

Programarea Calculatoarelor

Seria CC

Tema 2

Publicarea enunțului: 17.12.2016

Date ultimei modificări a enunțului: 04.01.2017 (actualizare cerinte 3,4,5)

Termen de predare: 13.01.2017, 23:55

Nu se acceptă temele trimise după termenul de predare !



Responsabili temă: *Gabriel Vîjială, Remus Ioniță*

Propunător: *Cătălin Icleanu*

Profesor titular: *Carmen Odubășteanu*

Facultatea de Automatică și Calculatoare
Universitatea Politehnica din București
Anul universitar 2016 – 2017

OBIECTIVE

În urma realizării acestei teme, studentul va fi capabil:

- să folosească funcții din biblioteca grafică ncurses
- să implementeze un joc minimalist folosind limbajul C
- să aloce/dezaloce dinamic memoria necesară stocării unor date
- să descopere îmbunătățirile care pot fi aduse unui joc pentru a-l face mai interesant
- să modularizeze codul prin funcții. Implementarea unei aplicații mai complexe, în cazul de

fata un joc cu o interfata grafica, necesita o atentie sporita la modularizarea codului.

DESCRIEREA PROBLEMEI

Tema proiectului din anul universitar 2016 - 2017 presupune realizarea binecunoscutului joc Hangman pe Linux, folosind biblioteca grafică ncurses.

Scopul jocului Hangman este de a ghici o expresie, pornind de la câteva caractere, date drept indiciu. De fiecare dată când se introduce un caracter care se regăsește în expresie se avansează spre castigarea jocului. Dacă caracterul introdus nu este conținut în expresie se desenează un nou element al personajului din spânzurație.

Pentru a înțelege mai bine jocul, vă recomandăm:

- <http://www.hangman.no>
- <https://hangmanwordgame.com/>
- [https://en.wikipedia.org/wiki/Hangman_\(game\)](https://en.wikipedia.org/wiki/Hangman_(game))

CERINȚE

Se dorește implementarea unei aplicații care să permită unui utilizator uman să joace Hangman, iar pentru acest lucru va trebui să realizați următoarele task-uri.

CERINȚA 1 - 15 puncte

Punctajul aferent acestei cerințe se va acorda pentru realizarea unui meniu principal din care utilizatorul aplicației să poată selecta și alege cel puțin următoarele opțiuni:

- **New Game**
- **Resume Game**
- **Quit**

Navigarea în meniu se va face cu ajutorul tastelor directionale (săgețile sus / jos), iar opțiunea selectată în mod curent va fi marcată cu un caracter în dreptul ei sau cu orice alt efect considerat potrivit (de exemplu un highlight pe scris). Selectarea uneia dintre opțiuni se va face cu **Enter**.

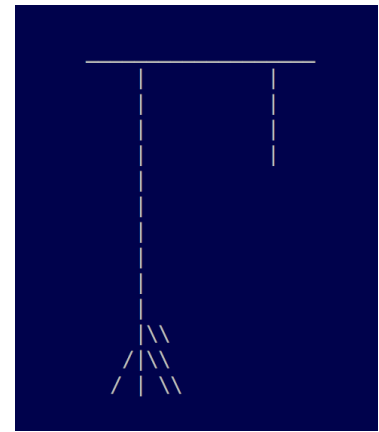
Rezultatul primei opțiuni (**New Game**) va fi începerea directă a unei noi runde, ce presupune afișarea ecranului de joc, care va conține, cel puțin, desenul cadrului spânzurației și un panou de control, ambele descrise la cerința 2. După începerea jocului se poate reveni la meniu apăsând tasta **Q**, iar într-o astfel de situație opțiunea **Resume** devine accesibilă. Astfel, selectarea acestei opțiuni va determina revenirea la ultimul joc început. Pentru a închide aplicația, se va selecta opțiunea **Quit**.

Pe măsura ce adăugați noi facilități aplicației voastre, puteți adăuga și alte opțiuni în meniu, specificându-le în fișierul README.

CERINȚA 2 - 15 puncte

În cadrul acestei cerințe veți defini layout-ul ecranului de joc. Pentru acordarea punctajului aferent acestei cerințe, aplicația voastră trebuie să conțină:

- Desenul cadrului spanzurătorii (un exemplu este dat în imaginea din dreapta)
- Un panou de control care conține cel puțin:
 - Data și timpul curent (se actualizează la apăsarea unei taste)
 - O legendă a comenzilor valide (instrucțiuni de interacționare utilizator-aplicație)



CERINȚA 3 - 20 puncte

Aplicația va porni la prin rularea executabilului numit `hangman`.

- Se verifică dacă programul a fost rulat cu argumente.
 - În cazul în care nu s-au dat argumente programului, se va afișa la ieșire șirul `[Eroare]: Nu s-au dat argumente de comandă`. După afișare se va ieși din program cu codul 1. (Funcția `main` va returna valoarea 1, nu valoarea 0).
- Fiecare argument al programului reprezintă numele unui fișier, care conține câte o expresie pe fiecare linie. Se vor citi toate aceste fișiere linie cu linie, salvându-se în memoria internă a programului expresiile procesate în modul următor:
 - Se vor șterge toate caracterele neprintabile și se vor converti toate literele în litere mici (folosind `isalpha()`, `isprint()`, `tolower()` din `ctype.h`).
 - Dacă vreo linie nu conține caractere printabile, expresia de pe această linie va fi ignorată (nu va fi memorată)
 - **La final, expresiile memorate vor conține doar caractere printabile (unde toate literele vor fi litere mici).**
 - Dacă vreun fișier nu poate fi deschis, se va afișa la ieșire șirul: `[Eroare]: Fișierul %s nu poate fi deschis`. După afișare se va ieși din program cu codul 1. (Funcția `main` va returna valoarea 1, nu valoarea 0).
- În panoul de control se va adăuga și o linie care conține numărul de expresii valide și numărul total de caractere ale acestor expresii valide (numărul de caractere printabile).
- Aplicația va trebui să primească orice număr de fișiere, fiecare având orice număr de linii (expresii). Numărul total de caractere pentru un set de fișiere este mai mic decât `2.000.000.000`.

OBSERVAȚII:








- Va trebui să funcționeze corect comanda `./hangman README hangman.c Makefile`.
- Trebuie să se folosească vectori care își dublează capacitatea la atingerea limitei stabilite la pasul curent (folosind funcția `realloc` din `stdlib.h`).

CERINȚA 4 - 30 puncte

Pentru a primi punctajul pentru acest task, trebuie implementată logica jocului propriu-zis:

- Se alege o expresie aleatoare din expresiile memorate (citite, linie cu linie, din fișierele de input și parsate conform cerinței 3). De exemplu: `"ana are mere"`.
- Se generează un șir, denumit în continuare șir „ascuns”, după următoarea logică:
 - Se păstrează, pentru fiecare cuvânt, primul și ultimul caracter.
 - Pentru cuvinte de 2 caractere se păstrează doar primul caracter.
 - Pentru cuvinte de un singur caracter se afișează `_`.
 - Fiecare caracter afișat o dată se afișează în toate spațiile corespunzătoare din expresie. De exemplu:
 - Pentru șirul `"ana are mere"`, se va genera șirul „ascuns” `"a_a a_e me_e"`

- Pentru sirul "x ya z axc", se va genera sirul „ascuns” "_ ya _ a_c".
- Se printeaza sirul „ascuns” deasupra desenului cu spanzuratoarea.
- Se implementeaza logica jocului: **La apasarea oricarei taste (corespunzatoare unei litere mici sau unui numar sau simbol) se ia urmatoarea decizie:**
 - Atunci cand caracterul este introdus pentru prima oara in jocul curent:
 - Daca se gaseste in expresie, se va afisa si in sirul „ascuns”.
 - Daca nu se gaseste in expresie, se actualizeaza desenul cu victima spanzuratorii (se trece la starea urmatoarea starii curente). De asemenea, caracterul greșit se afișează într-o listă sub desenul spânzurătorii.
 - Atunci cand caracterul a mai fost tastat anterior nu se va intampla nimic.

| Numar Greseli | 0 Start | 1 | 2 | 3 | 4 | 5 | 6 Game Over |
|------------------------|---|---|---|---|---|---|---|
| Stare Spanzuratoare |  |  |  |  |  |  |  |

Dupa terminarea jocului (WIN – s-au completat corect toate caracterele din expresie; **Game Over** – se ajunge in starea finala 6) se va afisa un mesaj corespunzator. Utilizatorul va avea de ales intre a porni un nou joc prin tasta **N** sau a se intoarce la meniu prin tasta **Q**. După ce un joc este finalizat, dacă utilizatorul se întoarce la meniu prin tasta **Q**, opțiunea **Resume Game** nu mai este disponibilă.

CERINȚA 5 - 10 puncte

Daca utilizatorul nu apasa pe nicio tasta timp de 7 secunde, **aplicatia alege un nou caracter printabil** in mod aleator. Caracterul ales nu poate fi din cele introduse anterior sau din cele deja cunoscute ("a" din sirul „ascuns” "a_a a_e me_e"). Jocul continuă ca și când utilizatorul ar fi apăsat acea literă.

OBSERVATIE

Tinand cont ca nu va este impus un format strict al aplicatiei, din punct de vedere grafic, veti primi puncte **bonus** pentru aspectul artistic al aplicatiei si pentru diverse imbunatiri sau idei originale pe care le aduceti.

Facilitatile adaugate vor fi punctate proportional cu originalitatea si calitatea implementarii, dar punctajul total acordat pentru aceasta tema nu va depasi **130 puncte**.

Puteti lua in calcul si urmatoarele exemple pentru puncte bonus: posibilitatea de a relua jocul dupa inchiderea aplicatiei; afisarea celor mai bune scoruri – high scores; folosirea de culori, animatii etc.

PRECIZĂRI

- Punctaj: 130 puncte = 90 puncte (cerintele 1-5) + 10 puncte (README + „Coding Style”) + 30 puncte (Bonus). Pentru obtinerea punctajului legat de scrierea codului, cititi [sectiunea dedicata](#) de pe site-ul de laboratoare
- Tema va fi corectată parțial de un sistem automat de verificare. Pentru a primi punctajul aferent cerințelor, aveți grijă să respectați toate cerințele subiectului, precum șirurile meniului (**New Game, Resume Game, Quit**) să apară exact așa cum s-a precizat (nu

- newGame, Joc Nou, Începe Distracția* etc.), ordinea selecțiilor din meniu (ex. **New Game** să fie prima variantă, care selectează apăsând tasta **Enter**), tastele de comandă (ex. se reintră în meniu cu **Q**), șirurile de eroare trebuie să fie corecte.
- Pentru obtinerea bibliotecii ncurses (pe o distributie de Linux bazata pe Debian, instalati cu: `apt-get install libncurses5-dev`)
 - Detalii despre biblioteca ncurses gasesc in urmatoarele materiale:
 - ✓ [documentatia oficiala a bibliotecii](#)
 - ✓ [o scurta introducere in folosirea bibliotecii ncurses](#)
 - ✓ [exemple](#)
 - Rezolvarea va porni de la arhiva temei – aflata pe site-ul de cursuri;
 - Tema va fi rezolvata obligatoriu in limbajul C. Nu folositi elemente ale limbajului C++
 - Vetii incarca pe site-ul de cursuri o arhiva zip <grupa>_<nume>_<prenume>.zip (de exemplu, 313CC_Popescu_Maria.zip) care va contine fisierele sursa, Makefile si README. Fisierele trebuie sa se regaseasca direct in radacina arhivei
 - Precizati in README cerintele rezolvate si modul in care se interactioneaza cu aplicatia voastra. Explicati, pe scurt, cum ati realizat implementarea cerintelor. Specificati, de asemenea, tot ce ati implementat ca bonus
 - Temele care nu compileaza la comanda `make` un executabil numit `hangman` nu vor fi punctate
 - Arhivele care nu sunt trimise sub formatul de mai sus nu vor fi punctate
 - Temele sunt individuale. Copierea va fi sanctionata – anulara punctajului temei pentru toti studentii implicati. Sursele copiate de pe Internet vor fi, de asemenea, anulate.

RESTRICȚII

- Numarul total de caractere este mai mic decat 2.000.000.000
- Nu se vor folosi variabile globale
- Memoria se va aloca dinamic
- Eliberați memoria alocată dinamic. Folosiți `valgrind --tool=memcheck --leak-check=full ./hangman` pentru a verifica dacă memoria este eliberată corect.