

UX/UI

Intro

```
* The coordinates (0, 0, 0) represents the octocube  
*/  
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

				5	6			
				7	8			2
				3	4			

* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate
* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```

This week?

- One of four UX/UI weeks
- Alongside the game project
- Understand what UI and UX are
- What to think about when designing UI and UX
- Common pitfalls, best practices

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

* <code>



* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate

* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```


Who am I?

- Anna Jenelius
- Bachelor's Degree in Game Development (Stockholm University)
- First game development job in 2011 (Animation)
- Started at Paradox in 2012
 - QA Tester
 - Assistant QA Manager
 - Senior QA Manager
- Quit my job, started Talecore Studios
- One-person indie developer
- Head Teacher for the Game Design program here at FutureGames
- Also: Writer, making a card game, organizing gamedev meetups (Link in Park/Link in Bar), etc



What have I done?

Independent Game Developer

Midvinter
Shantee's Choices (in development)
Capitals (in development)

Writer

Rain of Reflections (in development)

Analog Games

IndieCards (in development)

Motion Capture Assistance

Battlefield 3

Animation

The Horror of MS Aurora

Game Jams

Castle Game Jam
Global Game Jam
International Love Ultimatum Jam

QA

A Game of Dwarves
Ancient Space
Cities in Motion
Cities in Motion 2
Cities: Skylines
Crusader Kings II
Dungeonland
Europa Universalis IV
Hearts of Iron III
Hearts of Iron IV
Hollowpoint
Impire
Knights of Pen and Paper +1 Edition
Knights of Pen and Paper II
Leviathan: Warships
Magicka

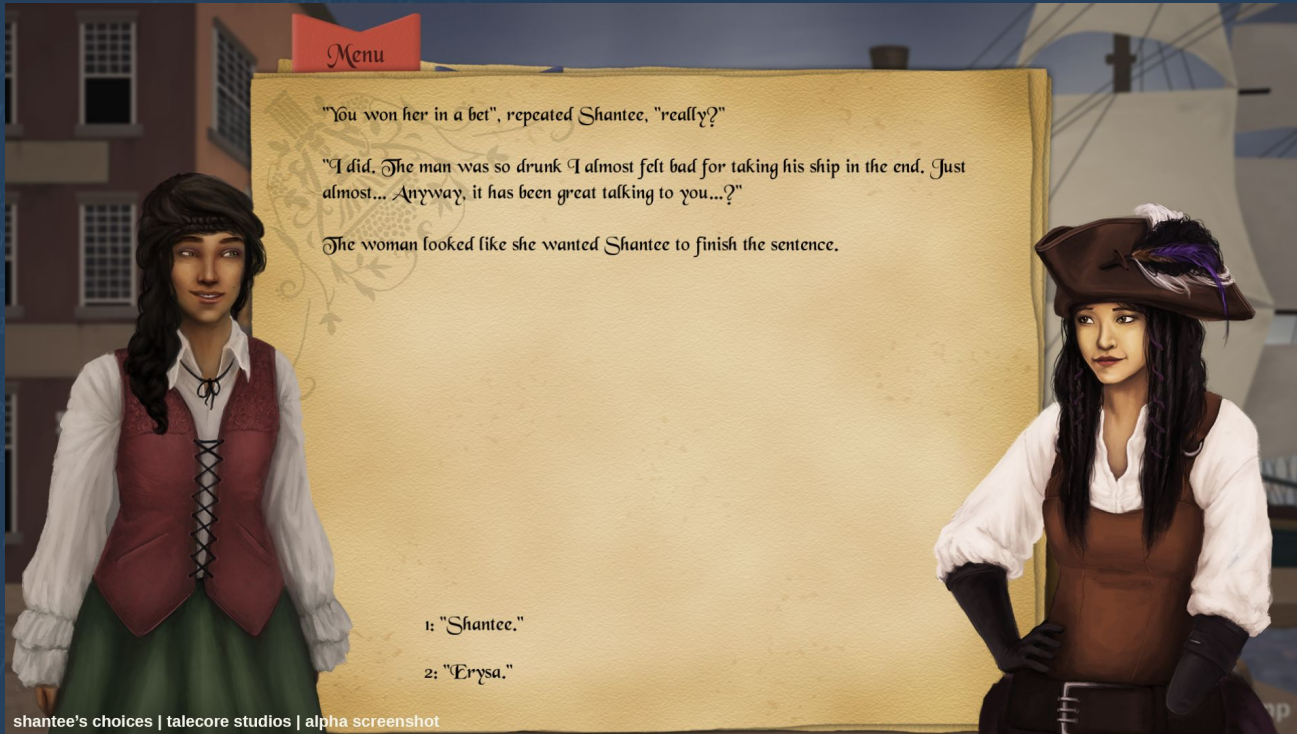
Magicka 2

Magicka: Wizards of the Square Tablet
Magicka: Wizard Wars
Pillars of Eternity
Runemaster (Cancelled)
The Showdown Effect
Starvoid
Stellaris
Sword of the Stars II
Teleglitch: Die More Edition
Warlock: Master of the Arcane
Warlock 2: The Exiled
War of the Roses
War of the Vikings

Midvinter



Shantee's Choices



shantee's choices | talecore studios | alpha screenshot

Capitals



Q: What does UX and UI stand for?

```
 * Common fields
 * Format: {field: [0]}
 * Placeholder
 * Placeholder
 * Dotted heuristic to distinguish known hosts from known hosts2.
 * Is second field entirely decimal digits?
 * Example: "192.168.1.1"
 * Treat all fields as host keys
 * Format: {hostname: [0]}
 * (PUTTY doesn't store the number of bits)
 * hostname = mac (long, hexadecimal)
 * hostname = "xxx"
```

```
 * The coordinates (0, 0, 0) represents the octocube
 */
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:



```
 * @param int $x the x coordinate
 * @param int $y the y coordinate
 * @param int $z the z coordinate
 * @return int the number of the sector (0 if x =
```

```
static function get_sector ($x, $y, $z) {
```

Answer:

User Experience and User Interface

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

	5	6	
	-	-	-
7	8		2
3	4		

* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate
* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```


Why do we use Uls?

```
    * Common fields
    * Format: hostid[0]
    * regionnumber = []
    * keytype = ""
    * placeholder

    * Drotty heuristic to distinguish known_hosts from known_hosts2:
    * Is second field entirely decimal digits?
    * If so, it's "old" style fields[1]

    * Treat all fields as host keys
    * Format: hostid[0]
    * regionnumber = []
    * keytype = ""
    * placeholder

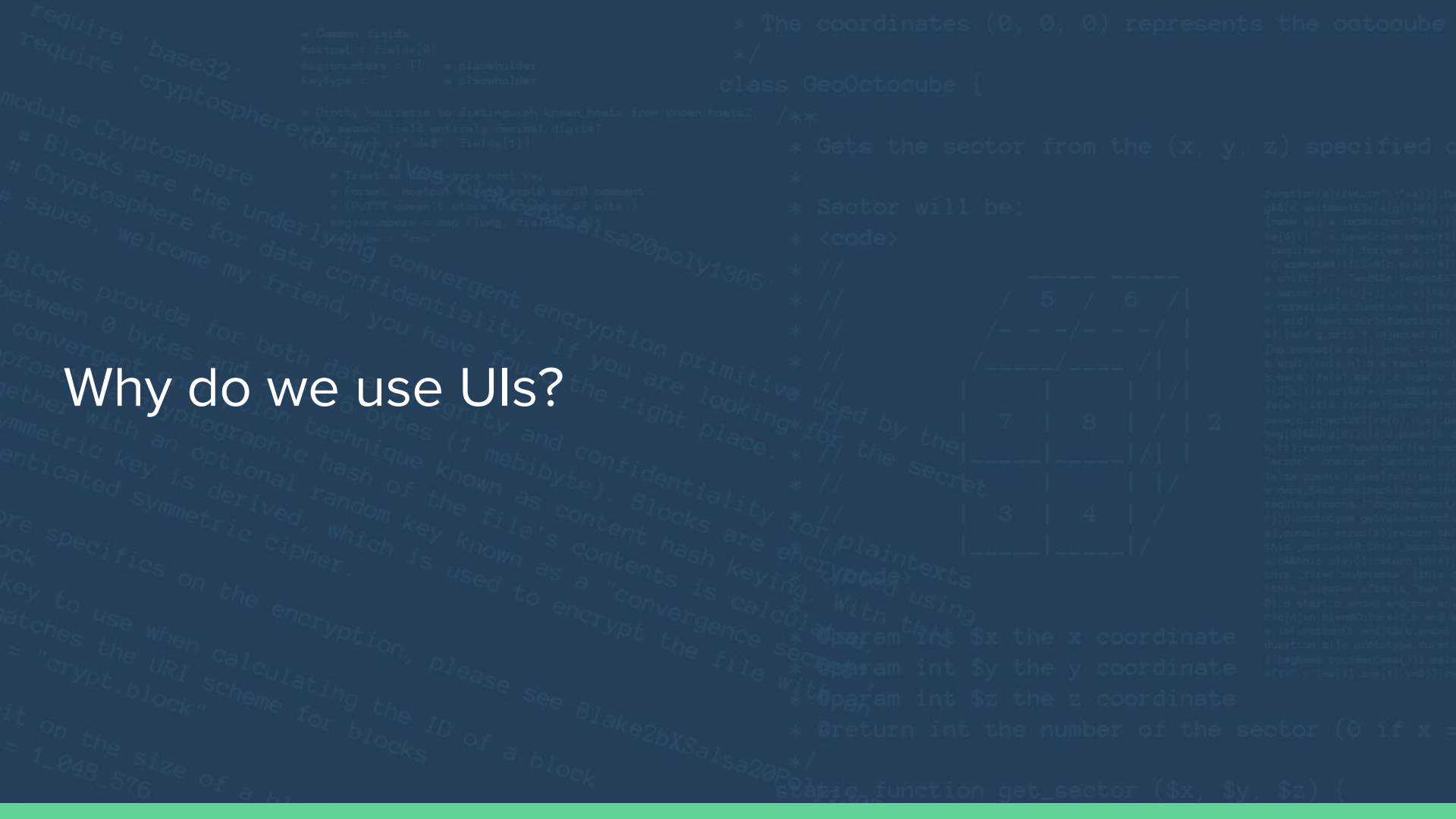
    * PUTTY doesn't store the order of bits
    * regionnumber = map (long, fields[1])
    * keytype = "ssh"
```

```
    * The coordinates (0, 0, 0) represents the octocube
    */
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:


```
    * @return int the number of the sector (0 if x =
    * param int $x the x coordinate
    * param int $y the y coordinate
    * param int $z the z coordinate
    * @return int the number of the sector (0 if x =
    static function get_sector ($x, $y, $z) {
```



Purpose

- Display information
- Interact with game

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

		5	6	
		7	8	2
		3	4	

```
* @param int $x the x coordinate
* @param int $y the y coordinate
* @param int $z the z coordinate
* @return int the number of the sector (0 if x =
static function get_sector ($x, $y, $z) {
```

What kinds of UIs?

```
* Common fields
format = fields[0]
regionnumbers = [] * placeholder
keytype = "" * placeholder

* Drotty heuristic to distinguish known_hosts from known_hosts2:
is second field entirely decimal digits?
return (r"%d*" % fields[1])
```

```
* Treat all files as host keys
* Format: hostkey [key type] [comment]
* (PUTTY doesn't store the type of bits)
regionnumbers = map (long, fields[1].split("."))
keytype = ""
```

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

* <code>



* Used by the
* secret

* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate
* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```

```
function() { return ($x >= 0 && $y >= 0 && $z >= 0) ? 1 : ($x >= 0 && $y >= 0 && $z < 0) ? 2 : ($x >= 0 && $y < 0 && $z >= 0) ? 3 : ($x >= 0 && $y < 0 && $z < 0) ? 4 : ($x < 0 && $y >= 0 && $z >= 0) ? 5 : ($x < 0 && $y >= 0 && $z < 0) ? 6 : ($x < 0 && $y < 0 && $z >= 0) ? 7 : ($x < 0 && $y < 0 && $z < 0) ? 8 : 0; }
```

Menus



HUD



Inventory



Codex/logs



Other?

```
    * Common fields
    * Postal = fields[0]
    * regionnumbers = []
    * laytype = ""
    * placeholder

    * Drotty heuristic to distinguish known_hosts from known_hosts2.
    * Is second field entirely decimal digits?
    * @param r["id", fields[1]]
    * @return bool

    * Travel all fields, type host key
    * Format: hostkey: [key] expid: [expid] comment: [comment]
    * (PUTTY doesn't store the size of bits)
    * regionnumbers = map (long, fields)
    * laytype = "raw"
```

```
    * The coordinates (0, 0, 0) represents the octocube
    */
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:

* <code>

* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate

* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```

```
function get_sector($x, $y, $z) {
    $x = floor($x/10);
    $y = floor($y/10);
    $z = floor($z/10);
    $sector = 0;
    for ($i = 0; $i < 10; $i++) {
        for ($j = 0; $j < 10; $j++) {
            for ($k = 0; $k < 10; $k++) {
                if ($i == $x && $j == $y && $k == $z) {
                    $sector = $i * 100 + $j * 10 + $k;
                }
            }
        }
    }
    return $sector;
}
```

Terminology

Diegetic: Interface that is included in the game world -- i.e., it can be seen and heard by the game characters.

Non-diegetic: Interface that is rendered outside the game world, only visible and audible to the players in the real world.

Spatial: UI elements presented in the game's 3D space with or without being an entity of the actual game world (diegetic or non-diegetic).

Meta: Representations can exist in the game world, but aren't necessarily visualized spatially for the player; these are **meta representations**.

Diegetic



Diegetic



Non-Diegetic



Non-Diegetic



Spatial



Spatial



```
require 'base32'  
require 'cryptosphere'  
module Cryptosphere  
  # Blocks are  
  # Cryptosphere  
  # sauce, welco  
  # Blocks provid  
  # between 0 byte  
  # convergent en  
  # approach, a cry  
  # together with an  
  # symmetric key f  
  # authenticated symm  
  # re specifics  
  # lock  
  # key to use whe  
  # matches the UR  
  # = "crypt:blo  
  # it on the size of  
  # = 1_048_576  
  # Common fields  
  Fieldset = Fieldset[]  
  maxnumbytes = 11  
  keytype = ""  
  # placeholders  
  # placeholders  
  # Ducky heuristic to distinguish known hosts from known hosts2. /**  
  # second field entirely decimal digit  
  # "0x00" fields[]  
  # The coordinates (0, 0, 0) represents the octocube  
  */  
  class GeoOctocube {  
    # Gets the sector from the (x, y, z) specified  
    def self.get_sector(x, y, z)  
      # ...  
    end  
  end  
end
```

Meta



Meta



Is the representation visualized in the 3D game space?

no yes

no	non-diegetic representations	spatial representations
yes	meta representations	diegetic representations

Is the representation existing in the fictional game world?

no

yes

```
function Octocube::getSector($x, $y, $z) {  
    $x = floor($x/$radius);  
    $y = floor($y/$radius);  
    $z = floor($z/$radius);  
    $x = ($x < 0) ? -$x : $x;  
    $y = ($y < 0) ? -$y : $y;  
    $z = ($z < 0) ? -$z : $z;  
    $x = ($x > 7) ? 7 : $x;  
    $y = ($y > 7) ? 7 : $y;  
    $z = ($z > 7) ? 7 : $z;  
    $sector = ($x * 8 + $y) * 8 + $z;  
    return $sector;  
}
```

Games with Uls you like?

```
/* Common fields
 * Postal = fields[0]
 * numnumbers = {}
 * keytype = ""
 *
 * Dirty heuristic to distinguish known hosts from known hosts2.
 * Is second field entirely decimal digits?
 * if (isdec("fields[1]"))
 *
 * Treat all fields as host keys
 * Format: hostkey [email] modif comment
 * (PUTTY doesn't store the size of bits)
 * numnumbers = num (long, float, int)
 *
 * Example: "192.168.1.1"
 */
```

```
/* The coordinates (0, 0, 0) represents the octocube
 */
class GeoOctocube {
```

* Gets the sector from the (x, y, z) specified

* Sector will be:



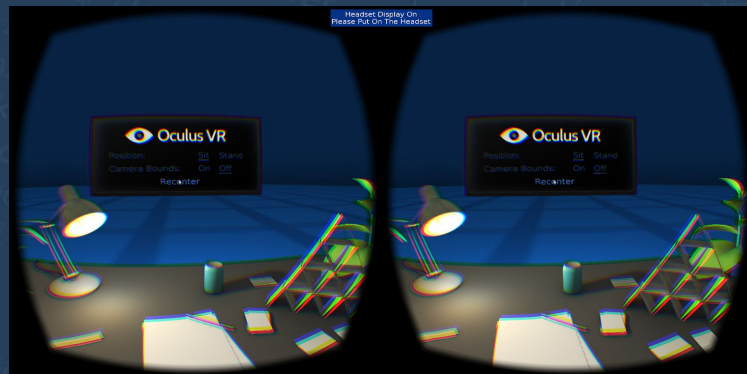
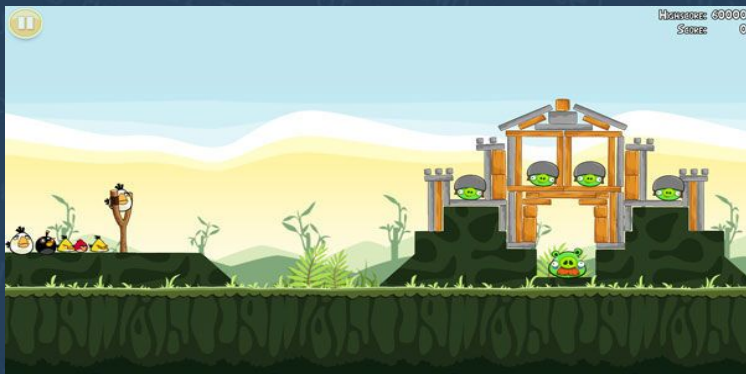
* @param int \$x the x coordinate
* @param int \$y the y coordinate
* @param int \$z the z coordinate

* @return int the number of the sector (0 if x =

```
static function get_sector ($x, $y, $z) {
```

```
function get_sector ($x, $y, $z) {
    $x = floor($x/10);
    $y = floor($y/10);
    $z = floor($z/10);
    $sector = 0;
    if ($x > 0) {
        $sector = 5;
    }
    if ($y > 0) {
        $sector = 6;
    }
    if ($z > 0) {
        $sector = 7;
    }
    if ($x < 0) {
        $sector = 8;
    }
    if ($y < 0) {
        $sector = 3;
    }
    if ($z < 0) {
        $sector = 4;
    }
    return $sector;
}
```


Platform specifics



PC vs. console

- Elements close to edge
- Input method
- Lists vs. grids

```

* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {

```

* Gets the sector from the (x, y, z) specified

* Sector will be:



```

* @param int $x the x coordinate
* @param int $y the y coordinate
* @param int $z the z coordinate
* @return int the number of the sector (0 if x =
static function get_sector ($x, $y, $z) {

```

PC vs. console



PC vs. console



PC vs. console



Mobile

- Big elements
- No hover

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

```
* Gets the sector from the (x, y, z) specified
```

```
* Sector will be:
```

```
* <code>
```

	5	6	
-	-	-	-
7	8		2
3	4		

```
* @param int $x the x coordinate
* @param int $y the y coordinate
* @param int $z the z coordinate
```

```
* @return int the number of the sector (0 if x =
```

```
static function get_sector ($x, $y, $z) {
```

Mobile



VR



```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
    * Gets the sector from the (x, y, z) specified
    *
    function getSector($x, $y, $z) {
        $x = floor($x/100);
        $y = floor($y/100);
        $z = floor($z/100);
        $sector = 0;
        if ($x < 0) {
            $sector = 1;
        }
        if ($y < 0) {
            $sector = 2;
        }
        if ($z < 0) {
            $sector = 3;
        }
        if ($x > 0) {
            $sector = 4;
        }
        if ($y > 0) {
            $sector = 5;
        }
        if ($z > 0) {
            $sector = 6;
        }
        return $sector;
    }
}

function getSector($x, $y, $z) {
    $x = floor($x/100);
    $y = floor($y/100);
    $z = floor($z/100);
    $sector = 0;
    if ($x < 0) {
        $sector = 1;
    }
    if ($y < 0) {
        $sector = 2;
    }
    if ($z < 0) {
        $sector = 3;
    }
    if ($x > 0) {
        $sector = 4;
    }
    if ($y > 0) {
        $sector = 5;
    }
    if ($z > 0) {
        $sector = 6;
    }
    return $sector;
}

static function getSector($x, $y, $z) {
    $x = floor($x/100);
    $y = floor($y/100);
    $z = floor($z/100);
    $sector = 0;
    if ($x < 0) {
        $sector = 1;
    }
    if ($y < 0) {
        $sector = 2;
    }
    if ($z < 0) {
        $sector = 3;
    }
    if ($x > 0) {
        $sector = 4;
    }
    if ($y > 0) {
        $sector = 5;
    }
    if ($z > 0) {
        $sector = 6;
    }
    return $sector;
}
}
```




```
require 'base32'  
require 'cryptosphere'  
module Cryptosphere  
  # Blocks are the units  
  # Cryptosphere for data  
  # sauce, welcome my friend  
  
  Blocks provide for both  
  between 0 bytes and 1024  
  convergent encryption  
  approach, a cryptographic  
  together with an optional  
  symmetric key is derived  
  authenticated symmetric  
  more specifics on the  
  block  
  key to use when calculating  
  matches the URI scheme  
  = "crypt: block"  
  
  fit on the size of  
  = 1_048_576
```

The coordinates (0, 0, 0) represents the octocube

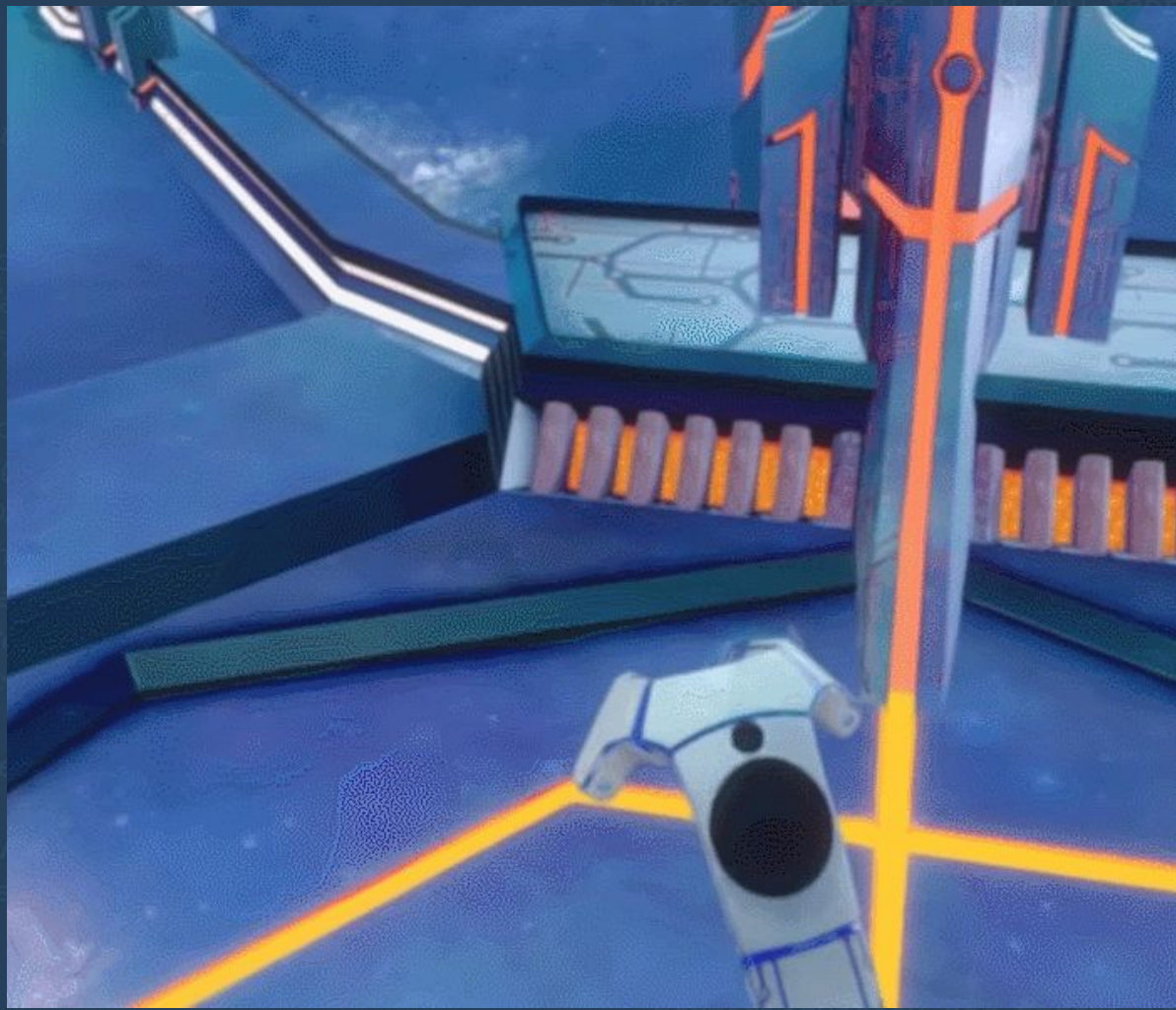
x, y, z) specified

```
function return value of  
get(x, y, z) {  
  if (x < 0 || y < 0 || z < 0)  
    return null  
  return data[x][y][z]  
}  
  
function set(x, y, z, value)  
{  
  if (x < 0 || y < 0 || z < 0)  
    return null  
  data[x][y][z] = value  
}  
  
function delete(x, y, z)  
{  
  if (x < 0 || y < 0 || z < 0)  
    return null  
  data[x][y][z] = null  
}
```

nate
nate
nate
the sector (0 if x =
x, \$y, \$z) {



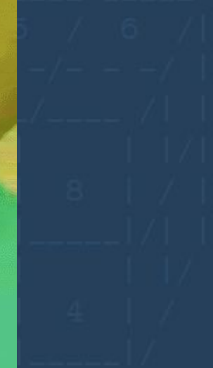
2





* The coordinates (0, 0, 0) represents the octocube
*/

from the (x, y, z) specified



```
function returnSector(x, y, z) {
  let octocube = {
    [0]: {x: 0, y: 0, z: 0},
    [1]: {x: 1, y: 0, z: 0},
    [2]: {x: 0, y: 1, z: 0},
    [3]: {x: 1, y: 1, z: 0},
    [4]: {x: 0, y: 0, z: 1},
    [5]: {x: 1, y: 0, z: 1},
    [6]: {x: 0, y: 1, z: 1},
    [7]: {x: 1, y: 1, z: 1}
  }
  let sector = 0;
  for (let i = 0; i < 8; i++) {
    let [x, y, z] = octocube[i];
    if (x >= x) {
      if (y >= y) {
        if (z >= z) {
          sector = i;
        }
      }
    }
  }
  return sector;
}
```

x coordinate
y coordinate
z coordinate
number of the sector (0 if x = 0, y = 0, z = 0)

```
* static function get_sector($x, $y, $z) {
```


Immersion

- Give the info
- Don't take the player "out" of the experience
- Don't be intrusive

```
* The coordinates (0, 0, 0) represents the octocube
*/
class GeoOctocube {
```

```
* Gets the sector from the (x, y, z) specified
*
* Sector will be:
* <code>
```



```
* @param int $x the x coordinate
* @param int $y the y coordinate
* @param int $z the z coordinate
* @return int the number of the sector (0 if x =
static function get_sector ($x, $y, $z) {
```

Make important things obvious

- Don't have to see to "get"
 - Damage - red edges
 - Velocity in car game



Visibility of System Status

“The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.”

- Nielsen, 1995



Recap

- UI = User Interface
- There are several different kinds, different purposes
 - In/outside the game world
 - Visible/Invisible to the characters
- Different platforms have different pros and cons
- Basic design practices
 - Keeping immersion
 - Don't intrude
 - Visibility of system status



```
 * The coordinates (0, 0, 0) represents the octocube
 */
class GeoOctocube {
    /**
     * Gets the sector from the (x, y, z) specified
     *
     * Sector will be:
     * <code>
```

```
static function get_sector (int $x, int $y, int $z) {
    // 0 is the origin (0, 0, 0)
    // 1 is the sector (0, 0, 0)
    // 2 is the sector (0, 0, 1)
    // 3 is the sector (0, 1, 0)
    // 4 is the sector (0, 1, 1)
    // 5 is the sector (1, 0, 0)
    // 6 is the sector (1, 0, 1)
    // 7 is the sector (1, 1, 0)
    // 8 is the sector (1, 1, 1)

    if ($x < 0 || $y < 0 || $z < 0) {
        return -1;
    }

    if ($x > 1 || $y > 1 || $z > 1) {
        return -1;
    }

    // 0 is the origin (0, 0, 0)
    // 1 is the sector (0, 0, 0)
    // 2 is the sector (0, 0, 1)
    // 3 is the sector (0, 1, 0)
    // 4 is the sector (0, 1, 1)
    // 5 is the sector (1, 0, 0)
    // 6 is the sector (1, 0, 1)
    // 7 is the sector (1, 1, 0)
    // 8 is the sector (1, 1, 1)

    return ($x * 8 + $y * 4 + $z * 2);
}

static function get_sector ($x, $y, $z) {
    // 0 is the origin (0, 0, 0)
    // 1 is the sector (0, 0, 0)
    // 2 is the sector (0, 0, 1)
    // 3 is the sector (0, 1, 0)
    // 4 is the sector (0, 1, 1)
    // 5 is the sector (1, 0, 0)
    // 6 is the sector (1, 0, 1)
    // 7 is the sector (1, 1, 0)
    // 8 is the sector (1, 1, 1)

    return ($x * 8 + $y * 4 + $z * 2);
}

static function get_sector ($x, $y, $z) {
    // 0 is the origin (0, 0, 0)
    // 1 is the sector (0, 0, 0)
    // 2 is the sector (0, 0, 1)
    // 3 is the sector (0, 1, 0)
    // 4 is the sector (0, 1, 1)
    // 5 is the sector (1, 0, 0)
    // 6 is the sector (1, 0, 1)
    // 7 is the sector (1, 1, 0)
    // 8 is the sector (1, 1, 1)

    return ($x * 8 + $y * 4 + $z * 2);
}

static function get_sector ($x, $y, $z) {
    // 0 is the origin (0, 0, 0)
    // 1 is the sector (0, 0, 0)
    // 2 is the sector (0, 0, 1)
    // 3 is the sector (0, 1, 0)
    // 4 is the sector (0, 1, 1)
    // 5 is the sector (1, 0, 0)
    // 6 is the sector (1, 0, 1)
    // 7 is the sector (1, 1, 0)
    // 8 is the sector (1, 1, 1)

    return ($x * 8 + $y * 4 + $z * 2);
}
}
```

More questions?

anna@futuregames.nu

@theanaka

* Gets the sector from the (x, y, z) specified

* Sector will be:

```
class GeoOctocube {
    //
    //          | 5 | 6 |
    //          |---|---|
    //          | 7 | 8 | 2
    //          |---|---|
    //          | 3 | 4 |
    //          |---|---|
}
```

param int \$x the x coordinate
param int \$y the y coordinate
param int \$z the z coordinate

* @return int the number of the sector (0 if x =

static function get_sector (\$x, \$y, \$z) {