

# PacMan Replica



Starting screen – ul jocului

Realizat de:  
Stahie Dragoş Marian

## CUPRINS:

I.	Descrierea proiectului	.....	2
II.	Diagrama de clase	.....	3
III.	Clase	.....	4
IV.	Mecanicile jocului	.....	5
V.	Provocări întâmpinate	.....	8
VI.	Posibilitați de dezvoltare ulterioară	.....	9
VII.	Bibliografie	.....	10

# I. Descrierea Proiectului

Ținta proiectului a fost recrearea jocului PacMan care a fost lansat ca „Arcade Game” în anul 1980 și l-a avut ca designer pe Toru Iwamatsu.

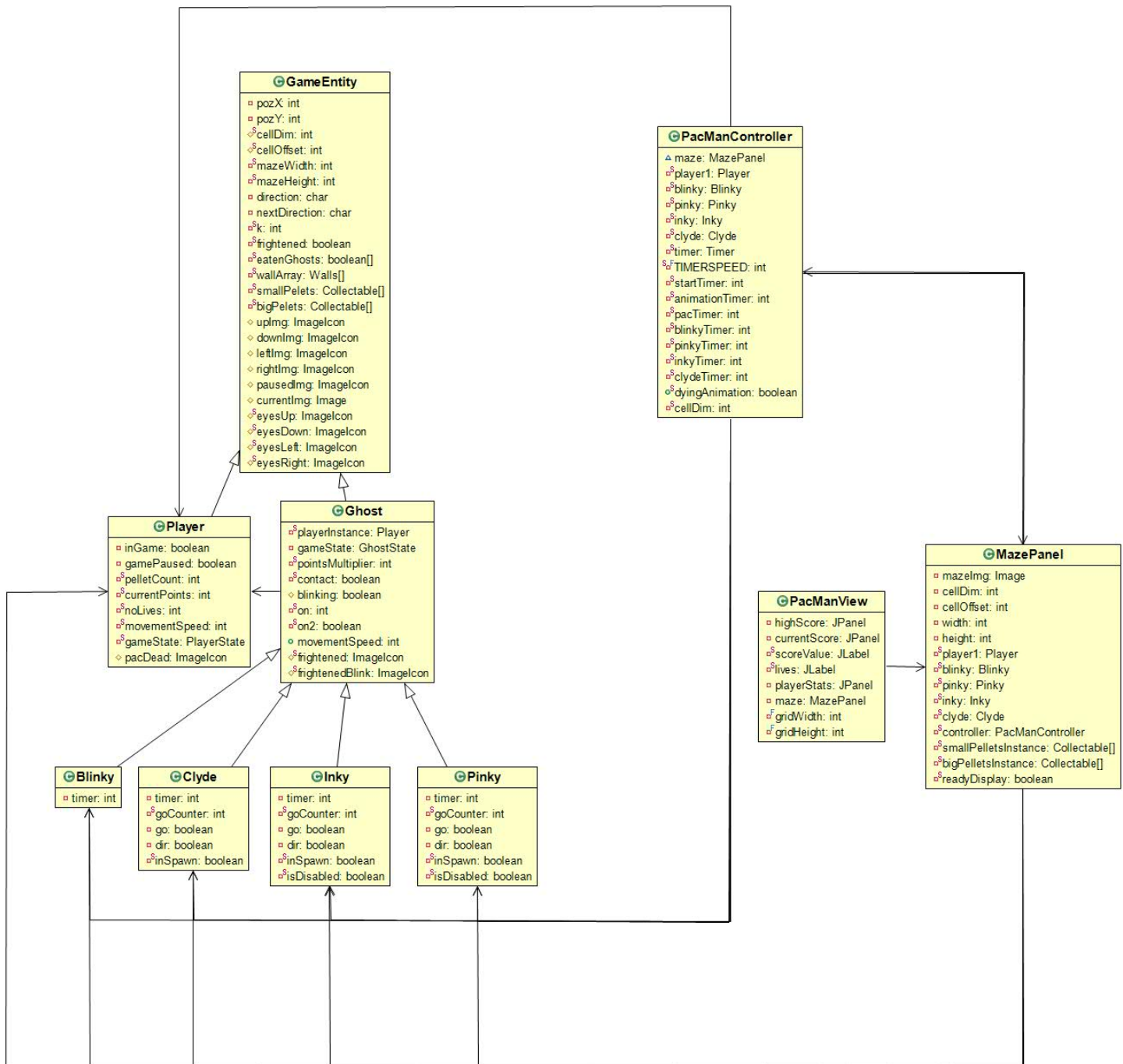
Principalul obiectiv pe care l-am urmărit pe parcursul implementării a fost recrearea mecanicilor de joc atât pentru **jucător**, cât și pentru **fantomele** care îl urmăresc pe acesta.

Jucătorul îl controlează pe PacMan cu săgețile de la tastatură, navigând labirintul cu scopul de a colecta toate pellet-urile. Pe parcursul jocului, fantomele au rolul de a încerca să îl împiedice pe PacMan din a colecta pellet-urile. Dacă una dintre fantome îl prinde pe PacMan, atunci jucătorul pierde o viață din totalul de trei vieți de la început, iar jocul continuă restartând pozițiile pentru PacMan și fantomițe.

În labirint sunt prezente și patru „power pellets” care, atunci când sunt mâncate de PacMan, sperie fantomele și rolurile se inversează pentru câteva secunde, PacMan fiind capabil să mănânce fantomele pentru puncte în plus.

Fiecare pellet valorează 10 puncte, power pellet-urile valorând câte 50 de puncte fiecare. Fiecare fantomă mâncată de către PacMan valorează începând de la 200 de puncte pentru prima și ajungând la 1600 de puncte pentru a patra fantomă mâncată folosind un singur power pellet (a doua valorează 400 și a treia 800, valoarea dublându-se de la o fantomă la cealaltă).

## II. Diagrama de clase



Proiectul a fost implementat după modelul MVC (Model – View – Controller), clasa **GameEntity** și toate subclasele ei fiind **modelul**, clasele **PacManView** și **MazePanel** reprezentând **view-ul**, iar clasa **PacManController** implementând **controllerul**.

### III. Clase

Clasa **GameEntity** este principala clasă din model, conținând parametri și metode aplicabile oricărei entități din joc (atât pentru PacMan cât și pentru fantome).

Clasa **Player** extinde clasa GameEntity, adăugând parametri și metode specifice jucătorului.

Clasa **Ghost** extinde clasa GameEntity, adăugând parametri și metode specifice oricărei fantome.

Clasa **Blinky** extinde clasa Ghost, adăugând parametri și metode care permit fantomei roșii (pe nume Blinky) să se deplaseze și să interacționeze cu PacMan separat de celelalte fantome.

Clasa **Inky** extinde clasa Ghost, adăugând parametri și metode care permit fantomei albastre (pe nume Inky) să se deplaseze și să interacționeze cu PacMan separat de celelalte fantome.

Clasa **Pinky** extinde clasa Ghost, adăugând parametri și metode care permit fantomei roz (pe nume Pinky) să se deplaseze și să interacționeze cu PacMan separat de celelalte fantome.

Clasa **Clyde** extinde clasa Ghost, adăugând parametri și metode care permit fantomei portocalii (pe nume Clyde) să se deplaseze și să interacționeze cu PacMan separat de celelalte fantome.

Clasa **PacManView** este principala clasă din view, conținând parametri și metode cu ajutorul cărora este afișat pe ecran jocul și statisticile legate de joc (numărul de vieți și punctele acumulate).

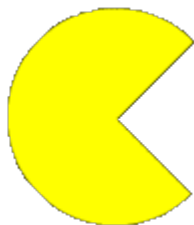
Clasa **MazePanel** face parte tot din view, implementând afișajul labirintului și a entităților din joc.

Clasa **PacManController** este singura clasă care implementează controllerul pentru joc, preluând input-ul jucătorului și traducând-ul în direcții pentru PacMan. Totodată, această clasă mai are atât rolul de a comanda acționarea fantomelor cât și actualizarea imaginilor de pe ecran (apelând metoda `repaint()` pentru `MazePanel`).

## IV. Mecanicile Jocului

Scopul jocului este de a colecta toate pellet-urile și de a termina cu un punctaj cât mai mare. În acest sens, o serie de mecanici au fost implementate pentru a face jocul cât mai interesant și pentru a împiedica fiecare partidă să semene prea mult cu precedenta.

### Modelul de mișcare pentru PacMan:



De la tastatură, jucatorul îl controlează pe PacMan cu ajutorul săgeților. Astfel se selectează următoarea schimbare de direcție, care este efectuată la prima intersecție, cât timp PacMan se poate deplasa mai departe în direcția selectată. În cazul în care deplasarea pe direcția selectată nu este posibilă, PacMan așteaptă la intersecție pentru următorul input al jucatorului.

Când PacMan mănâncă pellet-uri mici, obișnuite, sunt adăugate câte 10 puncte. Totodată PacMan se deplasează cu viteză redusă cât timp mănâncă.

Când PacMan mănâncă oricare dintre cele patru power pellet-uri, viteza acestuia crește și capătă abilitatea de a mânca fantomele pentru un timp limitat. După ce timpul expiră, lucrurile revin la normal, PacMan fiind nevoit să se ferească în continuare de fantome pentru a nu pierde o viață.

Atunci când PacMan este prins de o fantomă, este redată animația corespunzătoare și o viață este scăzută. După aceasta, jocul se reia, toate entitățile fiind resetate la pozițiile inițiale.

PacMan poate trece prin tunelele din laterale pentru a ajunge în partea opusă a labirintului, fără a încetini cât timp este în tunel.

### **Modelul de mișcare pentru fantome:**

Toate fantomele încep în modul SCATTER, după 7 secunde trecând în modul CHASE. În modul SCATTER fiecare fantomă are ținta pe care o urmărește setată în câte un colț al labirintului.

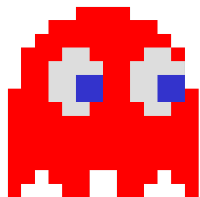
În modul CHASE fiecare fantomă își calculează într-un fel diferit ținta pe care o urmărește pentru a încerca să-l încolțească pe PacMan. Modul CHASE durează 20 de secunde, după care fantomele trec înapoi în modul SCATTER, schimbând astfel între CHASE și SCATTER pe tot parcursul jocului.

Fantomele intră în modul FRIGHTENED atunci când PacMan mănâncă un power pellet. În acest mod, viteza de deplasare le scade la jumătate și nu mai urmăresc o țintă anume pentru a se deplasa spre aceasta, alegând în schimb o direcție aleatoare la fiecare intersecție.

Există două modalități pentru o fantomă de a ieși din modul FRIGHTENED: fie se scurge timpul asociat acestui mod, caz în care trece în modul CHASE; fie este mâncată de către PacMan, trecând în modul EATEN.

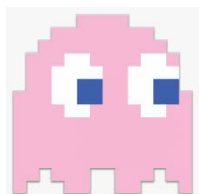
În modul EATEN, fantomele sunt afișate ca o pereche de ochi care se deplasează spre casa fantomelor pentru a reînvia și a intra înapoi în labirint. În acest mod, viteza de deplasare crește considerabil pentru ca fantomele să ajungă înapoi în acțiune cât mai repede posibil.

Fiecare fantomă poate să-și schimbe direcția 180 de grade doar în trei situații: la trecerea din modul CHASE în modul SCATTER și invers și la intrarea în modul FRIGHTENED. Pe lângă această regulă, mai există restricția că fantomele nu pot alege să schimbe direcția în sus atunci când se află în zona de spawn pentru fantome sau în zona de spawn pentru PacMan.



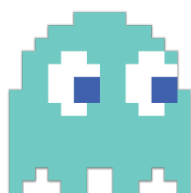
Blinky are cel mai simplu mod de a-și calcula ținta pe care să o urmărească în încercarea de a îl ajunge pe PacMan: coordonatele urmărite sunt în tocmai **coordonatele lui PacMan**.

În modul SCATTER, Blinky are ținta amplasată în colțul din dreapta sus al labirintului, astfel înconjurând peretele din colțul respectiv până la trecerea în alt mod.



Pinky își calculează ținta adăugând 4 pătrățele înaintea lui PacMan (în direcția lui de deplasare). Datorită unui overflow în jocul original (cauzat de faptul că se foloseau variabile pe 8 biți), atunci când PacMan este orientat în sus, Pinky are ținta fixată la 4 pătrățele în fața lui PacMan, dar și la 4 pătrățele la stânga. Pentru a păstra mecanicile jocului funcționând la fel ca originalul, această „eroare” de calcul a fost introdusă intenționat, întrucât variabilele de lucru nu erau pe 8 biți și overflow-ul nu s-ar fi produs de la sine.

În modul SCATTER, Pinky are ținta amplasată în colțul din stânga sus al labirintului, astfel înconjurând peretele din colțul respectiv până la trecerea în alt mod.



Inky își calculează ținta găsind punctul simetric poziției lui Blinky față de poziția lui PacMan.

În modul SCATTER, Inky are ținta amplasată în colțul din stânga jos al labirintului, astfel înconjurând peretele din colțul respectiv până la trecerea în alt mod.



Clyde este cea mai interesantă dintre fantome, având ținta aceeași cu poziția actuală a lui PacMan doar cât timp are cel puțin 8 pătrățele distanța până la PacMan. Când se apropie prea mult, își schimbă ținta în aceeași țintă ca în modul SCATTER, până când iese din nou din raza de 8 pătrățele față de PacMan.

În modul SCATTER, Clyde are ținta amplasată în colțul din dreapta jos al labirintului, astfel înconjurând peretele din colțul respectiv până la trecerea în alt mod.



## V. Provocări Întâmpinate

Pe parcursul implementării funcționalității jocului am întâmpinat diverse provocări. Dintre toate, cea mai importantă a fost dezvoltarea pentru a lua în considerare posibilitatea funcționării jocului în diverse rezoluții.

Pentru a rezolva problema dependenței de rezoluție a obiectelor afișate pe ecran am avut două variante de implementare:

- Folosirea unei matrice pentru a reține unde sunt pereții labirintului și pe unde au voie entitățile să se deplaseze
- Folosirea unui sistem de coordonate pentru pereții labirintului care să se adapteze automat la rezoluția jocului

Prima soluție are avantajul că detectarea coliziunilor cu pereții este foarte simplă și eficientă ca timp de execuție, fiind determinată doar de accesul indexat în cadrul matricei pentru celula unde se dorește executarea verificării. Din cauza multitudinii de probleme la afișarea corectă a labirintului cumulată cu timpul limitat pe care îl aveam pentru dezvoltarea proiectului *am decis să abandonez prima idee în favoarea celei de-a doua variante.*

A doua soluție are atât avantajul preciziei în cazul detectării coliziunilor dintre entități și pereți cât și simplitatea afișării corecte a entităților și labirintului. Dezavantajele acestei implementări sunt dificultatea scrierii coordonatelor pereților pentru a se adapta automat la schimbări de rezoluție și parcurgerea unui vector care conține detalii legate despre fiecare perete (coordoanate de început și final atât pe axa X cat și pe axa Y) de fiecare dată când este necesară detectarea unei coliziuni.

Pentru a facilita adaptarea la diverse rezoluții am decis că reconstruirea dimensiunilor ferestrei se va face pe baza dimensiunii unei celule, în loc de a încerca determinarea dimensiunii unei celule în funcție de rezoluție. (știind că întreg jocul este încadrat într-o matrice de 28 x 36 celule)

## VI. Posibilitați de Dezvoltare Ulterioară

Posibilitățile de dezvoltare ulterioară sunt multe și sunt limitate fie de recrearea fidelă a originalului fie de imaginația dezvoltatorului. Acestea fiind spuse, trei idei de dezvoltare care urmează să fie puse în practică sunt:

Dezvoltarea HUD-ului – implementarea afișajului pentru vieți, scor și scorul record conform jocului original

Dezvoltarea unui sistem de salvare – implementarea unui sistem pentru a putea salva progresul din cadrul jocului

Dezvoltarea opțiunii de a schimba rezoluția – implementarea unui meniu de unde se vor putea selecta diferite rezoluții pentru joc

## VII. Bibliografie

Eclipse IDE: mediul de dezvoltare utilizat pentru implementarea și testarea proiectului.

Adobe Photoshop: programul software folosit pentru decuparea și crearea imaginilor pentru joc.

Sprite Sheet (sursa pentru imaginile din joc): <https://www.sprites-resource.com/fullview/57634/>

AI-ul pentru fantome: <https://www.youtube.com/watch?v=ataGotQ7ir8>

Despre jocul original:  
[https://www.gamasutra.com/view/feature/132330/the\\_pacman\\_dossier.php](https://www.gamasutra.com/view/feature/132330/the_pacman_dossier.php)

Resurse pentru lucrul cu Java Swing:

<https://stackoverflow.com/questions/6222442/overlay-images-on-a-gridlayout-in-java>

<https://stackoverflow.com/questions/3078178/jpanel-in-puzzle-game-not-updating/3078354#3078354>

<https://stackoverflow.com/questions/1466240/how-to-set-an-image-as-a-background-for-frame-in-swing-gui-of-java>

<https://www.codegrepper.com/code-examples/java/how+to+make+an+object+move+with+arrow+keys+in+java>

<https://www.codegrepper.com/code-examples/java/how+to+make+an+object+move+with+arrow+keys+in+java>

<https://stackoverflow.com/questions/8110975/how-to-make-a-rectangle-in-graphics-in-a-transparent-colour>

<https://stackoverflow.com/questions/19868151/resizing-a-jframe-and-maintain-aspect-ratio>