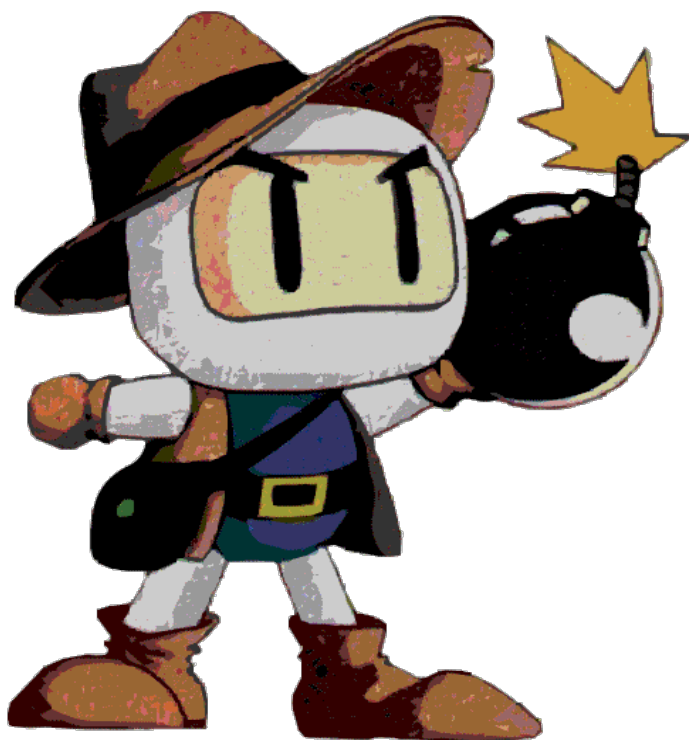


# Proiect Ingineria programelor

## Bomberman Online

### Etapa 3



**Echipa ChaosDev**

Pavaluca Matei

Ciocan Mihai

Palianos Cristian

Cristea Paul

## Prezentare Etapa 3

### Tehnologii utilizate

Aplicatia noastra se imparte in doua componente principale : clientul si serverul.

**Clientul** este implementat in limbajul [Python](#) utilizand biblioteca [pygame](#) pentru dezvoltarea propriu-zisa a jocului si sockets pentru comunicarea cu serverul.

Am considerat o decizie buna alegerea acestui limbaj deoarece ofera atat un timp scurt pentru dezvoltarea aplicatiei cat si suport foarte bun pentru diverse module de care am avea noi (precum sockets si [pygame](#)).

Stocarea si incarcarea hartilor se face folosind fisiere [.XML](#) pentru a beneficia de multiplele avantaje pe care le ofera acest format (in mod special biblioteci de incarcare/salvare si portabilitate).

Pentru aplicatia de **server** am folosit Java si api-ul specific de comunicare pe network prin ServerSocket.

Motivul pentru care am utilizat Java in implementarea jocului este usurinta lucrului cu api-ul pentru partea de networking si faptul ca avem experienta in utilizarea acestui limbaj.

### Descrierea modulelor la nivel inalt

**Aplicatia client** este formata din urmatoarele clase:

- Modulul Principal se ocupa de controlul intregii aplicatii client. El comanda si asigura comunicarea intre celelalte module
- Modulul de comunicare cu serverul. Se ocupa de transmiterea / receptionarea mesajelor de la server. Fiecare client are un socket prin care comunica cu serverul
- Modulul GameWorld se ocupa cu rulearea jocului pe baza datelor primite de la server. Acesta contine tot, de la parsarea inputului pana la desenarea jocului si redarea sunetelor

**Aplicatia server** este formata din urmatoarele clase:

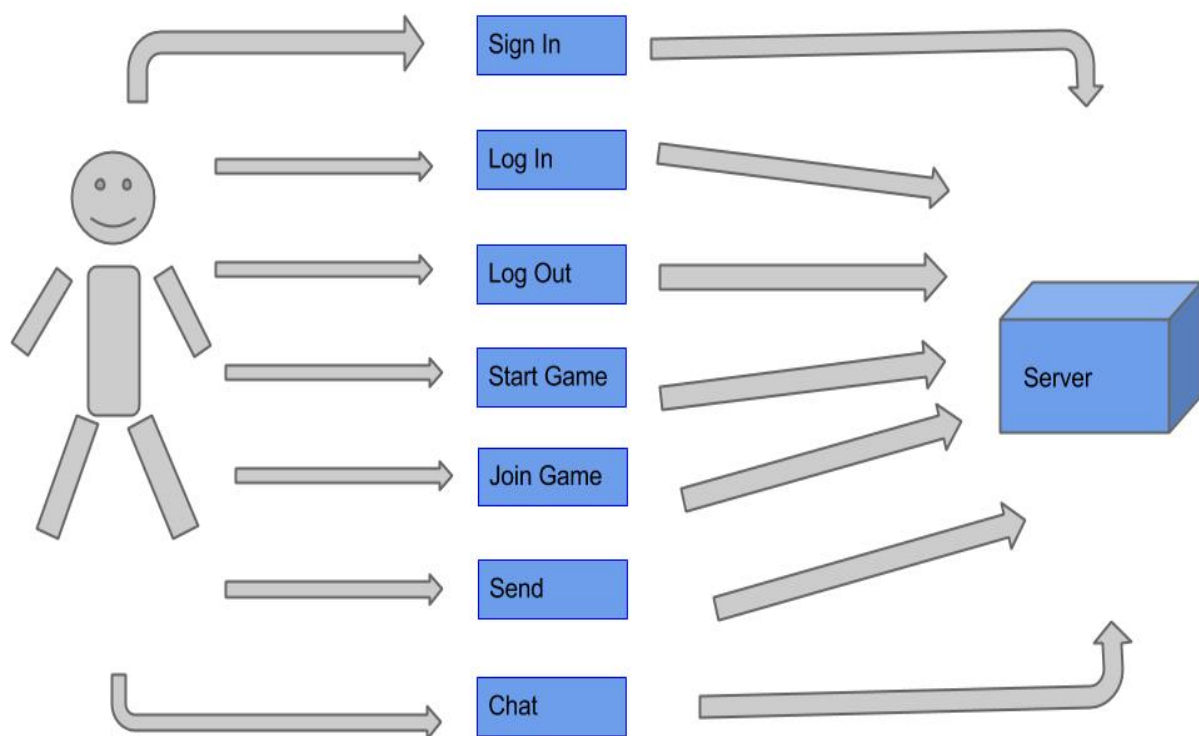
- Server.java ce se ocupa de acceptarea de conexiuni de la clienti fiecare client avand access la un Socket prin care va putea comunica cu serverul pentru derularea jocului
- Client.java defineste un client ce asteapta informatii pe socket pentru updatarea jocului local si poate trimite mesaje despre starea playerului propriu
- GameWorld.java este utila pentru definirea unui joc care are lista de clienti care participa

la joc si retine informatii despre joc cum ar fi harta, id jocului, etc;

- Tile.java defineste tipul de tile de pe harta ce va fi retinuta in GameWorld.java sub forma de matrice
- Diverse clase ajutatoare (POJOs)

## Modul de operare al serverului

Serverul ruleaza un thread ce asteapta conexiuni de la clienti. Clientul va putea practic crea un joc, astfel ceilalti jucatori putand da join pentru a se alatura jocului. Fiecare joc va avea un id propriu si o lista de jucatori participanti. Fiecare client va trimite informatii despre starea curenta a jocului iar serverul va face o serie de verificari asupra noilor pozitii pe harta proprie (mutari ilegale etc.). Serverul va trimite apoi celorlalti jucatori noile informatii pentru ca fiecare sa updateze versiunea locala.



## **Detalii organizatorice**

Pentru varianta de inceput a aplicatiei am realizat teste de comunicare intre server si mai multi clienti pentru verificarea scalabilitatii si timpului de transfer al mesajelor serverul fiind rulat pe un PC normal. In urma acestor teste am realizat faptul ca implementarea noastra va rula corect pentru 4 - 8 clienti in cazul in care serverul este rulat pe un PC si se scaleaza odata cu hardware-ul.