



Universitatea Tehnică de Construcții București

Facultatea de Hidrotehnica

## Programarea Calculatoarelor si Limbaje de Programare II

**Profesor Coordonator:**

**Ing. Cosmin Fudulu Florentin**

**Ing. Gabriela Olteanu**

**Studenti:**

**Nedelcu Cosmin**

**Popa Alexandru-Cristian**



Universitatea Tehnică de Construcții București

Facultatea de Hidrotehnica

## PONG GAME

**Profesor Coordonator:**

**Ing. Cosmin Fudulu Florentin**

**Ing. Gabriela Olteanu**

**Studenți:**

**Nedelcu Cosmin**

**Popa Alexandru-Cristian**

## CUPRINS

Motivarea alegerii temei .....	4
Obiective Propuse.....	4
Descriere generală .....	5
Funcționalități principale .....	5
Controlul paletelor de către doi jucători (W/S și ↑/↓) .....	5
Mișcarea mingii și coliziuni .....	6
Sistem de scor pentru ambii jucători .....	6
Tehnologii folosite .....	7
C++.....	7
SFML .....	8
Visual Studio .....	9
GitHub Desktop .....	10
Radmin VPN .....	11
Structura Codului .....	12
Animation.h .....	13
CONCLUZIE.....	14
BIBLIOGRAFIE .....	15

## Motivarea alegerii temei

Am ales tema "Realizarea unui joc Pong multiplayer utilizând C++ și SFML" deoarece combină mai multe elemente fundamentale în formarea unui dezvoltator de software modern: programare orientată pe obiect, grafică 2D, rețelistică și munca colaborativă. Pong, deși este un joc simplu ca mecanică, oferă o bază excelentă pentru explorarea unor concepte tehnice complexe într-un mod accesibil și aplicativ.

Implementarea unui joc multiplayer implică înțelegerea noțiunilor de bază din networking, ceea ce este esențial în dezvoltarea aplicațiilor interactive moderne. În plus, utilizarea bibliotecii SFML pentru interfața grafică și gestionarea resurselor multimedia permite aprofundarea lucrului cu librării externe și dezvoltarea de abilități practice în C++.

Prin acest proiect am urmărit și dezvoltarea abilităților de organizare și colaborare în echipă, un aspect cheie în industria software. De asemenea, am integrat partea artistică a dezvoltării jocurilor prin crearea și gestionarea de spritesheet-uri, ceea ce adaugă o dimensiune creativă lucrării.

## Obiective Propuse

Prin realizarea acestui proiect ne-am propus aprofundarea unor concepte esențiale din domeniul informaticii, cu aplicabilitate directă în dezvoltarea jocurilor video. Un prim obiectiv a fost înțelegerea noțiunilor de bază din networking, necesare pentru a implementa funcționalitatea multiplayer a jocului. Aceasta presupune familiarizarea cu principiile comunicației în rețea, gestionarea conexiunilor și sincronizarea acțiunilor între doi jucători în timp real.

Un alt obiectiv a fost crearea unei interfețe grafice intuitive și funcționale folosind biblioteca SFML, care să permită afișarea și actualizarea în timp real a elementelor de joc (minge,

palete, scor). Acest demers ne-a ajutat să aprofundăm lucrul cu librării externe în C++, să folosim concepte precum ferestre, texturi, evenimente și randare grafică.

Totodată, ne-am dorit să învățăm să lucrăm eficient cu librării externe, în special cu SFML, parcurgând documentația oficială și integrând module diverse precum grafică, sunet și rețea, pentru a oferi o experiență de joc completă.

Am urmărit și dezvoltarea competențelor de organizare a muncii în echipă, repartizând sarcini, comunicând constant și utilizând instrumente de colaborare precum sistemul de control al versiunilor (Git), ceea ce ne-a pregătit pentru lucrul într-un cadru de tip profesional.

Nu în ultimul rând, un obiectiv important a fost crearea și utilizarea de spritesheet-uri personalizate, pentru a învăța cum se gestionează resursele grafice într-un joc și cum se pot realiza animații simple care îmbunătățesc aspectul vizual și atractivitatea proiectului.

## Descriere generală

Acest proiect este o implementare a jocului clasic "Pong Game" folosind biblioteca **SFML**. El implementează o variantă multiplayer, folosind biblioteca **SFML** pentru grafică, animații și rețea. Jocul suportă moduri de joc: singleplayer, host (server) și client.

Scopul acestui proiect este de a implementa jocul într-o manieră asemănătoare cu celelalte jocuri pong de pe piață, oferind o experiență bună prin competitivitatea sa, cât și posibilitatea de a juca multiplayer.

## Funcționalități principale

Controlul paletelor de către doi jucători (W/S și ↑/↓)

Controalele sunt alese în funcție de poziția jucătorului, dacă acesta alege paleta din stânga va folosi tastele W/S, respectiv ↑/↓ pentru jucătorul din dreapta, ele au direcții în sus și în jos și

împreună cu stripe-urile concepute creează un efect de lovire a mingii. Totuși faptul că animația are loc chiar și dacă mingea a trecut pe lângă paletă este tot un efect de animație de parca jucătorul a ratat lovitura.

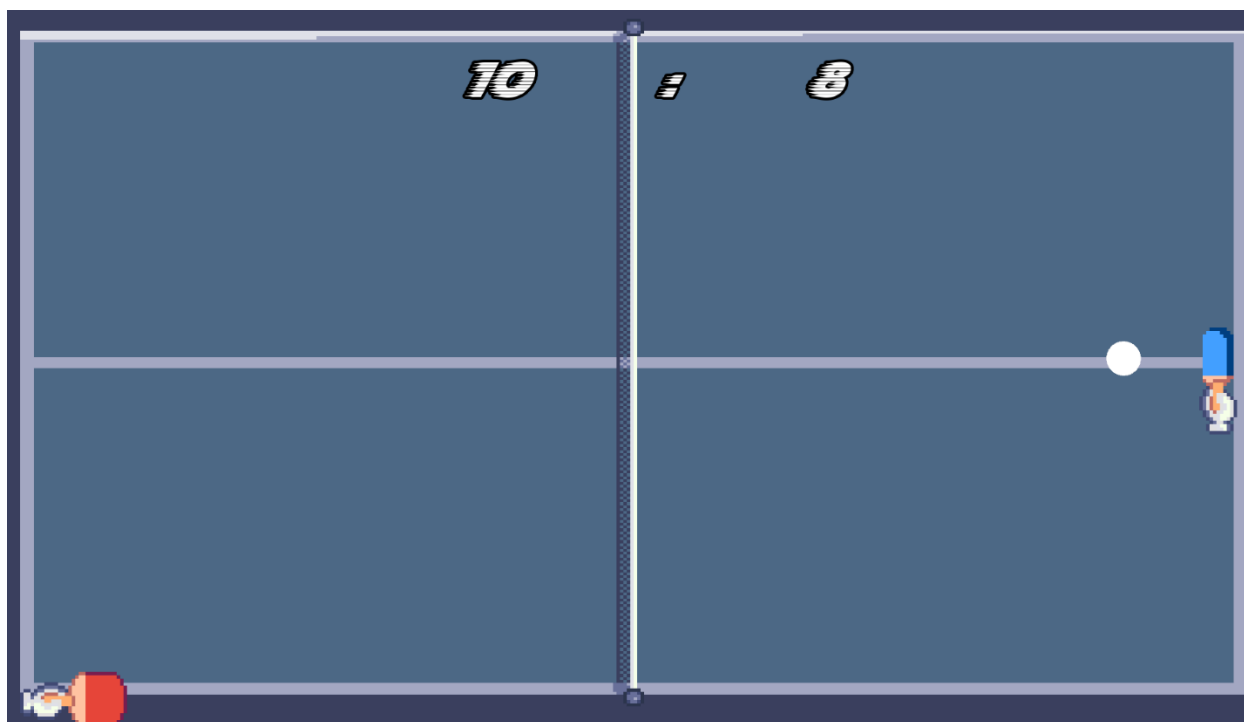
#### Mișcarea mingii și coliziuni

Mișcarea mingii își are originea din centrul mesei de ping pong și în funcție de cine obține punct, mingea se redirecționează spre el pentru a avea un mic avantaj în câștigarea de mai multe puncte.

Coliziunea are loc între hitbox-ul creat în jurul fiecărei paletă și mingea care este o formă implementată cu ajutorul librăriei, iar acestea când se intersectează în coordonate asemănătoare se respinge în direcții opuse cu aceeași viteză (se înmulțește cu -1).

#### Sistem de scor pentru ambii jucători

Sistemul de scor este unul simplu ce funcționează pe principiul logic al rezoluției unui ecran. Dacă mingea iese din limitele impuse de modul fullscreen automat al jocului atât în partea negativă a axei  $oX$ , cât și din partea pozitivă mai mult decât suportă ecranul (în cazul nostru avem fullscreen pe o rezoluție FHD  $\Rightarrow 1920 +$  dimensiunea mingii în pixeli), scorul se incrementează cu 1.



*Fig. 1 Interfața jocului Pong Game*

## Tehnologii folosite

Cele 5 tehnologii utilizate în construirea unui cod ușor de înțeles, bine structurat, sigur în privința riscului de a pierde datelor și posibilitatea opțiunii de multiplayer sunt următoarele:

C++ este un limbaj de programare de nivel înalt, orientat pe performanță, derivat din C, care suportă programare procedurală, orientată pe obiect și generică.

Este foarte utilizat în dezvoltarea de:

- Jocuri video
- Sisteme de operare
- Motoare grafice
- Aplicații ce necesită rulare rapidă și consum redus de resurse.

În proiectul nostru "Pong Game", C++ a fost limbajul principal în care am scris toată logica jocului. Iată câteva moduri concrete în care l-am folosit:

**Structurarea codului:** am creat fișiere separate pentru animații (*Animation.h*), coliziuni (*Colliders.h*) și rețea (*Networking.h*), demonstrând modularitate;

- **Controlul logicii jocului:** mișcarea mingii, paletelor, scorul, gestionarea coliziunilor și dificultății se fac cu variabile și funcții în C++;
- **Gestionarea tastelor:** am folosit *sf::Keyboard::isKeyPressed* pentru a detecta tastele apăstate (W/S și săgeți);
- **Orientare pe obiect:** clasa *Animation* este un exemplu de utilizare a OOP în proiect (funcții, membri, încapsulare);
- **Structuri și funcții utile:** am folosit *std::string*, *std::to\_string*, *std::cin*, *std::cout* și alte funcționalități din STL (Standard Template Library).

SFML (Simple and Fast Multimedia Library) este o bibliotecă multimedia scrisă în C++, folosită pentru dezvoltarea aplicațiilor care implică grafică 2D, sunet, input și rețea. Este o alternativă mai ușor de utilizat față de biblioteci precum SDL și este ideală pentru proiecte educaționale și jocuri simple.

#### Ce oferă SFML:

- Module pentru grafică, sunet, rețea, timp și evenimente de input;
- Interfață simplă, orientată pe obiect;
- Portabilitate (Windows, Linux, macOS).

În proiectul nostru "Pong Game", SFML este coloana vertebrală a jocului. Am folosit-o în următoarele moduri:

#### Fereastra jocului și grafică 2D:

Am creat o fereastră full-screen cu *sf::RenderWindow* în care totul se desenează



```
//Initializing Window
RenderWindow window(VideoMode({ 1920, 1080 }), "PONG GAME",
Style::Fullscreen);
window.setFramerateLimit(120);
Vector2u WindowSize = window.getSize();
```

Am desenat fundalul cu *sf::Sprite* și *sf::Texture*;

Paletele sunt desenate folosind *sf::RectangleShape*;

Mingea este un *sf::CircleShape*;

Scorul este afișat cu *sf::Text*.

### Evenimente de tastatură:

Folosesc *sf::Keyboard::isKeyPressed* pentru a mișca paletele:

```
if (sf::Keyboard::isKeyPressed(sf::Keyboard::W))
{ ... }
```

### Gestionarea timpului:

Am utilizat *sf::Clock* și *sf::Time* pentru controlul animațiilor și a frame-urilor.

SFML

Am ales SFML deoarece se integrează foarte bine cu C++, este ușor de învățat și folosit, are documentație clară și comunitate activă și este ideal pentru proiecte educaționale și prototipuri de jocuri 2D.

Visual Studio este un mediu de dezvoltare integrat (IDE) dezvoltat de Microsoft, foarte folosit pentru programare în C++, C#, Python și alte limbaje. Este una dintre cele mai puternice unelte pentru dezvoltarea de aplicații Windows, desktop, web și jocuri.

Caracteristicile principale sunt oferite de faptul că au un suport complet pentru C++ și integrare ușoară cu biblioteca externă SFML, debugger-ul este foarte avansat, putem gestiona eficient proiectele și fișierele, are suport pentru Git și extensii.

Am ales Visual Studio deoarece integrarea bibliotecii **SFML** în Visual Studio Code ne-a dat bătăi de cap, dar cu această experiență am descoperit și alte medii de dezvoltare.

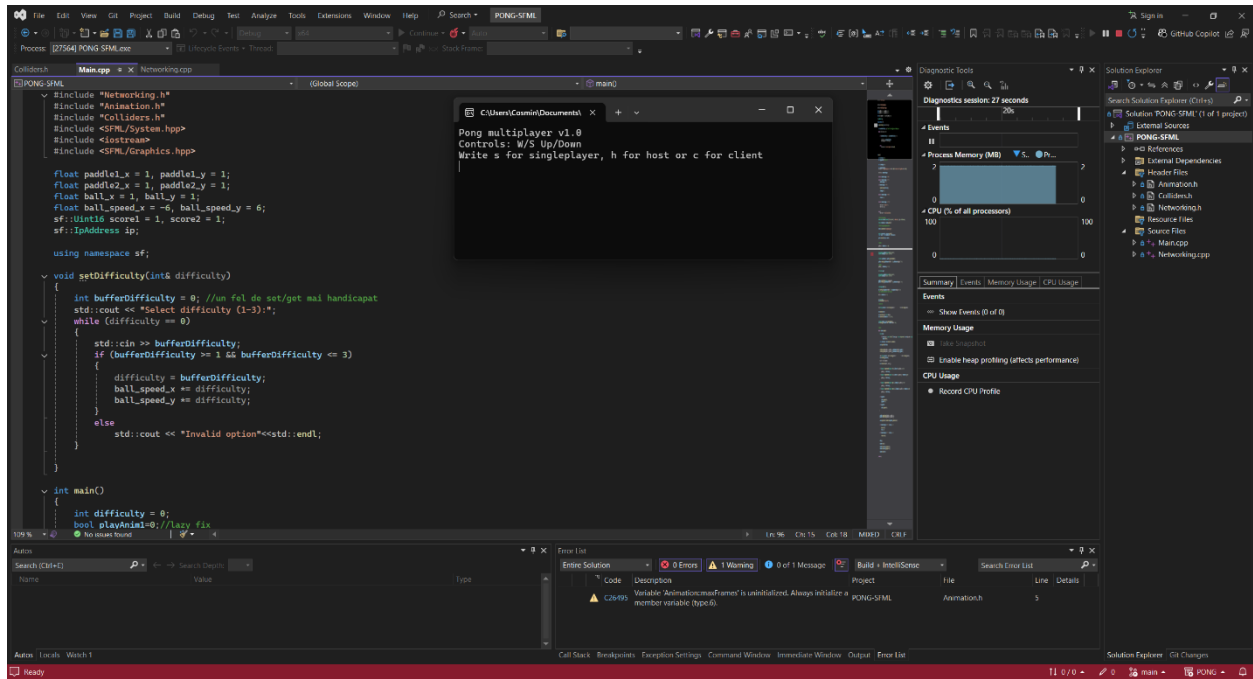


Fig. 2 Rularea jocului în Visual Studio

GitHub Desktop este o aplicație grafică care oferă o interfață prietenoasă pentru gestionarea proiectelor Git și interacțiunea cu platforma **GitHub**. Este ideală pentru programatori care vor să gestioneze versiunile codului fără să scrie comenzi Git în terminal.

### Caracteristici principale:

- Vizualizare modificări în fișiere;
- Commit-uri și mesaje de commit rapide;
- Sincronizare cu depozite GitHub (push, pull, fetch);
- Creare și gestionare de branch-uri;
- Istoric clar al dezvoltării proiectului.

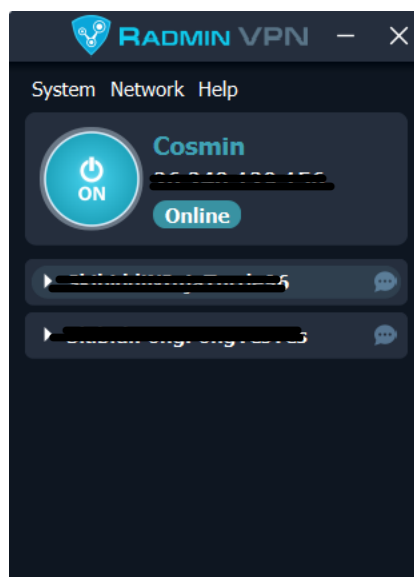
### Cum am folosit GitHub Desktop în proiectul nostru?

Pentru jocul “Pong Game”, GitHub Desktop m-a ajutat să, la crearea un repository Git unde am salvat tot codul sursă și resursele (imagini, fonturi), să versionez fiecare etapă importantă (adăugare animații, scor, multiplayer), să trimit actualizările direct în contul meu GitHub, pentru

backup și colaborare, să văd exact ce s-a modificat între versiuni și ceea ce m-a ajutat să repar ușor bug-uri și să revin rapid la versiuni anterioare dacă ceva nu mergea bine.

Am ales GitHub Desktop deoarece nu trebuie să folosim linia de comandă, ne ajută să ținem organizat codul și să colaborăm ușor cu între noi, este ideal pentru proiecte individuale sau de echipă, putem încărca proiectul nostru pe GitHub și îl putem arăta altora.

Radmin VPN este un software gratuit care creează o rețea virtuală privată (VPN) între calculatoare, ca și cum ar fi conectate într-o rețea locală (LAN), chiar dacă sunt la distanță. Este folosit des pentru jocuri multiplayer LAN prin internet, lucru la distanță pe aceleași fișiere și crearea unei rețele sigure între mai multe dispozitive.



*Fig. 3 Aplicați Radmin VPN*

Motivul pentru care am folosit Radmin VPN a fost că jocul „Pong Game” pe care l-am realizat este un joc în rețea **peer-to-peer** (gază-client), aveam nevoie de o soluție prin care două calculatoare diferite să se poată conecta între ele ca și cum ar fi în aceeași rețea locală. Aici intervine Radmin VPN: am creat un **grup VPN (rețea)** în aplicația Radmin, unul dintre

calculatoare (host-ul) a creat jocul și aștepta conexiunea, iar celălalt calculator (clientul) s-a conectat la rețea folosind IP-ul din Radmin. Astfel, cele două calculatoare s-au comportat ca într-o rețea LAN și jocul a funcționat ca un multiplayer real. Configurați este rapidă și gratuită, nu e nevoie să deschid porturi în router (fără NAT), funcționează bine și stabil pentru testare de jocuri multiplayer.

## Structura Codului

Partea Multiplayer (Networking.h/Networking.cpp)

La baza starea jocului este controlată în integritate de următoarele variabile globale:

```
float paddle1_x = 1, paddle1_y = 1;
float paddle2_x = 1, paddle2_y = 1;
float ball_x = 1, ball_y = 1;
float ball_speed_x = -6, ball_speed_y = 6;
sf::Uint16 score1 = 1, score2 = 1;
sf::IpAddress ip;
```

Networkingul se face printr-un model host-client în care logica jocului care contează este rulată pe masina jucatorului host, iar clientul doar primește variabilele pentru a face o interpretare grafică.

Pentru networking s-a folosit TCP deoarece garantează că pachetele vin ordonat și fără pierderi. De obicei se folosesc socketuri UDP pentru jocurile în timp real dar deoarece este nevoie să ai grijă de cazurile în care pachetele vin neordonat sau sunt pierdute nu a fost folosit pentru simplitate.

Dezavantajele TCP sunt minore dar simtite. Datorită felului în care modul TCP verifică pachetele este garantată o întârziere de minim 16ms care se poate întinde mult mai mult în cazurile în care pachetele au fost pierdute. Deoarece nu am implementat multithreading pentru network, jocul se blochează dacă delayul crește prea mult.

Pentru a face comunicarea între client și host sunt folosite următoarele funcții:

```
void host();
void client();
void host_tick();
void client_tick();
```

Funcția “host” porneste socketul TCP pe portul 50001 și așteapta să se conecteze clientul.

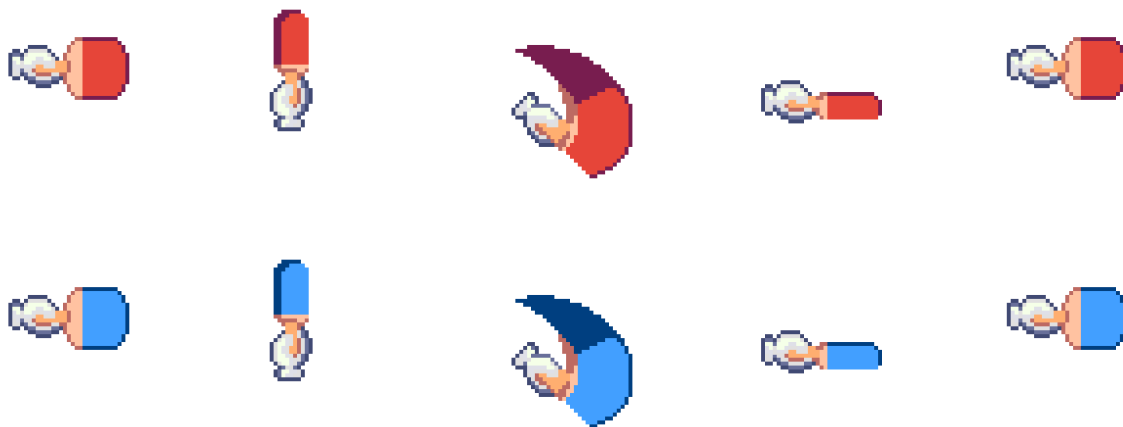
Funcția ”client” va cere o adresa ip la care se va conecta la socketul creat de host. Dacă nu există host la care să se conecteze se va afișa mesajul “Failed to connect.”

“host\_tick” și “client\_tick” sunt niște funcții care sunt chemate de 40 de ori pe secundă în care se transmit pachete care conțin variabilele dintre host și client.

Hostul trimite variabilele paletel, mingiei, vitezei mingiei, scorul și primește variabilele paletel 2, iar pentru client vice versa.

Animation.h:

Folosind clasa *sf::Sprite* și elementele ei am putut crea clasa Animation pentru a reda animații la coliziune. Sprite-urile din SFML au un dreptunghi care trece prin poza atribuită (un fel de UV mapping) pentru a alege doar o anumită parte a texturii pentru afișare. Schimbarea sprite-urilor a fost făcută prin mutarea acestui dreptunghi (care în acest caz are dimensiunile de 64x64 pixeli) către dreapta, pornind de la zero în incremente de 64 de pixeli.



## CONCLUZIE

Realizarea acestui joc **Pong Game** a reprezentat o experiență tehnică și creativă complexă, în care s-au împletit noțiuni fundamentale de programare cu aplicații moderne din domeniul rețelelor și al dezvoltării multimedia. Prin utilizarea limbajului **C++**, împreună cu biblioteca grafică **SFML**, mediul de dezvoltare **Visual Studio**, platforma de control al versiunilor **GitHub Desktop** și soluția de conectare VPN **Radmin**, am reușit să creez un proiect complet funcțional, interactiv și adaptabil atât pentru modul singleplayer, cât și pentru multiplayer online.

Proiectul demonstrează că și un joc simplu, precum Pong, poate deveni un prilej excelent pentru învățare, testare și aprofundare a conceptelor de rețelistică, animație, sincronizare și interfață grafică. Acest joc nu este doar un exercițiu de programare, ci și o dovadă că pasiunea, combinată cu tehnologia potrivită, poate transforma ideile în realitate digitală

## BIBLIOGRAFIE

- <https://en.wikipedia.org/wiki/C%2B%2B>
- <https://www.sfml-dev.org/download/sfml/2.6.2/>
- <https://visualstudio.microsoft.com>
- <https://github.com/apps/desktop>
- <https://www.radmin-vpn.com>
- <https://ezgif.com>
- <https://www.piskelapp.com/p/create/sprite/>