Proiect electron

Interfata

Obiecte

Masa de lucru

Un tab de L si l, pe care il vom imparti:

* ¾ L chenarul de lucru, culoare #20421E
* ¼ L chenarul cu piese, culoare #1E4546
* ¾ L si 1/5 l chenaru de erori, culoare #4472C4

Erori

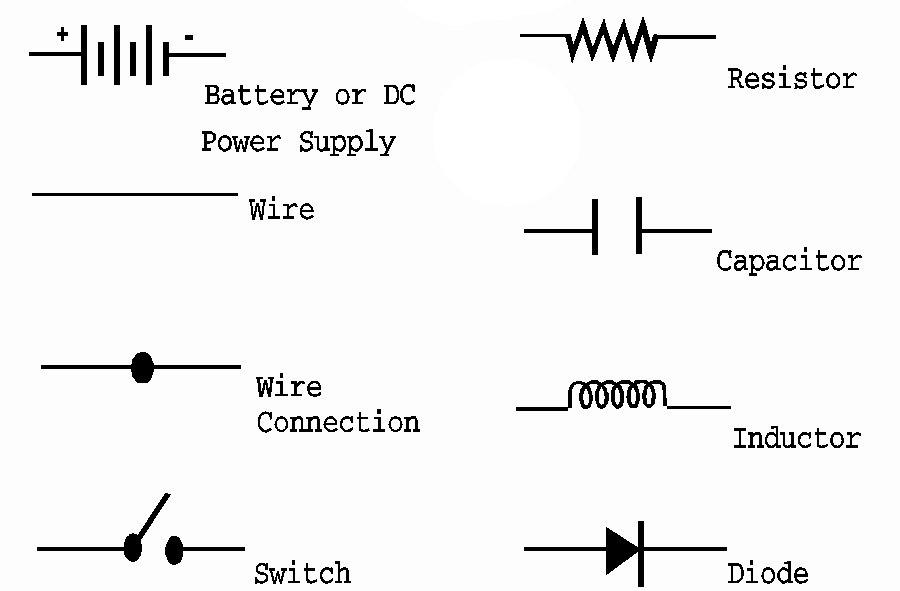
Obiectele, liniile, scrisul va fi de culoare alba

(#FFFFFF) pentru a se distinge de background.

Se va folosi drag & drop -> se vor selecta cu mouse-ul elementele din chenarul mic si se vor plasa pe masa de lucru (se va implementa in cod utilizarea mouse ului in program de catre utilizator)

Obiecte

Obiectele vor fi reprezentate in forma lor grafica dintr-o schema electrica obisnuita, in sistem de coordonate xOy, cu font alb. Ca obiecte, avem:

* Generator



* Consumator (bec)
* Rezistor



* Capacitor
* Bobina



* Dioda
* Intrerupator (inchis/deschis)
* Fir

Fiecare obiect trebuie declarant in format de struct, cu variabile precum:

* char nume
* char unitate\_masura
* int valoare

Data fiind implementarea prin liste, va trebui sa punem si un camp de adresa care ne va duce la urmatoarea componenta

Problematici

Exista cateva probleme de realizare a proiectului

1. implementare drag & drop a obiectelor pe masa de lucru. In final, putem gandi masa de lucru ca fiind memoria, si noi putem reprezenta obiecte ca fiind noduri.
2. Realizarea unui cod care sa permita user ului sa mute obiectele de pe masa de lucru. Acest lucru implica urmatoarele:

* Daca avem 2 elemente intr o anumita ordine si user ul le interschimba, trebuie ca ele si in cod sa si schimbe pozitia una fata de cealalta.
* Firele trebuie sa se ajusteze in functie de pozitia obiectelor (mutate sau nu)

1. Codul trebuie sa verifice in background corectitudinea schemei electrice:
   * Existenta unei surse
   * Corectitudinea legilor fizice (Ohm, Kirkoff etc.)

Si sa alerteze user ul in legatura cu erorile. (Trebuie implementata o var sa alerteze user ulde anumitele erori) ?

Idei

Pop up pentru erori

* In stanga/dreapta sus, daca exista erori sa apara un semn al exclamarii si daca se da click apar erorile intr un chenar pe mijloc

Implementare undo/redo

* se creeaza una/doua stive pentru memorarea starii programului

Save/Save Us

* salvare ca fisiere binar a starii programului

Programul va fi impartit in 2 parti

1. interpretorul
2. fisierele cu structul ce reprezinta starea proiectului

26/11/2021

* Programul initializeaza window-ul principal (setam W, H, r = w div h). (rezolvata)
* Cand intram in functia while, se va apela o functie care va returna constant coordonatele cursorului mouse lui.(rezolvata)
* Pagina principala va fi impartita in 4 chenare (case uri). Cele 4 chenare se vor afla in partea stanga a window-ului, iar in dreapta, titul ELECTRONAT. (ish?)
* Se va apela unul din case uri atunci cand event ul inregistrat de mouse, se afla in unu din cele 4 chenare. Cand se da click, marginile chenarului isi schimba culoarea.(si functionalitatea de hover ) (nope).

27/11/2021- Deadline – 30.11.2021

* Finalizata pagina principala
* Cand se da click in unul de chenare, functia va returna un integer, asociat urmatoarei actiuni(deschidere tabla de lucru-info, etc)
* Functie back(care va fi functie de legatura intre tabla de lucru si pagina principala)
* Design tabla de lucru (unde desenam in panou, stanga, componentele si delimitam spatiul alocat pentru tabla)
* DRAG AND DROP

29/11/2021

* Functionalitatea de scriere/deschidere a fisierelor(open/save project) va fi implementata cu ajutorul unei functii care va scrie intr-un fisier binar, starea tablei, la momentul salvarii(stack-ul impreuna cu vectorul de componente)
* Pentru scrierea fisierului binar, se vor folosi struct-uri, aferente componentelor de pe tabla, ce sunt continute de stack/vector.
* Definirea stack ului/vectorului si a structurilor aferente , va fi facuta dupa cum urmeaza:
  + - Se vor declara structuri pentru fiecare obiect grafic ce este desenat pe tabla(linie, cerc, etc) care va fi format din coordonatele corespunzatoare .
    - Se vor declara structure pentru fiecare componenta in parte(Dioda, resistor etc) care vor fi compuse din struct-urile aferente obiectelor grafice(linii, cercuri, etc)
    - Se va declara un struct de tip COMPONENTA, care va contine ca si parametrii structure de tipul componentelor declarate anterior(Dioda, resistor etc), si va fi initalizata doar structura aferenta desenului grafic. (Daca o componenta desenata este o dioda, atunci structul COMPONENTE va avea initializat doar atributul afferent diodei, iar celelalte for fi NULL)
    - In final, se va construi un struct SITUATIA\_TABLEI, care va contine stack ul si vectorul de COMPONENTE(vector pt a face mai usor scrierea/citirea componentelor) care va reprezenta situatia tablei la momentul salvarii/citirii.
    - In fine, acest struct va fi scris in fisierul binar, si salvat.