questions and you are on your way towards a balanced map.

(and you haven't even started building yet)

Designing Effective Spaces

What about new players ?
They don't know the locations yet !

Buff routes !

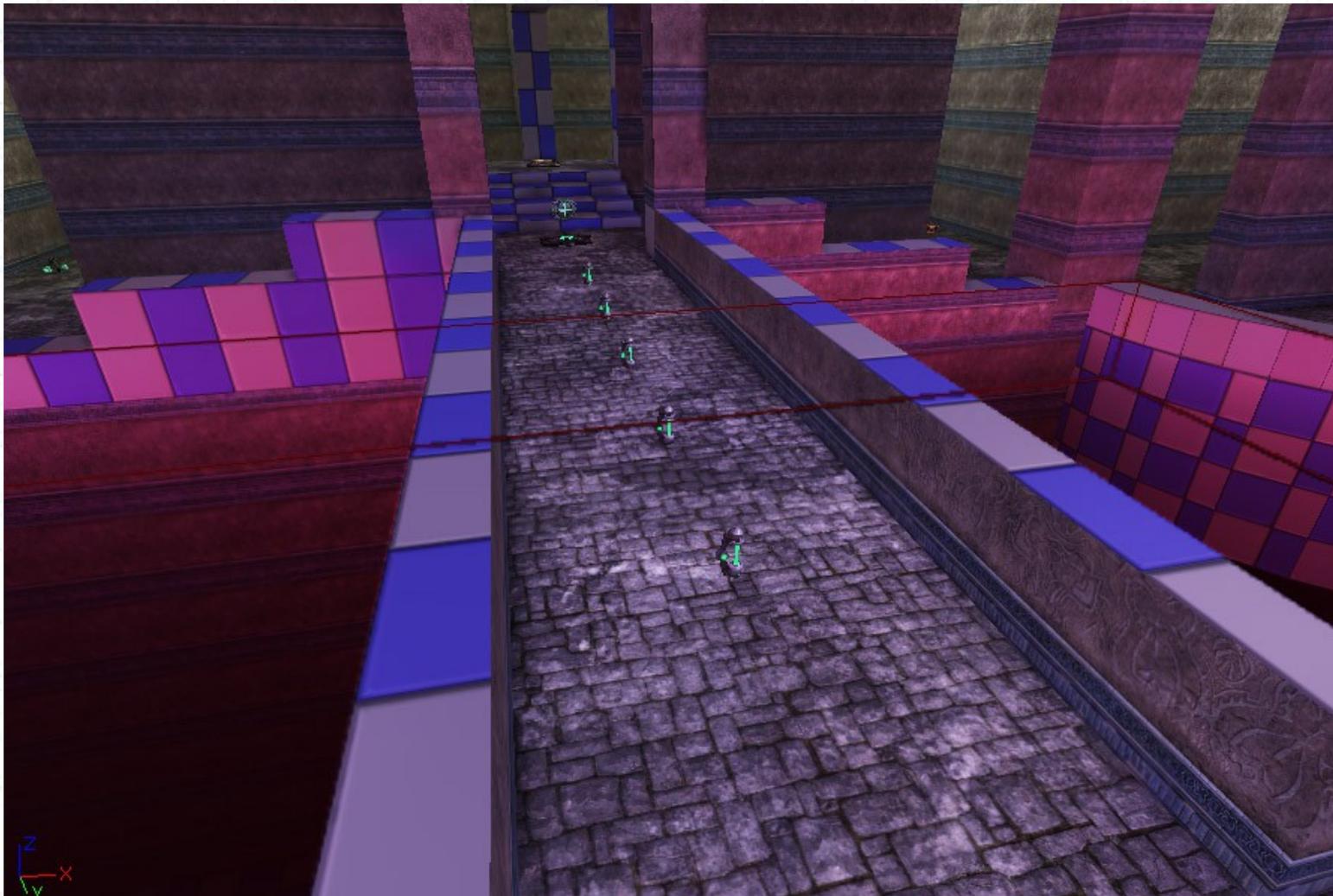
Designing Effective Spaces

Buff routes are a great tool to help you guide players towards a goal.

They can tempt players to move (more buffs = more winning), hint towards ninja-paths, guide players to important pick-ups, etc.

Designing Effective Spaces

Buff Route example:



Designing Effective Spaces

A buff route generally consists of multiple minor buffs in a string.

In UDK you could use for example health-vials or ammo-packs



Designing Effective Spaces

Another important aspect of any level is the use of **Height-levels**

http://youtu.be/x2m_hwDIntw

Designing Effective Spaces

DM maps are like Ogres ...
And onions ?

Anyway, they have layers!

Designing Effective Spaces

Teleporters and Jump pads can help you to create interesting movement among layers, they also prevent the need for massive staircases.

Designing Effective Spaces

Layers add movement options to the player.

They add a level of risk (falling down).

They add a lot of ways to use weapons.

In short, ogres onions and layers are great for creating choice.

Flow

“The ability to switch paths and navigate through a level effortlessly and efficiently.”

<http://youtu.be/VAVsJI2PCiM?t=8m30s>

Flow

Flow in a level is how smooth a:

- Beginner can follow the obvious paths
- Advanced players can use skill to gain benefit
- A pro can abuse mechanics to dominate

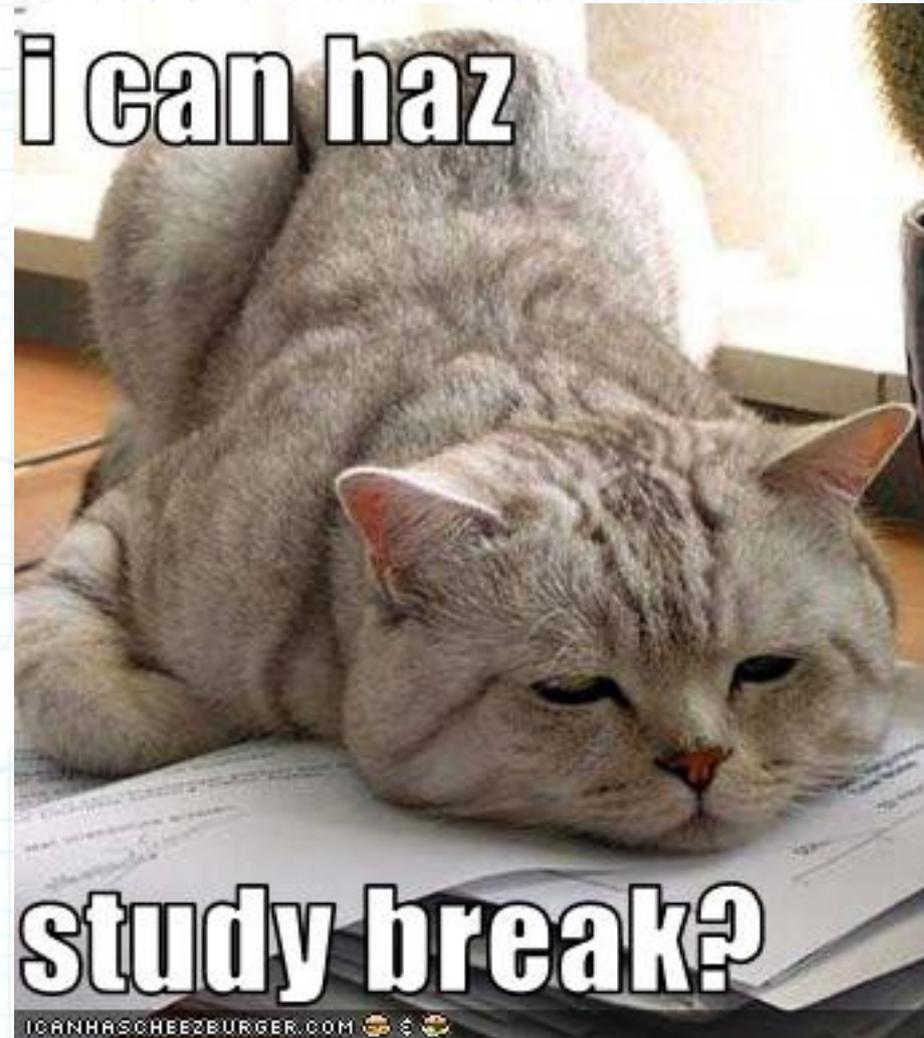
Flow

Flow is a critical aspect of your level. It can break a level. If the flow of a level is broken the level will become irritating to play.

You can test the flow of your level, testing is the only way to be sure.

No matter how good the theory behind it all, it comes down to how the level feels and flows.

Can I haz Break ?



Visibility

What can the player see from what location ?

What can he not see ?

What can his opponents see / find out indirectly
(sound, missing pick-ups, light triggers)

Visibility

Players deduce what to do from what they know or can predict.

Seeing and hearing...

And experience.

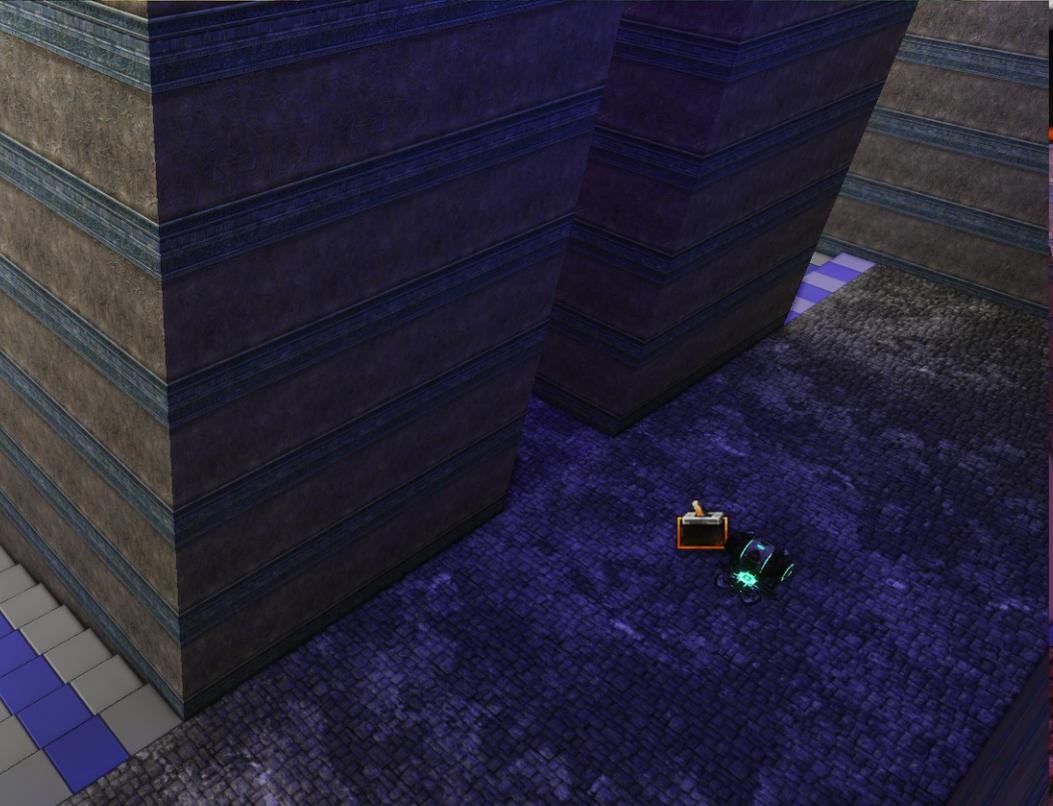
Controlling visibility is controlling the players movement.

Visibility

Player orientation depends largely on how well they know the level and how it looks.

If a level has a lot of similar architecture you can aid the player with landmarks and colour coding.

Visibility



UDK

Some love it, some hate it.

Right now you don't have a choice.

Its a powerful tool with some drawbacks.

Your job is to make something awesome
regardless.

UDK

Although the editor is easy to use, level design starts on paper (generally).

First make some basic sketches, catch problems before you spend hours building them.

Don't wait too long though, it takes time to learn UDK and I hope that at this point you have already started building

UDK

Things to do when working with UDK:

Stick to the grid!

Turn on Autosave!

Save previous versions

Stick with cubes and basic shapes

Start with a large grid setting and only go smaller when you really need to

Try to not use subtracts too much, it gets messy really fast

UDK

UDK tutorials for LD1

Basic BSP tutorial:

<http://www.worldofleveldesign.com/categories/udk/udk-simple-room-creation-part1-bsp-block-in.php>

BSP part 2:

<http://www.worldofleveldesign.com/categories/udk/udk-simple-room-creation-part2-bsp-brushes-workflow.php>

Basic Lights:

<http://www.worldofleveldesign.com/categories/udk/udk-simple-room-creation-part3-quick-lighting-in-game-testing.php>

Gameplay (jumppads, weapons etc):

<http://www.worldofleveldesign.com/categories/udk/udk-how-to-add-player-start-weapons-items-pathnodes-vehicles-jumppads.php>

Map settings:

<http://www.worldofleveldesign.com/categories/udk/udk-how-to-setup-deathmatch-teamdeathmatch-gametype.php>

(optional) Basic Kismet scripting:

<http://www.worldofleveldesign.com/categories/wold-members-tutorials/petebottomley/udk-kismet-introduction.php>

<http://www.worldofleveldesign.com/categories/wold-members-tutorials/petebottomley/udk-moving-doors.php>