**Colegiul Național “Liviu Rebreanu” Bistrita**



Lucrare de atestat

,,Breakout’’

- joc arcade -

Candidat: Profesor

îndrumător:

Mureșan Vlad Gălățan Constantin

Bistrita 2020

# Cuprins

1. Descrierea temei și motivarea alegerii ei ……………….… pag 3
2. Descrierea generala a aplicației ……...…………………….… pag 4

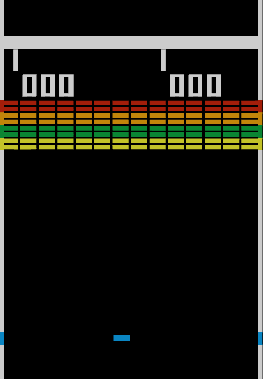
# Descrierea în detaliu a aplicației ……………………………... pag 5

1. Bibliografie …………………………………………………………… pag 11

# 

# 1.Descrierea temei și motivarea alegerii ei

Cred ca mulți tineri și chiar și adulți au fost atrași de computer prin intermediul jocurilor. Cand am început sa studiez informatica la liceu am început sa realizez cat de multă munca și linii de cod stau în spatele acelui mic joc pe care-l jucam.

Aplicația Block breaker, programată în limbajul C++, are la baza ca inspirație un astfel de joc de tip arcade numit *Breakout*. 

*Breakout* este unul dintre cele mai vechi jocuri de tip arcade, fiind publicat în 1976 de compania Atari Inc. A fost conceput de Nolan Bushnell și Steve Bristow și construit și programat de co-fondatorul Apple Steve Wozniak. Placa pe care acesta a construit-o a fost o sursa de inspirație importantă pentru multe aspecte în construirea computerului personal Apple II.

2. Descrierea generala a aplicației

Aplicația este programată în limbajul C++ folosind mediul integrat Qt Creator 5.14.2 și poate fi folosită ca mod de recreere și de asemenea, de înțelegere a principiilor programării orientate pe obiecte și ale programării cu interfața grafică.

În Breakout, o bilă se mișcă pe ecran și sparge o serie de blocuri la coliziunea cu acestea, iar jucătorul are misiunea de menține bila pe ecran controland o paletă și de a sparge toate blocurile.



Aplicația se deschide cu un ecran de titlu unde se oferă informații despre obiectivul jocului și modul în care jucătorul se poate folosi de controale să joace, iar la apăsarea butonului start jocul începe. Jocul se termina cand toate blocurile sunt sparte sau cand bila cade din ecran.

# 3. Descrierea în detaliu a aplicației

Aplicația are la baza desenarea obiectelor în Qt folosind interfața Graphics View Framework. Graphics View oferă o suprafață pentru gestionarea și interacțiunea cu un număr mare de elemente grafice 2D personalizate și un widget pentru vizualizarea elementelor, cu suport pentru zoom și rotire.

Interfața include o arhitectură de propagare a evenimentelor care permite capabilități precise de interacțiune dublă pentru articolele de pe scenă. Elementele pot gestiona evenimentele cheie, apasarea mouse-ului, mutarea, eliberarea și evenimentele cu dublu clic și pot urmări și mișcarea mouse-ului.

Folosind acestea clasa principala *game.h* moștenește clasa QGraphicsView și conține un QGraphicsScene care va afișa și gestiona toate elementele necesare jocului. Astfel am implementat clasele:

- *block.h* pentru blocurile care trebuie sparte de către bila

*- ball.h* care contine toate caracteristicile bilei și mișcării ei

- paddle.h pentru paleta controlata de user

După ce user-ul citește instrucțiunile și apasa Play label-urile si butoanele ce conțineau informații sunt ascunse și se creeaza scena și obiectele care vor fi adaugate pe ea , respectiv matricea blocurilor (fig-1) cu caracteristici stilizate în funcție de linie (fig 2) asemanator jocului original, mingea - care va porni inițial în sus- și paleta care va urmări mișcarea cursorului la apăsarea butonului stang al acestuia (fig-3).

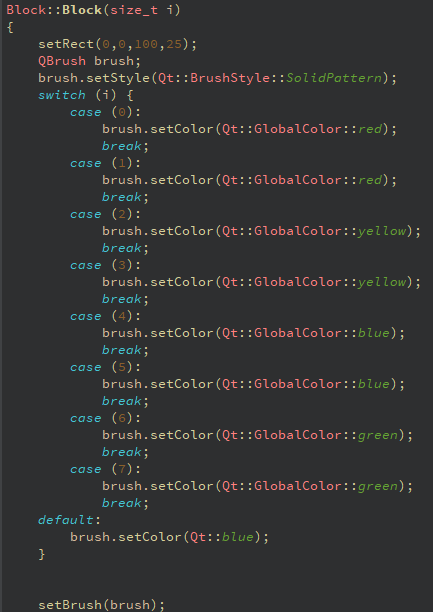
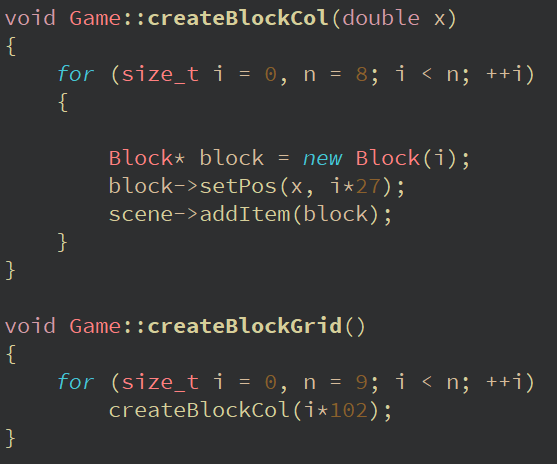


Fig-1

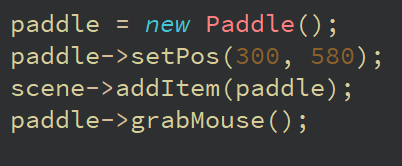
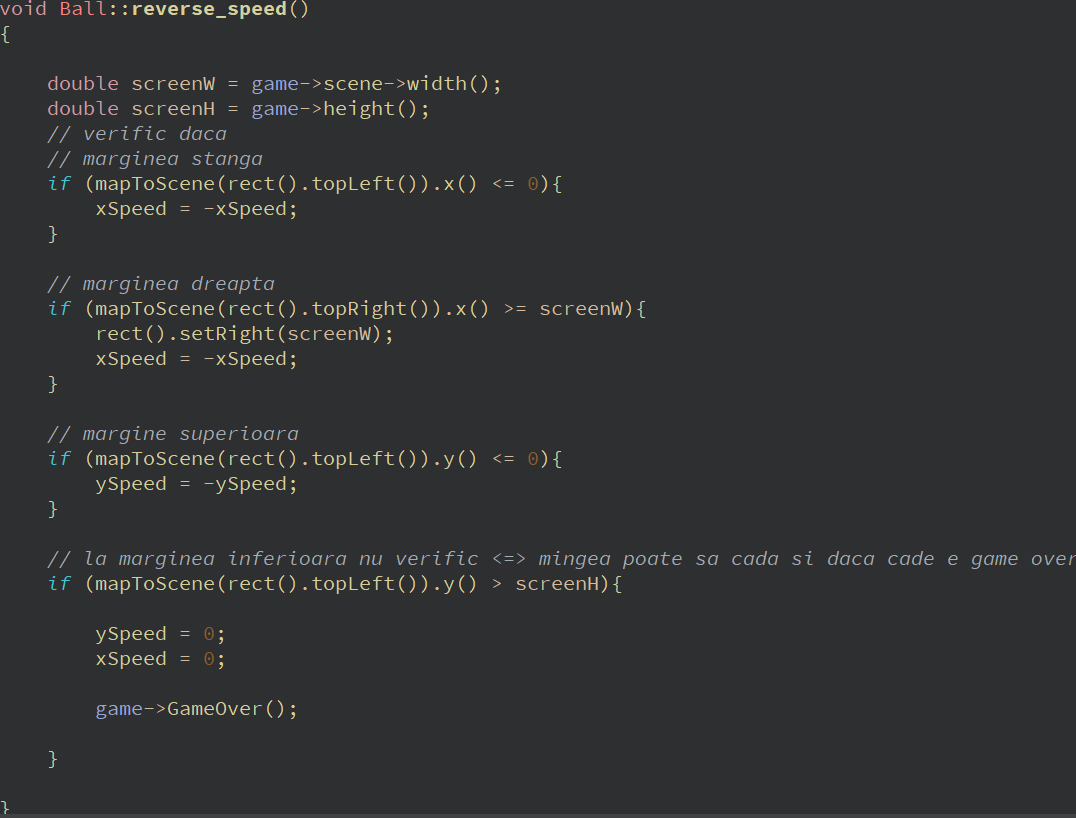


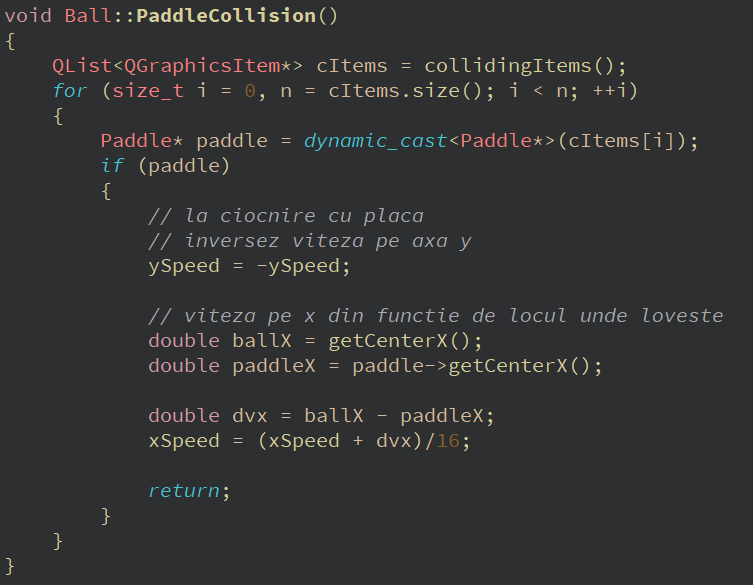
fig-2

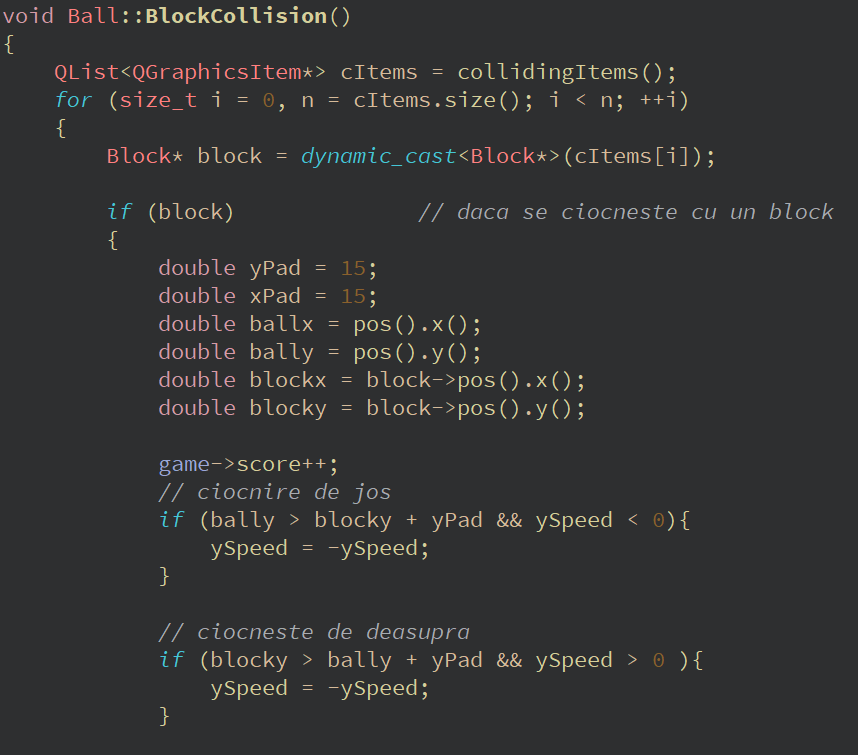
fig-3

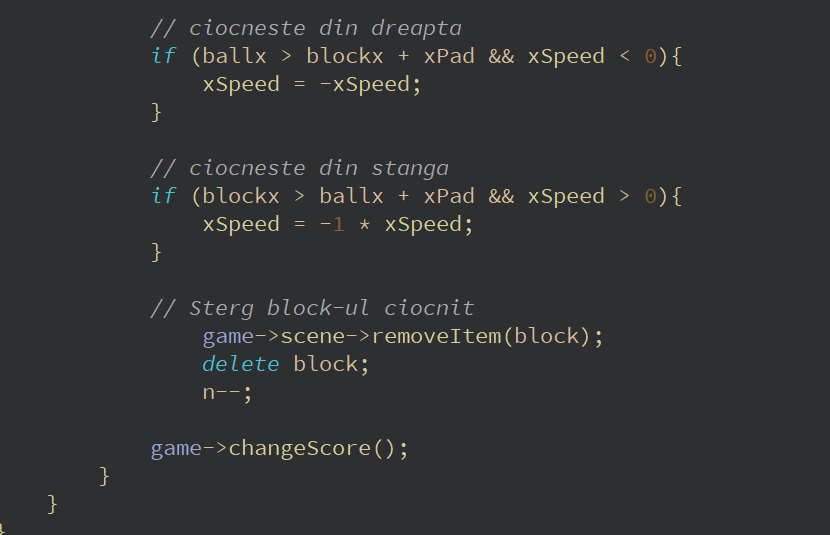
Mișcarea mingi se bazează pe redesenarea acesteia la un interval de timp bine stabilit (folosind un timer) care favorizează fluiditatea mișcării la un set nou de coordonate determinat de doua constante ce reprezinta vitezele pe cele doua direcții ale axelor - xspeed si yspeed. De asemenea, mișcarea se realizeaza verificand de fiecare data un set de condiții reprezentate de tipurile de interacțiuni pe care le poate avea bila:

1. Mișcarea în interiorul scenei prin verificarea coordonatelor bilei raportat la mărimea scenei

În tratarea coliziunilor ne folosim de mijloacele puse la dispoziție de interfața Graphics View Framework ce vizează interacțiunea dintre obiecte. Asfel, putem afla cu exactitate care elemente fac parte din coliziune.

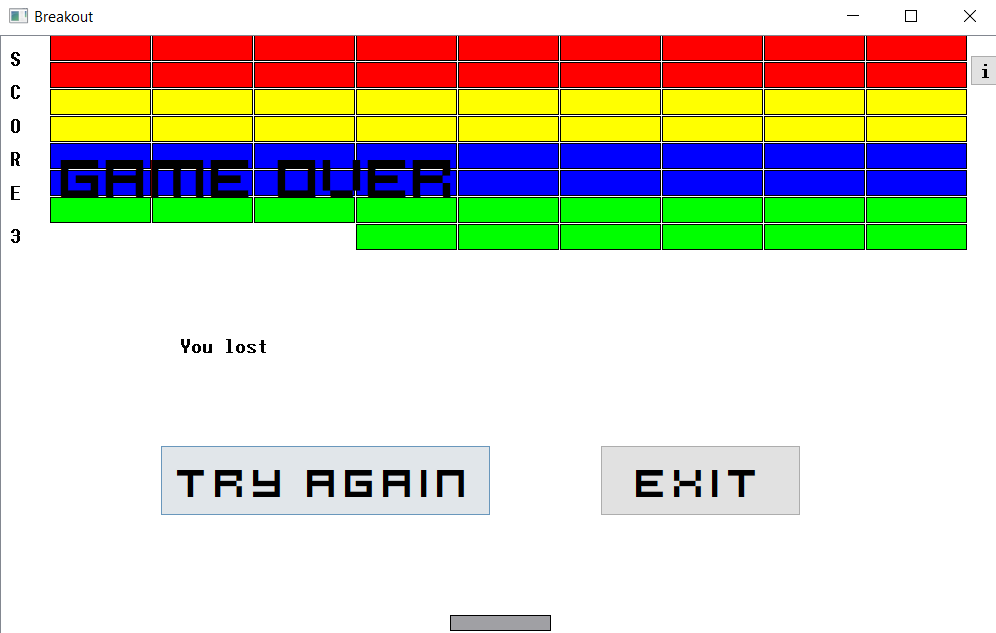
1. Coliziunea cu paleta (*paddle) :* 
2. Coliziunea cu blocurile :





La sfarsit daca jucătorul a completat cu succes obiectivul se afișează următorul mesaj pe ecran :



În caz ca jucătorul pierde se afișează :

# 

# 4. Bibliografie

i) Documentatie Qt:

1. Graphics View Framework : <https://doc.qt.io/qt-5/graphicsview.html>

## The Graphics View Coordinate System :

<https://doc.qt.io/qt-5/graphicsview.html#the-graphics-view-coordinate-system>

# QGraphicsItem Class :

<https://doc.qt.io/qt-5/qgraphicsitem.html>

ii) Forum-uri consultate :

1. StackOverflow :

<https://stackoverflow.com/questions/8061235/qt-breakout-game>

iii) Informatii:

1. Wikipedia:

<https://en.wikipedia.org/wiki/Breakout_(video_game)>