

The Box Forum Posts

Ideas debated in forums

Some things where outdated was project evolve others was and stayed accurate. Is more for people that like to debate ideas.

character generator

By having the character generator we can see how the structure works, now we can generate all aspects of our character, by creating the files for the character.

You can see in the pictures the Tool will create or handle the dependency's for us. So instead of us hard code a character files, which lead to bugs and errors the tool will generate the files standardize for us. Helping in 2 ways, one is a "tool" it self like name generator imbued and the other the files generation automatically.

The file output then can be done to any kind of language and in any kind of format for example: Unity3D C# language and in array's.

It will be more easy to work, because for example you do not need to enter the engine to work. You can go to your character, and let's say update name, strength, and dialog's. Click and is done.

Let's continue with the development, back to the theory :

[b]How to handle dependency's ?[/b]

If we use a standard C library with 1700 lines of code or something like that to handle the library, that stuff is loaded, all in to a scene or a game level.

Let's say that our main menu only uses JPG, then we need to load JPG's, PNG, PSD, and so on so if we handle this stuff using this principle. We load the image.c and the jpg.c which handles the necessary stuff and only loads the necessary stuff.

[attachment=0]example_dependencias.jpg[/attachment]

*Not to do things this way causes problems, which were debated in the first part of the posts, a previous example : like if you have memory problem. It will be easy to see the dependency's of the JPG library because you only need to read the image.c. While if you have a mixed library it may break stuff start altering things around. Which translates in a lot of hours of coding and debugging.

[b]How to handle variables ?[/b]

For example why to re declare pointers all the way in the project when we can have generic pointers declared :

```
int *int_pointer;  
char *char_pointer;
```

```
float *float_pointer;
```

When we need to use them inside our files, we know that the structure have generic declared pointers. This is used in? A form of structure, so the template have this stuff build in. In your file you access the int_pointer.

Pointers

Pointers are just a address to a variable they are not, a variable this means that it loads less stuff.

```
int large_number = 00987987987989890;
```

```
*int_pointer = large_number;
```

if we use our *int_pointer to pass it to a function, we don't move the variable around this is where we are saving resources. In our structure, we are saving lines of code by not re declaring stuff and by having pre build in variables.

Another pre build variable :

```
screen_width = 640;  
screen_height = 480;
```

variables that :

why should we re declare this stuff which time we start a project?

This stuff should be a standard variable.

We know that no matter what all application's use a screen_width, and screen_height unless we are developing more complex stuff for example robots, but in that case we can have 2 templates, 1 for applications and 1 for direct accessing code.

Once again we find to benefits, one is this stuff is going to be used or reused again. Another thing is the standardization, if we need to declare or import a function lather, it can be a function generic because it is accessing generic variables.

Preventing of declaring new stuff saves, coding time. Because lather you have to handle the dependency's. If they are already set this minimizes mistakes and error's inside the code.

To use this variable it not necessary to be the language it self often can be a database to make a automatic application for example, it can use the code template. So it save us development time to have pre build in templates.

People are starting to explore this kind stuff now with the : bootstrap. A CSS kind of framework or standardization that reclycle's stuff. If you want to know more about it can find it in:
<https://www.w3schools.com/>

"I found this stuff a couple of years ago, when i was designing my PHP framework before . "

[b]How our main file could look like ?[/b]

It looks like a OOP object with the exception that all things are generic and accessible. The difference is that a variable or a library is only needed several "layers" down or in another application is loaded there. If it generic and most of the things or several files need to load this stuff is moved to the top level.

```
[code]/* FUNCTION LIBRARY */
```

```
//#include <errno.h> // handle erros, not stable
#include <stdio.h> // header defines three variable types, several macros, and various functions for
performing input and output.
#include <stdarg.h>
#include <stddef.h> // ptrdiff_t on osx
#include <stdlib.h> // Marcro extension library
#include <string.h> // header defines one variable type, one macro, and various functions for
manipulating arrays of characters.
#include <limits.h> // limits the values of various variable types like char, int and long.
```

```
//provides a macro called assert which can be used to verify assumptions made by the program and
//print a diagnostic message if this assumption is false.
#include <assert.h>
#include <arm_neon.h>
#include <windows.h> // Handle the windows interface
```

```
/* GAME LIBRARY */
```

```
#include <file.c>
#include <spells.c>
#include <character.c>
#include <game.c>
#include <resolution.c>
#include <movement.c>
#include <keys.c>
```

```
/* GAME VARIABLES */
```

```
bool admin = 0 ;
bool game = 0;
int done = 0; // The window is open: enter program loop (see SDL_PollEvent)
screen_width = 640;
screen_height = 480;
```

```
/* ===== Pointers ===== */
```

```
int *int_pointer;
char *char_pointer;
float *float_pointer;
FILE *file_pointer; // This declaration is needed for communication between the file and program.
```

```
// Debugging and re-factor or re-writing code.
```

```
int main(int argc, char *argv[]){ // argc, e argv default?
```

}[/code]

The dependency theory part II: File Manager

We debated the A.I. in previous "chapters", or posts, so now we can see that applying in some way.

The dependency's can be lather altered and moved around width a program, the concept is still very rudimentary, but can lather add more stuff like : the AI structure. Maybe a AI manager that will handle file manager and variable manager at another lever for us to minimizing errors.

We could handle the dependency that our programming structure have with, for example: we need to remove JPG dependency for our program.

The program should or will allow to add and remove dependency snippets, like grab and drop tool.

Can you remember where all those dependency's are? Probably not, even if you do you still have to search all files by hand inside which is a tedious task to do.

Lather on if we need to add or remove stuff, it can be done with a couple of clicks and not search inside files, or remember where stuff is. Which withdraw any possible help that a person can get in to a project, because not all people are developers of the code.

That is separate in to 2 ways if we can say that. People that are familiar with the internal code and can "move around" and others that have no knowledge of the internal and just want to extend some kind of functionality.

Well to extend they need to know how the internal is working in order to know the dependency's. In this case we have to 2 ways of doing that one with the external program, and another inside the code. In a way the 2 are inter depended of one another. If that doses not happen inside the code can't be lather automated in the program, because it have no knowledge of that. It need some kind of "giant" AI memory structure just to do that.

In another way if is structured internally can be managed out side, at least in a easy way.

[attachment=0]file_manager.jpg[/attachment]

"this stuff remove redundancy and errors from the code and allow automation in creation templates and game projects "

Follow along Post :

What are modules :

<http://www.dummies.com/programming/c/how-to-link-2-source-code-files-with-c-programming/>

Strings :

This sounds a bit redundant in theory but when you get to practice is a completely different thing from other languages.

"Pointers in a way are a loose way of using strings."

Basic usage of string - https://www.tutorialspoint.com/cprogramming/c_strings.htm

More on string usages - <https://softwareengineering.stackexchange.com/questions/183842/is-initializing-a-char-with-a-string-literal-bad-practice>

Comparing usages - <https://stackoverflow.com/questions/8732325/how-to-declare-strings-in-c>

A Generic Linking of files :

<https://stackoverflow.com/questions/1433204/how-do-i-use-extern-to-share-variables-between-source-files>

User Feedback Replay

The dependency's in C works just fine as they hold do. PHP if build in C strangely if it not work also in C.

Main.c :

```
[code]#include <stdio.h>

#include "variables_interface.c"

int main(int argc, char **argv){ //

    //include "interface.h"

    int interface = 3;

    //printf(a);

    //include "interface2.c"

    if(interface == 2){
        //printf("Interface2");
        #include "interface2.c"
    }
    else if(interface == 3){
        //printf("Interface3");
        #include "interface3.c"
    }

    return 0;
}[/code]
```

interface2.c :

```
[code]#include <stdio.h>
printf("Interface2");[/code]
```

interface3.c :

```
[code]#include <stdio.h>
printf("Interface3");[/code]
```

variables_interface.c :

```
[code]int a = "texto do ficheiro variables interface";[/code]
```

This concludes : The scope of a library in a header format or a .h file is just the file it is attached to.

The "#include <stdio.h>" to make the printf work though is included on main file did not work in other included files.

But the #include "variables_interface.c" , "a" variable was available in our interface files.

In order for the printf to work it needed to include a <stdio.h>, in separate...

But if you include a .c file in to another .c file scope or environment of the "stuff" loaded is all the application. In another way: is accessible the main file where it is included and in sub included files.

Think: To replace standard library scope just need to replace the extension of those .h files with .c. Don't know if they are accessible to public or open source. But if they are don't see how it is a project not doable...

"removed 200 or 300 lines of code in redundancy. If some one new need to study this will make a difference?"

[b]Implementation theory[/b]

note : found this project which may help put more open source project in to the The box folder. Later when i have time i will add more projects.

<https://osgameclones.com/>

"My fighting game which i'm developing for example found open source fighting game in there "One most fall". A game with a lot of fans..."

A replay to lyberta and samerson this guy at libre engine have a good one for you people:

[quote]Duion {I Wrote}:
Why you want to develop another engine?[/quote]

[quote]Farrer {I Wrote}: And why not? ... Keep it up climjark, and don't let those "why doesn't you do what I think is better for you to do" discourage you. Do what *you* believe is worth for you to do. [/quote]

Let's analyse this SDL code to declare a window. This code in theory looks grate, and this people at SDL probably put a lot of afford in to developing it. But is not that good for our program, because it work's in a kind of pipeline, and this code already start to nesting dependency's.

```
[code] SDL_Window *window;           // Declare a window
SDL_Renderer *renderer;             // Declare a renderer

SDL_Init(SDL_INIT_VIDEO);           // Initialize SDL2

//Create an application window with the following settings:

window = SDL_CreateWindow("Game Window",           // window title
                           SDL_WINDOWPOS_UNDEFINED, // initial x position
                           SDL_WINDOWPOS_UNDEFINED, // initial y position
                           640,                     // width, in pixels
                           480,                     // height, in pixels
                           0                         // flags
                           );
renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);

SDL_RenderSetLogicalSize(renderer, 320, 240);

SDL_Surface *sheet = IMG_Load("sheet.png");

if(!sheet)
{
    printf("Cannot find sheet\n");
    return 1;
}

man.sheetTexture = SDL_CreateTextureFromSurface(renderer, sheet);
SDL_FreeSurface(sheet);

//load enemy

sheet = IMG_Load("badman_sheet.png");
if(!sheet)
{
    printf("Cannot find sheet\n");
    return 1;
}

enemy.sheetTexture = SDL_CreateTextureFromSurface(renderer, sheet);
SDL_FreeSurface(sheet);
```

```

//load the bg

SDL_Surface *bg = IMG_Load("background.png");

if(!sheet)
{
    printf("Cannot find background\n");
    return 1;
}

backgroundTexture = SDL_CreateTextureFromSurface(renderer, bg);
SDL_FreeSurface(bg);

//load the bullet

SDL_Surface *bullet = IMG_Load("bullet.png");

if(!bullet)
{
    printf("Cannot find bullet\n");
    return 1;
}

bulletTexture = SDL_CreateTextureFromSurface(renderer, bullet);
SDL_FreeSurface(bullet);

// The window is open: enter program loop (see SDL_PollEvent)
int done = 0;

//Event loop
while(!done)
{
    //Check for events
    done = processEvents(window, &man);

    //Update logic
    updateLogic(&man);

    //Render display
    doRender(renderer, &man);

    //don't burn up the CPU
    SDL_Delay(10);
}

// Close and destroy the window
SDL_DestroyWindow(window);
SDL_DestroyRenderer(renderer);
SDL_DestroyTexture(man.sheetTexture);
SDL_DestroyTexture(backgroundTexture);

```



```
SDL_DestroyTexture(bulletTexture);
SDL_DestroyTexture(enemy.sheetTexture);
```

```
for(int i = 0; i < MAX_BULLETS; i++)
    removeBullet(i);
```

```
// Clean up
SDL_Quit();[/code]
```

A example of the "pipeline" which is created :

```
[quote]/* SYSTEM LIBRARY */
```

```
    #include "error.c"
    #include "string.c"
    #include "paths.c"
    #include "file.c"
    #include "database.c"
    #include "image.c"
    #include "image_png.c"[/quote]
```

The "file.c" generates errors? If it does then error.c must come above "file.c". Another example : "image.c" uses "Paths.c" ? Then it must have available paths. It uses a inheritance system.

A example of paths file :

```
[code]
char path;
char string_length;
int path_length;
int extension_length;
char extension;
[/code]
```

Once is initialize other files will have a path variable available. That removes the need to passing variables in to functions. Which at first look does not seem but it creates a lot of redundancy.

```
"IMG_Load("bullet.png");"
```

This function is not standardized with the "pipeline" or inheritance which means we can't decompile, simplify or remove redundancy later. That's the main problem in using other engines library's code. Though I mention in the beginning. It is in theory in practice it may not work that well.

Another example :

```
[code] /* SYSTEM LIBRARY */
```

```
    #include "colors.c"
    #include "image.c"
    #include "image_png.c"
[/code]
```

Let's say we are inheriting a color structure. Then it can't inherit so it will be very complex to work with pre build library's that are not standardize with this way of inheritance. They do too much inside one file, or they do too little and receive no inheritance.

This will lead to more code? It leads to less code the so called dynamic code, it uses much more stuff. For example the variable of the loop : "for" which people seem to like "i" to represent a increment, is declared once in main and all files receive that variable. I was reading through "leaps of suna" code and they declare this variable hundreds of times, for example.

```
[code] /* main.c */  
    int i;  
[/code]
```

Without variable inheritance you can't write if and else on the and they handle a lot of stuff. The less functions you use the better less stuff goes to memory.

Our code it will end up by looking like files with a simple if and else statements structure with only 100 or 200 lines of code. But in case of C that may be a bit bigger since is lower level language.

Updates :

- Changed the licence to GPL

How the Software works?

It uses a "mapper" which is a feature above OOP. Basically it maps all our application, so when we write the code once it can be used in OOP or in procedural. The good about this is it can develop a whole application in assembly with a fraction of effort and the good performance aspect.

[attachment=1]the_box_mapper.jpg[/attachment]

In first picture we can see the buttons : Import, synchronize, settings.

If you have a project it will attempt to create a map out of it. For that you select a supported language, for example C, it detects how variables and functions are written, and start creating the map for us.

Or synchronize, we do some updates, and update the project, the map is automatically saved. But think could put a button to differentiate this. Save map.

[attachment=0]the_box_mapper2.jpg[/attachment]

In second picture we see the interface in action it can contract, which will allow for people on androids and gadgets to code.

In the create you manage the whole application by selection stuff.

The internal code of how to map the application. It needs to know where is all the stuff in the application. For example loading order, associated files, associated variables, etc...

Some pseudo code i did in PHP, need to convert it to C.

```
[code]Templates_icons = array(

"spells" => array(

"1" =>

array (
    "name" => "Magic spells by 1"
    "license" => "CC",
    "autor" => "Kifan",
    "sizes" => "246x, 128x, 48x",
    "icons" => array(),
),

"2" =>

array (
    "name" => "Magic spells by 2"
    "license" => "CC",
    "autor" => "Tuzgeg",
    "sizes" => "246x, 128x, 48x",
    "icons" => array(),
),

"abilitys" =>

"1" =>

array (
    "name" => "RPG spells by ..."
    "license" => "GPL",
    "autor" => "CC",
    "icons" => array(),
),

"2" =>

array (
    "name" => "RPG spells by ..."
    "license" => "GPL",
    "autor" => "CC",
    "icons" => array(),
),
);
```

```

languages = array(

"C" => "1", (between code);
"C++" => "1", (between code);
"Assembly" => "1", (between code);
"PHP" => "1", (between code);
);

default_values = array(

"number_of_lines" => "1", (between code);

);

process_array = array (

"3" => "variable_mapper",
"23" => "function_mapper",
"33" => "file_mapper",
"4" = "database_mapper",

)

variable_mapper = array (

"1" = array (

"name" => "npc",
"type" = "int",
"default" = "",
"starting_line" = 200,
"ending_line" = 201,
"process_position" => 1,
"connectios" => "",
"files" => array ("main", ),
"functions" => array ("combat",),

),

"2" = array (

"name" => "game_loading",
"type" = "bool",
"default" = "0",
"starting_line" = 200,
"ending_line" = 201,
"process_position" => 2,

```

```

"connectios" => "",
"files" => array ("main", ),
"functions" => array ("combat",),

),

);

```

```

file_mapper = array (){

"process_position" => 2,
"name" => "npc",
"connectios" => "",
"files" => "main",
"functions" => "combat, ",

}

```

```

function_mapper = array (){

"1" = array (

"process_position" => 3,
"name" => "npc",
"connectios" => "",
"files" => "main",
"variables" => array(),
"functions" => "combat, ",
)

}

```

```

database_mapper = array (){

"process_position" => 3,
"name" => "npc",
"connectios" => "",
"files" => "main",
"functions" => "combat, ",

}

```

Files -> variables & Functions & DB's

```

DB Mapper = array (){

"name" => "npc",
"connectios" => "",
"files" => "main",
"functions" => "combat, ",

```

}[/code]

How to create a library?

It should be the more specific why possible so load less and more accurate in to a project. For example spells for sci-fi, or spells for medivel RPG, actions post apocalyptic.

This whay stuff will be a bit more organizes.

THis is in case some one want to help organize stuff for example of open game art, in to librarys.

Reverse engineer?

If you study the laws is possible to reverse engineer on any thing you wish, it's legal. But you have to "remaster" it completely to be legal.

Let's say i want to reverse engineer Quake 3 GPL engine it's possible, but to be legal i have to re-write the code, but it can use it as a base for my project.

The first task for the project : will be to create the window, to start mounting the "planning" which is posted.

A tutorial on the window : http://www.winprog.org/tutorial/simple_window.html

The "planning" also uses 3D for that:

This page have update tutorials, on 3D graphics coded directly, for people who is to learn how to code. I was searching for it for some time but only found old books. In where there are highly updated information.

<https://www.3dgep.com/category/graphics-programming/directx/>

If some one have suggestions, on assemble the first part: a window, with 3D support, to load the basic stuff in to the window, and assemble simple stuff, like a name generator, that automatic, generate characters for us, etc...

[code]

```
extern int counter_1;  
extern int counter_2;  
extern int counter_3;  
[/code]
```

For what i have read :

The "extern" extension associated with the variable is what allow to link the variable with other files. So that C can work in a pipeline.

"Placing a non-extern variable definition outside the scope of any function makes it available anywhere within that program segment. It would need to have "extern" in order for other program segments within a full-fledged program to see it."

Some one posted "C does not work like that". Guess it does work. Programming language are modular they work in any way you want?

In the image we can see Lips of Luna code, it repeat's it's variables in which of it's functions.

8 lines of code, in 20 functions that's 160 lines of extra code. And passing variables to functions? another big problem...

[attachment=1]r.jpg[/attachment]

code repeated 20 to 30 times just in one file. If we are doing something like Quake engine that's "10.000" extra lines?

[code]

```
int x;
int y;
int a1;
int a2;
uint8_t* src;
uint8_t* dst;
LIMatRectInt src1;
LIMatRectInt dst1;
```

[/code]

With this style we place our counters (x,y,etc...) on our main file program and that's it we will no longer need to re declare any counter's.

How our main file will look like:

[attachment=0]r2.jpg[/attachment]

[code]

```
// ===== DECLARATION =====
```

```
extern int counter_1;
extern int counter_2;
extern int counter_3;
```

```
/* GENERIC Pointers */
```

```
extern int *int_pointer;
extern char *char_pointer;
extern float *float_pointer;
extern FILE *file_pointer;
[/code]
```

Counter's and Pointer declared only once and written only once.

Structural programming is OOP, the difference is: it's in the files and not in object's.

The problem is not to build a 3d engine. The problem is the consume of time, when you need to make changes, to variables and stuff, called refactor code.

I adopt this approach so that in procedural code, (high performance and simplicity of code) the code is maintainable and you don't need a very high skill and probably high mental capable person (genius) to maintain the code.

Once the main file is written, we have a structure, to add pieces. It's no longer needed to re write stuff (consumes a lot of time).

This code is only written once, and if we need it again we can use the same code. It removes redundancy or repetition.

Now are we looking in this projects?

Something like this file:

<https://gitlab.com/KilgoreTroutMaskReplicant/loom/blob/master/src/mouse.c>

Well written, with variables on top. It is not documented, it will be hard for any one coming to this file to understand what's going on. But the principle is there.

Clean code that we can work with.

Document the code : what connections is using, what and why is using some library's, if possible not to much explain the code (not redundant). If is to much redundant i will become also very hard to work with. Because you are reading the code and the documentation.

If is not written this way then the process will be to refactor, parts of the code to look like this way. Simple organized and documented (simple). To make it more easy to work with and learn.

Forget a step in the planning of the software :

It need a listing of the stuff that we added to the game. Thought that is needed to see it in 3D but thinking better is not 100% need. The software can work only was a linker of stuff. It already removes a lot of work for people. Is not needed to touch the code for this. But to do complex editing will be needed to load models in to the software.

For simple stuff : Unity3D for example, if you add the files to the assets folder when it loads it handles them. So is not 100% necessary to manual import them... But the software can be configured to edit the files manually to work with any other software. Removing the need for artists/editor's to know code and touch the code.

Can add/edit things in the game throw software only (interface).

[attachment=0]listagem.jpg[/attachment]

We can see in the image all the characters that we added to the game.

Think i want to lunch the software in a Linux packaged format.

So when the users download the software it already have a few games to tweak. This will be depende on game administrator if they want to cooperate it will take to much time for me, to get to know all this games.

A possible list :

- Ancient Beasts
- Open Dungeons
- Lips of Luna
- Masters of orion
- ...

don't know what other games are active, but it could be a large list. Or when the user install the software we can see a list of game that work with "The Box" and can download them.

Ancient beasts is a online game, but the software can come with a Apache and run it locally, so people can play around,
Don't know how Open Dungeons works but think is a local game.

Will see if clone games can be included in to have a large game list.

Also develop more tools to help out with this.

I was a Ultima Online player for good amount of years. Now if i want to play don't know where the community is.

Maybe create a server tool : which we can add and remove servers. Not just for one type of game but for all games :

Something like this status:

Unreal target (game) Lyberta (administrator/s) offline (status) 2 days (time to be active/reseting) .
Ultima Online (game) Lyberta (administrator/s) offline (status) 1 days (time to be active/reseting) .
Ancient Beasts (game) Lyberta (administrator/s) online (status) 0 days (time to be active/reseting) .

If you have or want to create costume server, like Ultima Online or Ancient Beasts, we can add it to the list and people will know where your server is and how to connect.

This will allow for example when a user download Ubuntu see the software, and install get a access in to a large working package. Think it will be nice for Linux users, to have a short cut in the exploration of open source games, and it will be nice for the developer which will have a much larger community for their games which could represent more help and support to this projects.

The tool list (debate):

Servers (mention)
Debugg (mention)
Database (make some games that are online work with it)
Name Generator (robust name generator included, to help create characters auto.)
WEb Browser (local server)
Tile Manager (i will be nice to have also a tile manager working, so to load imagens/icons and check how they look)
Map Manager (debated already)
3D editor (based on shapes, debate)
... (not much time for this now)

Was managing Lips of Luna project, and realize it consumes a lot of time if things are not organized, since i worked with Adobe Bridge which is an advanced browser or explorer files tool i thought on adding and developing a tool that do something like that to "the box". It can administrate files, do several projects in the same, time and do more elaborated tasks, display more information. Etc...

(images are compressed under 100kbts)

Several channels file browsing

How to do the automatic programming? think i come up with a solution to do it.

Think on "actions" :

You choose, a action that want to do and is supported by the "programming simulator enviornment", let's say you want to get all your spells from the database in a mysql extension. Select the action for that, and it loads a menu that have the pre build in functionality in to the interface. Only readable text (not programming functionality)

So is modular enough to handle all kind of actions, and simple to program, and support several languages. Then the software recognizes you choose C, or PHP, get the rules to build a command like that : PHP -> mysql -> db -> spells -> call) and assemble the call automaticaly.

How all the open source project's could be loaded in to a Database and then loaded in to the software. Width specific details about them, who develops, what the community think about the project (ranking), download them auto (create a command tool to do this), auto compile for linux, etc...

Any suggestions on how to improve are welcome.

About the post's : the images are extremely compressed but if there are any problems with the server think i can move the project to a Facebook page, if is causing any problems.

[attachment=1]projeccts.jpg[/attachment]

Was for the compiler tool : (make files auto) is something interesting to automat, in this case i have no experience with building files for Linux i did it 5 or 10 times. Most of the times was running only a command. To manage them with files programming, don't know what will be needed to do it. One think it needed to be connected to the GCC probably to be able to create the .exe or execute in Linux. I think it compiles differently in different Linux distro's. And the make files is just a linker. It is linking stuff. So the management to be done in the software is something like that, manage files to be linked. (guessing)

[attachment=0]compiler.jpg[/attachment]

This week i was extremely annoying with publicity, like in a 1 hour video they put 1 publicity in which song. In a concert of 1 hour, 20 or 30 musics you need to listen to 20 or 30 publicity'. This in Youtube and Facebook is not getting better... (ad bloquer not always work, some times they disable the web page if you use one.)

Then I thought why not add a social tool to the box. When you develop something you can publish and be directly online, and the followers or other users can see what you post, like a internal internet or a mini network.

A display page listing users (follow, contact, page, tags, etc...)

[attachment=0]social_tool1.jpg[/attachment]

A display of a user page

[attachment=1]social_tool2.jpg[/attachment]

How this works ?

A torrent server, to connect people and users upload stuff to other users. Some text, and images, links (maybe videos) with today internet connection will be no problem.

How to implement?

Will be needed a database to mount the structure of the page (style), link users, etc...

benefits ?

We can create our social group free from publicity and spam and administrate the way we want.

Theory for a standard level editor and open source map format

On thing that intrigue me is why there isn't a standard or open source format for the level map.

It's a very repetitive task to place stuff by hand, then to leave the engine and lose all that work. Do all that stuff again, annoying?

We could have a map in Unity 3D, Unreal E., blender and etc... that could save some serious development time. It probably increases also production by allowing several people to work in the same project with different programs. Maybe could also extend the functionality of 2D tiles render.

The format could do more stuff like save the object's position by numbers. We could load a map with a similar object position structure, and update only then necessary models and position's.

That "decompilation" or removing the dependency's of the objects it self with the position of the object could increase the production time.

For example we could see this files:

- > map editor -> standard format -> links to objects
- > map editor -> standard format -> links to object position

standard_level_editor.jpg

This also could save a lot of trouble since we no longer need to import things to our project, instead our project works with the existing library's.

Add the files : main_menu_GTK.c, gtk.h, characer_generator_RPG, inerface_main_menu.xml, information.txt, creator_tool.

main_menu_GTK.c -> to start which will just attempt do load a starting menu. Maybe it will be nice o have some options to start with.

gtk.h -> a costume delegator to main_menu_GTK.c and lather maybe more functionality (Variables and function delegator).

character_generator_RPG.c -> this will have a lot of : for example races, let's say we can come with 100 races. though on adding a rooster to races, so it will be easy to create new races. Also will have all kind of stuff in a

character_generator_RPG.c -> If some one need to mount a characer or a team will have every thing in there, just click and it will load all that stuff in to the files. Also can also be used to add NPC's to the game. Maybe link races, to a raw 3D model and a 2D image. Rooster and if possible a 2D race image.

inerface_main_menu.xml -> GTK can load xml files in to interfaces. I was reading the manual, but a lot of stuff is in code directly.

information.txt -> the jason or xml files are very heavy, in terms of reading a file with 12.000 lines. It didn't load in to my PHP server. So thinking in creating a simpler set of tags, then create the rules in C, to work as a xml or a jason file but in a .txt file. So we can have a file that accept text but is much more light heigh in loading. (Maybe add less characters to the files "tags").

creator_tool.c -> At this stage don't know how practical is to add stuff automatically to files. But what will do is add for examples races to files.

https://de.wikibooks.org/wiki/GTK_mit_Builder -> Found the manual on GTK3 is in German but it can be translated easily. (google translate or Crome translate page)

A "First" delegation to main:

FILE * FILE_pointer -> This line repeat in several places. It needed to be inherited. The explanation on how this works can be found in there. Still i documented on main.c, if some one look at the code: know what is doing. -> <https://www.cs.bu.edu/teaching/c/file-io/intro/>

Things to do:

- > adding races, if you need to add a race, just go in to he file, character generator rpg -> races -> races . (spells, items ...)
- > linking races to models, (at the moment didn't though on how to do this)
- > loading menu (options for : sound, shortcuts, load a game or the software).
- > Delegate to main (a bit abstract but this is how it works, our structure inheriting when possible, avoid repetition, principle of DRY can read about it).
- > create_tool.c (add rules, to add stuff automatically to files)

files and files locations:

game_information/elf.txt : information on the elf race rooster, will serve was a base to add races. taking the information out of this site was example : <https://forgottenrealms.fandom.com/wiki/Elf>

interfaces/main_menu_GTK.c : a starting or loading menu.

tools/character_generator_RPG.c : default info to add characers.

external_librarys_structures/gtk.h : a delegator to GTK functionality.

interfaces/interface_main_menu.xml : GTK load xml to interfaces.

tools/creator_tool : add information to files automatically, maybe do other stuff like linking (models, images).

optional Links :

<https://www.aidedd.org/dnd-builder/index.php?l=1> -> some information on how to build characters.

The initial objectives, are very simple:

Not going to start by re factor C library's, engines, or stuff like that because that is a lot of work to be done. Though in theory that is what i have in mind to do lather, apply the same principle for the language it self. But if people see it working on game functionality's first or other stuff think it will be more easy for them to understand how to apply that to a programming language.

Going to start simple by developing the "Character Generator", to handle a couple of things automatically, and see how things are evolving...

The previous picture (Character Generator) is missing :

Lists : all your characters that you have created. A section to load characters and to be able to edit them.

Location : where the character is or will be located, and to be able to move them to another location.

3D model management :

At this stage there is no 3D model management, but it will be nice in the future to have a working engine with the tool to be able to load and manage also the 3D model. But without the engine it can only be done by name. Still for example, get the list of 3D models and link one in to the character. But will have no clue if it's working, with the engine we can see if animations, spells, abilities, etc... are working and link new ones or remove functionally from the character.

[b]External Librarys : gtkradiant[/b]

People were debating the relevance of using C in these days. Here we have a actual and good open source project, which can be used in a stand alone or some parts could be integrated in to The Box Tool.

<http://icculus.org/gtkradiant/installation.html>

Converting gtkradiant map to unity 3D.

<https://www.youtube.com/watch?v=n1lZvDmh1Mg>

Git Hub i added a copy to the project, any way to project folder.

<https://github.com/TTimo/GtkRadiant>

Another project to take a look also added to the project folder the source of Quake 3:

<https://github.com/ioquake/ioq3>

Found this tutorial on mysql in C, Mysql since have this nice procedural style.

<http://zetcode.com/db/mysqlc/>

[code]

```
int main (int argc, char** argv)
{
    MYSQL mysql;

    if(mysql_init(&mysql) == NULL) {
        printf("\nInitialization error\n");
        return 0;
    }

    mysql_real_connect(&mysql,"localhost", "user","pass","dbname",0,NULL,0);

    printf("Client version: %s",mysql_get_client_info());
    printf("\nServer version: %s",mysql_get_server_info(&mysql));
    mysql_close(&mysql);
    return 1;
}
```

[/code]

Don't have the database module file done, still.
it may look something like this :

default values which will allow to tweak all the database settings with out the need to know how to code or to do alterations to the code.

```
[code]
// ===== DECLARATION =====

extern char database;
extern char database_password;
extern char database_user;
extern char database_host;
extern int database_column;
extern int database_row;
extern int database_ret;
extern int database_size;

// ===== DEFINITION =====

extern char database = "box";
extern char database_password = "";
extern char database_user = "test";
extern char

database_host = "localhost";
[/code]
```

Think i will opt to start having something working before go in to heavy code refactor, that way the software is already working and doing something... (lather analyse that better...)

I will opt with GTK+ it seems pretty simple in this tutorial, (only a couple Labels), for the menu. Is open source, can develop a very powerful application (2D).

It have that procedural linking style, which i like to work with, and is very simple.

This cover the loading of images in to the software, "external library"s.

If you want to follow there are a tutorial in the video. With the GTK can start building the menus for the program, and putting something working.

<https://www.youtube.com/watch?v=ajNvsv1ka4I>

For the 3D think i will use OpenGL, because it uses the same principle was C, it's cross platform, and can work width a minimal graphic quality.

For the OpenGL it seems is necessary "Glut", it have a similar style was GTK.

https://www.youtube.com/watch?v=3aJ8OR1C6pk&list=PLWzp0Bbyy_3jy34HlDrEWlcG3rF99gkvk

Maybe can use both Glut+GTK to have more functionality available.

Also remember another thing that we could add to the software is a debug, implement some kind of test's so we can minimize more the margin for erros which translates in more development time.

For who is following along we are at GTK, they recomend to install msys2 if on windows (don't imagine many ppl using linux):

<https://www.gtk.org/download/windows.php>

Install msys2

<http://www.msys2.org/>

then GTK recomend we get glade.

<https://www.youtube.com/watch?v=vOGK3TveDDk>

which i check what it is, is a graphical software to develop the app so we can develop something whitch just a few clicks.

[b]code[/b]

index for arrays in C?

We create 2 arrays one is the index and the other is the arrays, so lather it is coded in to the tool so it handle this for us so that it prevent mistakes.

For example :

```
[code]char spells_index [1][4] = {  
    {"Name", "power", "mana_cost", "time_casting"}  
};
```

```
char spells [10][4] = {  
    {"agility", "30", "10", "300"} ,  
    {"..", "..", "..", "..."} ,  
    {"..", "..", "..", "..."}  
};[/code]
```

You the see the connection, 0 position is the name index. while 4 or last position is the time casting.

Settextstyle function in c

Settextstyle function is used to change the way in which text appears, using it we can modify the size of text, change direction of text and change the font of text.

Declaration: void settextstyle(int font, int direction, int charsize);
font argument specifies the font of text, Direction can be HORIZ_DIR (Left to right) or VERT_DIR (Bottom to top).

Different fonts

```
enum font_names
```

```
{  
    DEFAULT_FONT,  
    TRIPLEX_FONT,  
    SMALL_FONT,  
    SANS_SERIF_FONT,  
    GOTHIC_FONT,  
    SCRIPT_FONT,  
    SIMPLEX_FONT,  
    TRIPLEX_SCR_FONT,  
    COMPLEX_FONT,  
    EUROPEAN_FONT,  
    BOLD_FONT  
};
```

C programming source code for settextstyle

```
#include <graphics.h>
```

```
#include <conio.h>
```

```
main()
```

```
{  
    int gd = DETECT, gm, x = 25, y = 25, font = 0;
```

```
    initgraph(&gd,&gm,"C:\\TC\\BGI");
```

```
    for (font = 0; font <= 10; font++)
```

```
    {  
        settextstyle(font, HORIZ_DIR, 1);  
        outtextxy(x, y, "Text with different fonts");  
        y = y + 25;  
    }
```

```
    getch();
```

```
    closegraph();
```

```
    return 0;
```

```
}
```

<https://pubs.opengroup.org/onlinepubs/7990989775/xsh/readaddr.html>