

To Do list

Code refactor

this is done by removing, all references to objects “s →” leaving the variables and replacing them with global variables or project variables.

leave the library find that code and put in the file and also refactor.

Get the code that is executabule and put in the .c file. Convert functions to a if and else, and a true state.

If lather is need is reconverted to a function, stuff that is less executed. The more exaustive code use will go to procedural, loops.

Debug tool

folder, debug tools

load all variables from header files and mount dependencys.

And print them to screen

Server HTTP

write a basic, server

<https://blog.abhi.host/blog/2010/04/15/very-simple-http-server-writen-in-c/>

A basic thing to evolve as planning is doing and file objcts.

Server TCP UDP

write a basic, connect server client

there is a totutorial for this in TCP and UDP

<https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>

A basic thing to evolve as planning is doing and file objcts.

Synchronize

to synchronize all people information,

Creating a file online, and the box connects and see if it have new stuff to update. Probably need to update online files and local files.

All stuff added should have a option to publish, so it will be available in synchronize.

We need to write a login for the user. Think the box it will work with a hash, so instead of people writting a password, user, email, it generate a unique hash, and people login width that. It is very secure and less problematic when finding what kind of login did you make this way you need to writte it down.

Game Lore tool

folder, tools

add history content to a “database”

search character creation web sites or software, add see more stuff to add

<https://forgottenrealms.fandom.com/wiki/Elf>

The end result should be something like this web page, a high elaborated background to add to games. So all this stuff will be automatically added to our game. Maybe create some kind of link : or in game lore tool or in projects.

The

3D debug tool

folder, debug tools

Refactor code to structure

load all the information from the 3D model to screen

Load 3D models

folder, stack

In lips of suna (i often call it lips of luna dexlexy, for me its lips of luna) they have a costume .blender 3D model loader.

Refactor code to structure,

A copy of Lips engine is in external projects folder on google drive

Do also the debug tool

Bitmap

load a procedural bitmap to the console

we need all code procedural since this is going to be part of graphics, is better to not have dependencys.

Interface

folder, application structures

Once we have the bitmap ready we can do the control and window, to start developing the interface.

Load images interface parts to console

Interface debug

folder, debug

load to screen all information in interface parts being loaded.

Image typs, position, layer, control, window, names.

Tutorial tool

folder, tools

load a tutorial by parts, info in the file.

Connect tutorials to files, A tutorial to A File.

List tutorials add

synchronize tutorials online width other users.

Projects tool

folder, tools
list projects (by theme or all)
add project
publish/unpublish project button
synchronize local, publish projects
synchronize global, get projects online list (by theme or all)
add lore to project
projects have more functionality but for now is enough, to evolve width objects and planning.

Torrents

Transform, web pages, projects, code and files to torrents.
Convert the torrents to usable files in the box
Update torrent files width information from which type.

The guardian

Monitor torrents activity
A simple firewall

Projects team

Create a team project
Add members
synchronize local and global
Files list
Upload files
Files by theme

Social tool

folder, tools
list web pages theme (by theme or all)
add web page
publish/unpublish pages button
synchronize local, publish page
synchronize global, get pages online list (by theme or all)

A text editor

Edit text and images to create a web page,
add to social tool
web templates to create auto pages.
Load images
create posts
active links to other pages
Boxes to move text and images in the page.

*Also works with market items.

Costume Game

Create a costume game
publish the game the IP or connection online
List costume games
List servers
list users

Messages

write a message system
all synchronize should get your messages,
for now is only kind of email, but may evolve to a chat.

Database

Create database
link database with project files

Mysql Server

At the moment don't have many information on (tutorials, server, code)
A local server to test databases
A interface on the database to link project files and databases
Auto generate queries
Test information from queries
To test information in the simulation environment, since project may not be compiled get the information from files.

Market

folder, tools
add a item
list items (by theme or all)
publish/unpublish item button
synchronize local, publish items
synchronize global, get items online list (by theme or all)

File Browser

Work directly on files and connect them to code.
Add work groups to visualize information organized

Project List

Load all projects in a form or text
edit items (3D models, images, functions, text, etc...)

Costume Function

create functions without touching code.

Part functions in to actions.

Choose which actions a function affect and generate function

Founding

list projects asking for founding (may be in projects also)

Click in a found link for project

Separate project in to budget parts

Attribute a value to which item.

Create tasks. With value

Project progress

Total budget

Individual tasks budget.

Font tool

Create a font

Load a font

Test a found in a template (maybe text editor)

Font tool debug

Output all information of the font to screen

Tournament manager

Create a tournament

Create a calendar

add users

add results

pay subscription

information on tournament (season, starting, price, number of players)

also manage commercial invent (work the same way)

list by theme or all

Stream

Publish streams to "costume games"

Create users group

Create a record screen software

(this have a very low priority don't know how complex it is. know it uses codes from videos and compression)

User

publish your user profile

login in

(most functions require login)

synchronize depends on login, record synchronize history with users name

Delete information from synchronize will only be available to administrators. (errors, or bad publish)

Language

folder, tools

load auto text to variables to have a language functionality.

Code editor

link functions

link files

the links see files code on in a houer menu type or open files width links

support for the C language.

Compile in the C language

Configure the GCC for the box

Compiler

Create costume make files, in the interface

See dependencys

add dependencys librarys

Add operating sytem (it will be not that easy, maybe lather)

Animation

generate auto animations from templates

edit animations

3D models auto

add figure (human, creature, object, vehicle)

Map the lines of a figure

Create costume objects based on figures