

Pac-Man Game

Descriere

Acest proiect reprezintă o implementare a jocului Pac-Man folosind principiile Programării Orientate pe Obiecte (POO). În cadrul acestui proiect, am avut de îndeplinit mai multe cerințe care m-au ajutat să înțeleg mai bine concepte fundamentale ale POO, cum ar fi moștenirea, polimorfismul, excepțiile, pointerii și funcțiile virtuale. Jocul este implementat în C++ și utilizează biblioteca SFML pentru a crea o interfață grafică simplă.

Funcționalități și concepte implementate

- Moștenirea (Inheritance): Am implementat moștenirea prin clasa de bază Ghost și cele două clase derivate, RandomGhost și ChaseGhost. Aceasta a permis comportamente diferite pentru fantome în funcție de tipul lor.
- Funcții virtuale pure (Pure Virtual Functions): Clasa Player conține funcțiile virtuale pure draw și update, care sunt implementate în clasele derivate. Aceasta asigură polimorfismul în cadrul jocului.
- Apelarea funcțiilor virtuale prin pointeri (Virtual Functions via Pointers): Funcțiile virtuale sunt apelate prin pointeri la clasa de bază în fișierele utile.cpp, în cadrul funcțiilor updateGhosts și drawGhosts, ceea ce demonstrează polimorfismul în acțiune.
- Dynamic casting (Dinamic Casting): Am folosit dynamic_cast pentru a genera comportamente diferite în funcție de tipul fantomei:
 - RandomGhost (fantoma roșie): Când atinge jucătorul, crește viteza tuturor fantomelor.
 - ChaseGhost (fantoma verde): Când atinge jucătorul, duce la game over.
- Excepții (Exceptions): Am implementat excepții pentru a gestiona corect citirea fișierului map.txt, evitând eventualele erori de la încărcarea datelor.
- Atribute și funcții statice (Static Attributes and Functions): Am folosit atribute și funcții statice pentru a număra numărul de fantome din joc, gestionând astfel comportamentele acestora la nivel global.
- Culori și animații (Colors and Animations): Atunci când jucătorul pierde, ecranul devine roșu și fereastra jocului se închide automat după 3 secunde, pentru a oferi o experiență vizuală de sfârșit de joc.

Instrucțiuni de compilare și rulare

Pentru a compila și rula proiectul pe Windows folosind PowerShell, urmați pașii de mai jos:

1. Deschideți un terminal Windows PowerShell. Eu am folosit terminalul din Visual Studio Code.
2. Folosiți următoarea comandă pentru a compila proiectul:

```
g++ src\main.cpp src\Map.cpp src\Player.cpp src\Ghost.cpp src\RandomGhost.cpp src\ChaseGhost.cpp  
src\utile.cpp `  
>> -IC:/Libraries/SFML-2.5.1/include `
```

```
>> -I$PWD/include `  
>> -I$PWD/src `  
>> -LC:/Libraries/SFML-2.5.1/lib `  
>> -lsfml-graphics -lsfml-window -lsfml-system `  
>> -o main.exe
```

După ce proiectul este compilat cu succes, puteți rula jocul folosind comanda: ./main.exe

Stadiul proiectului

Acest proiect este încă într-un stadiu prematur, iar multe elemente urmează să fie implementate înainte de tema 3. Printre îmbunătățirile planificate se numără:

- Efecte sonore pentru o experiență de joc mai interesantă.
- Adăugarea mai multor tipuri de fantome cu comportamente diferite.
- O grafică îmbunătățită pentru hartă și personaje.

De asemenea, sunt planificate îmbunătățiri ale codului și optimizări pentru o performanță mai bună.

Bibliografie

- SFML - Simple and Fast Multimedia Library: <https://www.sfml-dev.org/>
- Documentația SFML (Graphics, Window, System):
<https://www.sfml-dev.org/documentation/2.5.1/>
- Programarea Orientată pe Obiecte în C++ - Tutoriale și Exemple:
<https://www.learncpp.com/> <https://cplusplus.com/doc/tutorial/>
- Stack Overflow: <https://stackoverflow.com/>
- Visual Studio Code & c_cpp_properties.json / tasks.json / launch.json
<https://code.visualstudio.com/docs/cpp/config-msvc>
- Microsoft PowerShell + MinGW (g++) <https://www.mingw-w64.org/>
- Tutoriale de pe YouTube:
 - <https://www.youtube.com/watch?v=vC0d1rDmPBs>
 - <https://www.youtube.com/watch?v=cMhmSmlNeR0>
 - <https://www.youtube.com/watch?v=FeMbLV72Z6M&t=86s>