this document contains the uptodate VERSION CHANGES

(last modification made 14.8.25 17:48)

1.0 5/1

House volume's 500x1200

walls have been built

1.1 20/1

New floor included (flat)

therefore 1st floor has become 2nd floor now

controls for moving and rotating element(per time) has been enabled

r: move up (++y)

v: move down (--y)

d: move to: left (++x)

f: move to: right (--x)

r: move forwards (++z)

v: move backwards (--z)

u: rotate up (++y)

n: rotate down (--y)

j: rotate to: left (++x)

k: rotate to: right (--x)

o: rotate forwards (++z)

l: rotate backwards (--z)

q: shifts between camera and current selected element

1.2 10/2

Made room for windows inside walls

In order to save last camera position/rotation ck.js has been

created, tested and included (library which stores cookies in the web browser instead of connecting to database everytime)

1.4 25/2

Mapped with texture: walls, windows, doors, 1st floor and 2nd floor

In order to reduce source code t.js has been

created, tested and included (library which parses incoming string/vector and returns geometric vertices (only supported by WebGLRenderer)

Stairs included

1.5 29/3

Controls have undergone some changes (for better)

in order to become navigation friendlier move.js has been

created, tested and included

scenario that is in front of you always will be your NORTH (even if you rotate)

r: able/enable camera rotation

while r is enabled -180deg (-PI/2) to the right and 180deg (PI/2) to the left

+/-: increase/decrease speed of movement (not rotating speed)

arrow keys: left/right go in x axis whilst up/down go in z

ctrl+arrow keys:

ctrl+up increases camera position in y (go up)

ctrl+down decreases camera position in y (go down)

since mousepad is blocked due to any arrow key (while being pressed), alt helps

alt: move ahead (perfect to navigate together with mousepad)

alt+ctrl: move backwards

shift+arrow keys:increse/decrease rotation of camera in 90deg (PI/2)

shift+left: ++y

shift+right: --y

1.6 10/4

if's been minimized:

before:

true?callback():0

now:

true&&(callback())

//:0 false||do nothing||garbage

explanation:

x=0?5:4 //x=4 because 0 represents false that's why x isn't 5

!=contrary, !0=true whilst !1=!2=!3... false

1.7 21/4

Nice transition added to rotation camera (shift+Arrow key)

Updated Camera' y axis after rotating //bug:(it was looking backwards)

1.8 29/4

OOP (Object Orient Programming) has been supported/implemented to all 5 libs, due to reduction of amount of variables in the code,

each library has its own object:

vectrix.js has v, ck.js has K, t.js has t, move.js has m, index.js has h //House/Home

in contrast it increases the amount of characters, though

// before: ck() now: K.ck()

Callbacks ad() from Lib. t.js function have changed since it only expected 1 parameter (string) therefore it'd to be called about 66 times

```
Scene.add(new T.AmbientLight())
Scene.add(t.ad('1 2[door/0]'))
Scene.add(t.ad('1 2[door/1]'))
Scene.add(t.ad('1 2[door/2]'))  //...66
```

but now, it expects a vector (wich may be filled with either string(s) or object(s) furthermore it adds its shape(t.s) to scene.

```
t.s=Scene
t.ad([new T.AmbientLight(),'1 2[door/0]','1 2[door/1]','1 2[door/2]'])
```

String variables containing Numbers passed to parseFloat // before: '5' now: 5

1.9 2/5

Since page's loaded, ck.js SAVES Current Camera's Position and rotation every onblur event (when it looses its focus or user shifts to another tab)

and when user decides to reload page

you can make yourself sure, presssing s it SAVES(Current Camera's Position & Rotation)

2.0 21/6

application cache has been tested, fixed, approved and enabled

benefits:

before:

time loading web page:~4s (localhost)

accepts offline mode:NO

update:NO, it has to download WHOLE web page (again and again)

now:

time loading web page:~2s

accepts offline mode:YES!

update: it only downloads user changes

2.1 1/7

in order to handle incoming assets xhr.js has been

created, tested and included

it instances a XMLHttpRequest, it requests first parameter plus .js extension if it's got one and if statusText is OK then it proceeds evaluate second parameter.

't.ad(eval(this.response))'

this.response will be vector enclosed in '' ready to eval

2.2 5/7

in order to reduce number of scripts (8) inside head, filler.js has been place there instead

once page's finished loading, it fills head on the fly with all necessary scripts and when it's done, it deletes any traceback on document.head's DOM

2.3 7/7

apache's been replaced with node.js' express server localhost:3000

and an application's been created with express

jade as html

stylus as css

/*2.4 18/7

!detected conflict with ck lib and all cookies ended up to NaN because it's suppost to only be cookies with 1-6 order stored therein but since now it'll (co-exist) look for them first and gathered them into vector (6).

updated rsz lib's cursor*/

2.5 31/7

in order to drag and resize an object, drgNrsz.js has been created, tested and approved; you can enable/disable it pressing m key.

2.6 16/8

ops.js has been added; it brings to user 10 basic controlling options:

1:add, //6 different textures each one holds 4x4 of its type

ceiling,door,floor,stairs,wall,window //finished on 25th

2:dragNresize //same as pressing m

3:change picture //finished on 24th

4:activate/deactivate rotation (while moving mouse)

5:rotate camera 90° to the left Π

6:rotate camera 90° to the right -Π

7:decrease light strenght (by -.15 til´ 0) //since Mesh´s got__

8:increase light strenght (by .15 til´ 1) //_MeshLambertMaterial

9:export house (as casa3d.js) //finished on 23rd

10:toggle fullscreen mode //finished on 22nd

2.7 9/9

favicon designed in rw-designer.com and updloaded to casa3d.hostzi.com