Fazan

[14.01.2020]

Îndrumător: Student:

dr.ing.Daniel Morariu Mănig Alexandra-Iulia(221/2)

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versiune | Descriere | Autor |
| 20.11.2019 | 1 | Implementarea celor 2 clase și crearea unor anumite butoane (Exit,Start), și imaginea pe interfață. | Mănig Alexandra-Iulia |
| 07.12.2019 | 2 | Rețeaua | Mănig Alexandra-Iulia  Canciu Bogdan |
| 21.12.2019 | 2.1 | O aplicație care reprezintă Clientul. Am adăugat un edit box în care clientul scrie IP-ul pentru a se conecta la server printr-un buton Conect. Pe lânga acestea am mai adăugat un edit box în care clientul introduce cuvântul pe care vrea să îl trimită și butonul de trimitere . | Mănig Alexandra-Iulia |
| 21.12.2019 | 2.2 | O aplicație care reprezintă Server-ul. Am adăugat un edit box în care Server-ul va introduce cuvântul și un buton de trimitere (Trimite cuvânt). | Mănig Alexandra-Iulia |
| 02.01.2020 | 3 | Am făcut o funcție în care Clientul trimite prima dată prima literă la începerea jocului iar în momentul în care apasă pe butonul “Trimite litera” , ii se va trimite serverului în mod aleatoriu o literă din alfabet cu care să înceapă cuvântul lui. Totodată în cazul în care Server-ul va încerca să trimită un cuvânt care începe cu o altă literă , decât cea primită nu va fii posibil deoarece am adăugat o funcție care să verifice acest lucru. | Mănig Alexandra-Iulia |
| 04.01.2020 | 3.1 | Am realizat o funcție care să verifice ca ultimele două litere ale cuvântului primit de la adversar să fie primele două litere din cuvântul pe care se dorește a fii trimis , în caz contrar , nu se poate trimite cuvântul . | Mănig Alexandra-Iulia |
| 10.01.2020 | 4 | O funcție ,care de fiecare dată când este apăsat butonul “NU STIU“ să se trimită acest mesaj adversarului și să se adauge câte o litera la scorul aferent (adică acel FAZAN).Totodată ,scorul care o să meargă pană la 2 , adică se va realiza jocul de două ori , imaginea castigatorului și învinsului , și ultimele modificări. | Mănig Alexandra-Iulia |

Cuprins

[1.Specificarea cerințelor software 4](#_Toc29882730)

[1.1 Introducere 4](#_Toc29882731)

[1.1.1 Obiective 4](#_Toc29882732)

[1.1.2 Definiții , Acronime și Abrevieri 5](#_Toc29882733)

[1.1.3 Tehnologii utilizate 5](#_Toc29882734)

[1.2 Cerințe specifice 5](#_Toc29882735)

[1.3 Rețeaua 6](#_Toc29882736)

[1.4 Descriere 6](#_Toc29882737)

[1.5 Fluxul de evenimente 6](#_Toc29882738)

[1.6 Metodele de verificare 9](#_Toc29882739)

[1.7 Descriere 9](#_Toc29882740)

[1.8 Fluxul de evenimente 9](#_Toc29882741)

[2. Implementare 11](#_Toc29882742)

[2.1 Diagrama de clase 11](#_Toc29882743)

[2.2 Descriere detaliată 12](#_Toc29882744)

[3.Bibliografie 13](#_Toc29882745)

# 1.Specificarea cerințelor software

## 1.1 Introducere

Jocul Fazan este un joc de cuvinte, de abilitate mintală pentru doi sau mai mulți jucători. La începutul jocului ,primul jucător spune cu glas tare „A!” și parcurge, în gând, alfabetul, reluându-l de la început dacă e nevoie. Al doilea jucător îl întrerupe după un timp iar primul jucător anunță litera la care a ajuns în acel moment și spune un cuvânt care începe cu litera respectivă. Jocul continuă prin numirea, pe rând, de către jucători, a unui cuvânt care să înceapă cu ultimele două litere ale cuvântului rostit de jucătorul precedent. Când un jucător nu poate găsi un asemenea cuvânt, este închis sau încuiat și, ori el însuși, ori jucătorul următor, începe iar jocul spunând un cuvânt dorit de el.

De fiecare dată când un jucător este închis, acesta primește, pe rând, câte o literă din cuvântul Fazan. Jucătorul închis o singură dată „este F”, cel închis de două ori „este Fa” și așa mai departe . până la Fazan, moment în care jucătorul este considerat învins. Jocul continuă până sunt învinși toți jucătorii în afară de unul, care este declarat învingător.

### 1.1.1 Obiective

**Realizate:**

* Să învăț să lucrez în C++ Builder
* Crearea claselor și utilizarea lor
* Crearea constructorilor
* Utilizarea moștenirii
* Să trimit cuvinte în rețea
* Utilizarea unei funcții care în momentul apăsării butonului “Trimite literă” , va trimite o literă random din alfabet adversarului
* Utilizarea unei funcții care să verifice ca ultimele 2 litere ale cuvântului primit de la Client/Server este identic cu primele două litere ale cuvântului dat de Server/Client
* Calcularea scorului fiecărui jucător în functie de regulă
* Realizarea unui program funcțional și ușor de înțeles

**Nerealizate:**

* Utilizarea unei biblioteci care să verifice dacă cuvintele trimise de cei doi jucători sunt corecte sau dacă există în dicționar
* Utilizarea polimorfismului
* Utilizarea unui zar care să se rotească pe toate părțile la începutul jocului atunci când se trimite litera cu care trebuie să înceapă jocul , iar litera să fie scrisă pe zar, atunci când este apasat butonul (Trimite litera).
* Utilizarea unei funcții care , în momentul în care se schimbă scorul să o ia de la capăt astfel încât clientul să trimită din nou o literă random din alfabet cu care mai apoi serverul să înceapă cuvântul.

### 1.1.2 Definiții , Acronime și Abrevieri

Pentru a da nume variabilelor folosite în aplicație am încercat să folosesc o denumire cât mai sugestivă a ceea ce fac ele sau a ceea ce sunt . Spre exemplu ,câteva dintre componentele de pe forma 2 în care are loc jocul propriu-zis, sunt următoarele:

- btnexit2(care sugerează al doilea buton de iesire din joc , primul fiind cel de pe prima formă)

-nameBox și enemyNameBox( numele celor doi jucători )

-sendCuvantButton (semnifică butonul la apăsarea căruia se trimite cuvântul scris, adversarului și viceversa)

-cuvList și cuvEnemyList (simbolizează listele în care se adaugă cuvintele scrise atat de jucătorul 1 cât si de jucatorul 2)

-btnNuStiu(acest buton pe lângă că trimite în locul cuvântului, textul “NU STIU“ adversarului, adaugă o literă în plus la propriul scor și permite scrierea de către adversar a oricărui cuvânt)

### 1.1.3 Tehnologii utilizate

* Pentru realizarea proiectului s-a folosit mediul de dezvoltare ***Borland C++ Builder 6***
* Pentru realizarea documentației s-a folosit ***Microsoft Word 2016***

## 1.2 Cerințe specifice

Proiectul are la bază următoarele funcționalități:

* btnexitClick(TObject \*Sender) – la apăsarea butonului **Exit** din aplicația Client/Server se iasă din aplicație și se eliberează memoria
* btnstartClick – la apăsarea butonului **Start** se trece Form1 pe hide și trece pe visibil Form2
* String textprimit-această variabilă este folosită în primă fază în momentul în care adversarul își trimite numele, jucătorului îi apar restul de componente pe formă ,până ca acesta să trimită, toate componentele în afară de cele implicate sunt pe visible false
* btnNuStiu- când unul dintre jucatori(client sau server) apasă pe butonul” NU STIU „ se trimite pe socket, către client sau către server, depinde cine a apăsat butonul, textul "Litere:" ,după care, în funcție de câte litere are jucatorul care a apăsat pe buton la un moment dat; să presupunem că nu are nici o literă, apasă prima oară pe buton trimite pe socket urmatoarea chestie: "Litere:f" în momentul în care inamicul citește acest lucru, schimbă în box-ul de litere al inamicului valoarea care este după "Litere" și îi adaugă inamicului litera f dacă e deja la litera f și apasă pe buton, trece la fa și așa mai departe până la fazan în momentul în care a ajuns la fazan, se șterg toate literele ambilor jucători iar inamicul celui care a facut "fazan" primește 1 punct
* brnNameSelectClick – la apăsarea acestui buton se trimite pe socket adversarului numele pe care l-a ales jucătorul , după care va trece pe hide
* String textDeTrimisAuxiliar – această variabilă este folosită pentru a verifica dacă cuvântul scris de server începe cu prima literă trimisă de către client
* String textAuxiliar- reține ultimele două litere din cuvântul trimis de inamic

**Funcţionalitate 1**

### 1.3 Rețeaua

Aplicația realizată trimite mesaje de la un utilizator la altul folosind **Rețeaua**.

### 1.4 Descriere

În C++ Builder există 2 tipuri de componente:

* Componente vizuale – sunt acele componente care rămân pe formă și în timpul rulării aplicației. Toate componentele folosite până acum au fost componente vizuale;
* Componente nevizuale – sunt acele componente care ajută la configurarea aplicației în timpul dezvoltării acesteia și dispar de pe formă în momentul rulării acesteia;

Pentru a putea trimite cuvinte între 2 utilizatori am realizat 2 aplicații: una Client și una Server.

* În aplicația Server am adăugat o componentă nevizuală numită **„ServerSocket”**- este o componentă care ajută la configurarea programului pentru a avea acces la un Socket de pe placa de rețea. Pentru configurarea acestei componente (Server Socket) în ObjectInspector” trebuie specificate următoarele caracteristici: **Port** (2000) și **Active** (true). După aceea componenta ServerSocket va monitoriza placa de rețea pe portul specificat și va aștepta să primească sau să trimită mesaje (șiruri de caractere). Pentru aplicația Server vom activa partea de recepție mesaje (prin evenimentul **Server SocketClientRead**) și partea de conexiune (prin evenimentul **Server SocketAccept** )
* În aplicația Client, pe lânga componentele vizuale am adăugat o componentă nevizuală numită **„ClientSocket”** - permite conectarea la un server de pe rețea, printr-un port, și permite transmiterea și recepționarea de mesaje de tip text prin aceasta. Pentru a configura această componentă se specifică **Portul** (2000), apoi la apăsarea butonului **Connect** se va prelua IP-ul dintr-un edit box si se va activa conexiunea. Pentru aplicația Client vom activa partea de recepție mesaje prin evenimentul ,,**ClientSocketRead”.**

### 1.5 Fluxul de evenimente

#### 1.5.1 Fluxul de bază

Pentru a începe jocul se rulează cele 2 aplicații. În aplicația Client se va introduce în **eAdresa** IP-ul și apoi se va apăsa butonul **Connect**. În momentul apăsării se vor executa instrucțiunile:

void \_\_fastcall TForm1::btnConnectClick(TObject \*Sender)

{

String text;

if(InputQuery("Connect", "Introduceti IP:", text)){ // va aparea fereastra pentru introducerea

IP-ului

if(text.Length()!=0){

ClientSocket->Address=text;

ClientSocket->Active=true;

ShowMessage(ClientSocket->Address);

btnstart->Show();

Dacă IP-ul este corect înseamnă că această conexiune a avut loc, iar apoi va apărea câte o căsuță pentru fiecare din cei doi jucători în care va trebuii să își scrie numele:

void \_\_fastcall TForm1::ClientSocketRead(TObject \*Sender,

TCustomWinSocket \*Socket)

{

String textprimit;

textprimit=Socket->ReceiveText();

if(receiveNume == false){

Form2->enemyNameBox->Text=textprimit; // după ce adversarul își trimite

Form2->enemyNameLabel->Visible=true; // numele ,vor fii trecute pe forma

Form2->enemyNameBox->Visible=true; //și restul de componente, adică

receiveNume=true; // vor trece pe Visible=true

if(Form2->enemyNameBox->Visible == true){

Form2->fazanLabel->Visible=true;

Form2->cuvantBox->Visible=true;

Form2->cuvList->Visible=true;

Form2->labelLitereInamic->Visible=true;

//Form2->btnNuStiu->Visible = false;

Form2->enemyCuvantBox->Visible=true;

Form2->cuvEnemyList->Visible=true;

Form2->btnLiteraInceput->Visible=true;

Form2->lblLitere->Visible=true;

Form2->lblScor->Visible=true;

Form2->litereBox->Visible=true;

Form2->scorBox->Visible=true;

Form2->lblLitereEnemy->Visible=true;

Form2->litereEnemyBox->Visible=true;

Form2->lblScorEnemy->Visible=true;

Form2->scorEnemyBox->Visible=true;

În continuare după ce fiecare dintre cei doi jucători și-au ales numele și restul componentelor au devenit vizibile , Clientul va trimite întotdeuna prima literă la apăsarea butonului “Trimite litera “ , după care acest buton va trece pe (Visible=false) . În mod aleatoriu la apăsarea acestui buton ii se va trimite Server-ului o literă din alfabet cu care va trebuii să înceapă cuvântul său.

void \_\_fastcall TForm2::btnLiteraInceputClick(TObject \*Sender)

{

btnLiteraInceput->Visible=false; // butonul trece pe (Visible

srand(time(NULL)); //= false), iar în mod alea-

char deTrimis = 'a' + rand()%26; //toriu se va trimite pe

Form1->ClientSocket->Socket->SendText(deTrimis); //socket o literă

}

Atunci când Clientul-ul apasă pe butonul **Trimite Litera** se va trimite un mesaj Server-ului :

Form1->ClientSocket->Socket->SendText(deTrimis);.

#### 1.5.2 Pre-condiții

1.Se deschid ambele aplicații

2.În aplicația Server se va apăsa butonul start , după care se va da connect în aplicația Client și se va introduce IP-ul pentru ca acesta să se conecteze la Server.

3.Pentru a începe jocul cei doi jucători va trebuii să își scrie numele în căsuțele care apar iar dupa ce vor da alege nume vor putea începe jocul.

4.Clientul va apăsa (Trimite litera) iar prin box-ul de cuvinte ii se va trimite Server-ului litera cu care trebuie să înceapă cuvântul său.

5.Server-ul trimite primul cuvânt care începe cu litera care i-a fost trimisă.

6.Clientul trimite al doilea cuvânt care va începe cu ultimele două litere ale cuvântului trimis de către server, și așa mai departe.

7. Când unul dintre jucători va primi un cuvânt al cărui sufix nu mai poate forma alt cuvânt, sau pur și simplu nu își amintește un cuvânt care să înceapă cu acel sufix acesta se va declara învins și va apăsa butonul NU STIU când scorul (FAZAN) se va actualiza.

8. Jocul continuă iar adversarului celui căreuia a trimis mesajul NU STIU continuă prin a trimite un cuvânt care începe cu orice literă dorește urmând să se utilizeze aceiași pași până când unul din jucători ajunge să fie FAZAN, moment în care jocul s-a terminat prima rundă.

9. La urmatoarea rundă se începe de către adversarul celui care a trimis textul NU STIU , la fel cu orice literă își dorește .Jocul merge până când unul dintre jucători va ajunge la scorul 2, ceea ce înseamnă maxim trei runde, acest lucru dacă după două runde jucătorii sunt la egalitate .După ce unul dintre jucători a ajuns la acest scor va arăta pe interfața câștigătorului o imagine care îi va spune că a câștigat și pe cea a pierzătorului o imagine care îi va spune că a pierdut.

**Funcționalitate 2**

### 1.6 Metodele de verificare

În aplicația Fazan sunt utilizate 2 metode ale clasei principale care verifică litera în Server și care verifică ultimele doua litere respectiv primele doua litere atât în Client , cât și în Server

### 1.7 Descriere

Pentru a ușura munca jucătorilor am realizat 2 metode de verificare, care nu le vor permite să greșască sau să trișeze.

### 1.8 Fluxul de evenimente

#### 1.8.1 Fluxul de bază

Tot timpul jocul este început de Server, dar Clientul îi trimite litera cu care să înceapă primul cuvânt . Dacă cuvma Server-ul nu este atent ce cuvânt trimite prima dată programul nu îi va permite să trimită un cuvânt cu o altă literă.

Atunci când se intră pe metoda de verificare literă se execută:

if(adaugareCuv == true && text!=""){

if(ct==0){

String textDeTrimisAuxiliar = text.SubString(0,1); //această funcție va

if(textDeTrimisAuxiliar == enemyCuvantBox->Text.SubString //verifica doar atuci

(enemyCuvantBox->Text.Length(),1)){ //când se începe jo-

jucator1->adaugaCuvant(text,jucator1->getContor()); //cul

cuvList->Items->Add(text);

adaugareCuv = false;

sendCuvantButton->Visible=false;

btnNuStiu->Visible = false;

Form1->ServerSocket->Socket->Connections[0]->SendText(text);

ct=1;

}

}

Cu alte cuvinte, Clientul nu va puea trimite un cuvânt până ce acesta nu este corect.

Atunci când se intră în metoda de verificare a cuvintelor va intra pe acest (else) și se va executa:

else{

String textAuxiliar = enemyCuvantBox->Text.SubString(enemyCuvantBox->Text.Length()-1,2);

String textDeTrimisAuxiliar = text.SubString(0,2); // astefel încât , va

if(textDeTrimisAuxiliar == textAuxiliar){ //lua din box-ul adversarului ,cu-

jucator1->adaugaCuvant(text,jucator1->getContor()); //vântul și începând cu

cuvList->Items->Add(text); // ultima poziție a cuvântului

sendCuvantButton->Visible=false; //va lua 2 litere de la dreapta

Form1->ServerSocket->Socket-> // la stânga și le va verifica

Connections[0]->SendText(text); // pentru a fii “==” cu cele 2 litere

} // ale cuvântului pe care se dorește

} //a fii trimise de către jucător

#### 1.8.2 Pre-condiții

1.Se deschid ambele aplicații

2.În aplicația Server se va apăsa butonul start , după care se va da connect în aplicația Client și se va introduce IP-ul pentru ca acesta să se conecteze la Server.

3.Pentru a începe jocul cei doi jucători va trebuii să își scrie numele în căsuțele care apar iar dupa ce vor da alege nume vor putea începe jocul.

4.Clientul va apăsa (Trimite litera) iar prin box-ul de cuvinte ii se va trimite Server-ului litera cu care trebuie să înceapă cuvântul său.

5.Server-ul trimite primul cuvânt care începe cu litera care i-a fost trimisă.

6.Clientul trimite al doilea cuvânt care va începe cu ultimele două litere ale cuvântului trimis de către server, și așa mai departe.

7. Când unul dintre jucători va primi un cuvânt al cărui sufix nu mai poate forma alt cuvânt, sau pur și simplu nu își amintește un cuvânt care să înceapă cu acel sufix acesta se va declara învins și va apăsa butonul NU STIU când scorul (FAZAN) se va actualiza.

8. Jocul continuă iar adversarului celui căreuia a trimis mesajul NU STIU continuă prin a trimite un cuvânt care începe cu orice literă dorește urmând să se utilizeze aceiași pași până când unul din jucători ajunge să fie FAZAN, moment în care jocul s-a terminat prima rundă.

9. La urmatoarea rundă se începe de către adversarul celui care a trimis textul NU STIU , la fel cu orice literă își dorește .Jocul merge până când unul dintre jucători va ajunge la scorul 2, ceea ce înseamnă maxim trei runde, acest lucru dacă după două runde jucătorii sunt la egalitate .După ce unul dintre jucători a ajuns la acest scor va arăta pe interfața câștigătorului o imagine care îi va spune că a câștigat și pe cea a pierzătorului o imagine care îi va spune că a pierdut.

#### 1.8.3 Post-condiții

#### 

Jucătorii nu vor avea probleme dacă nu vor introduce un cuvânt valid, deoarece lor nu li se va permite să trimită cuvinte care nu vor respecta regulile jocului.Totodată se așteaptă că jucătorii vor fii cinstiți și atunci când nu vor știi să trimită un cuvânt existent în vorbire , vor apasa butonul NU STIU.

Scorul se v-a actualiza automat, când unul din jucători va apăsa butonul NU STIU.

# 2. Implementare

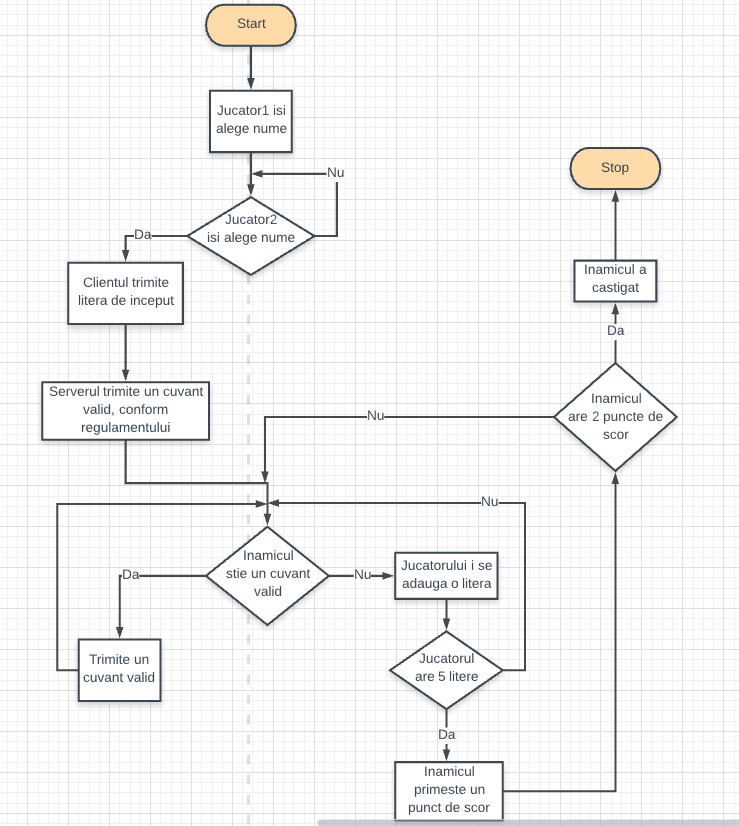
## 2.1 Diagrama de clase

|  |
| --- |
| FazanPlayer |
| # fazan : string;  #scor:int; |
| +FazanPlayer();  +FazanPlayer(String , int );  + ~FazanPlayer();  +void setFazan(String );  +String getFazan();  +void setScor(int );  +int getScor(); |

|  |
| --- |
| Player |
| #nume:string;  #istoricCuvinte:string;  #contorCuvinte:int; |
| +Player();  +Player (String , int ,String);  +~Player();  +void setNume(String);  +String getNume();  +String istoricJucator(int );  +void adaugaCuvant(String , int);  +int getContor();  +void setContor(int); |

+

## 2.2 Descriere detaliată



# 3.Bibliografie

* Programare Orientată pe Obiecte. Principii – Macarie Breazu – Editura Universității “Lucian Blaga” din Sibiu 2002
* https://ro.wikipedia.org/wiki/Fazan\_(joc)
* http://webspace.ulbsibiu.ro/daniel.morariu/html/StudentDoc/POO/Lab8\_UtilizareaMediuluiC++Builder.pdf
* http://webspace.ulbsibiu.ro/daniel.morariu/html/StudentDoc/POO/Lab9\_Forme\_C++Builder.pdf
* http://webspace.ulbsibiu.ro/daniel.morariu/html/StudentDoc/POO/Lab10\_Socket(Retea)C++Builder.pdf