

Jocul: **SokoBall**



Profesor: Student:  
Raluca Brehar Casap Dumitru

**Cuprins**

1. Specificația proiectului----------------3
2. Proiectare -----------------------------4   
   2.1.Planificarea proiectării--------------4   
   2.2.Proiectare în Java-------------------6
3. Implementare--------------------------8
4. Posibilități de dezvoltare ulterioară---9
5. Referințe------------------------------10

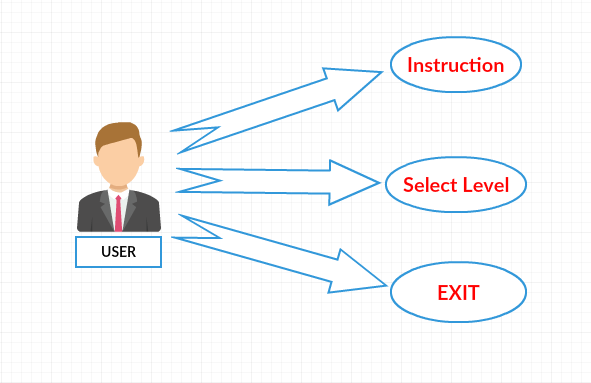
**1.Specificația proiectului**  
 Să se proiecteze jocul SokoBall în limbajul JAVA.Jocul are aceleași principii ca și jocul clasic Sokoban,schimbări făcînduse doar în grafică. Eroul principal este Simson,care este un mare fan al Barcelonei. Folosind butoanele Up,Down,Left și Right Simson poate fi mișcat pe spațiile libere.Scopul jocului este să împingă toate mingele pe pozițiile destinație,care sunt reprezentate prin emblema FC Barcelonei.Odata ce mingea se suprapune peste emblemă, ea se colorează în stil Barca.Mișcările și timpul este limitat.Cu expirarea timpului,mingele se desumflă,iar cînd timpul se termină mingele nu mai pot fi împinse și jocul este pierdut.



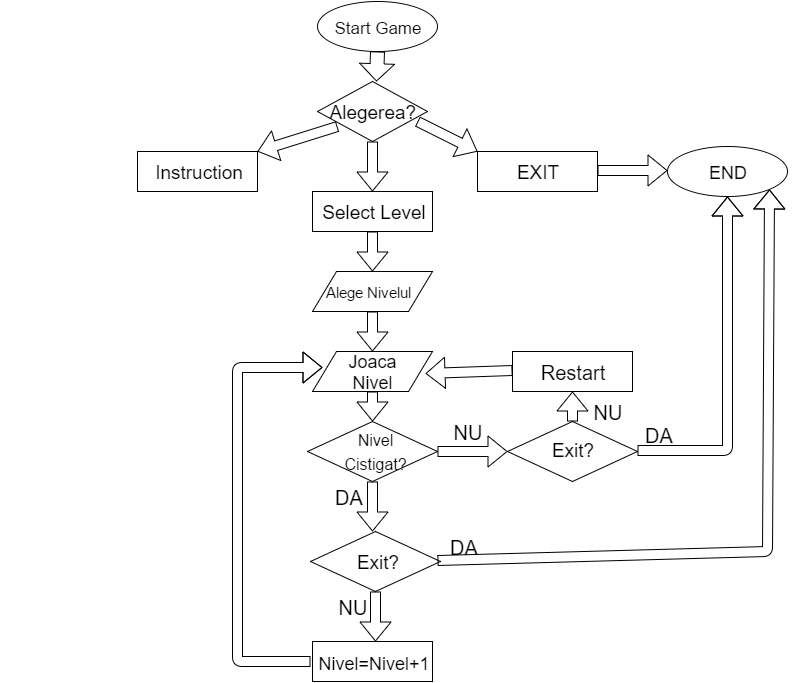
**2.Proiectare**

*2.1.Planificarea proiectării*

Use Case Diagram

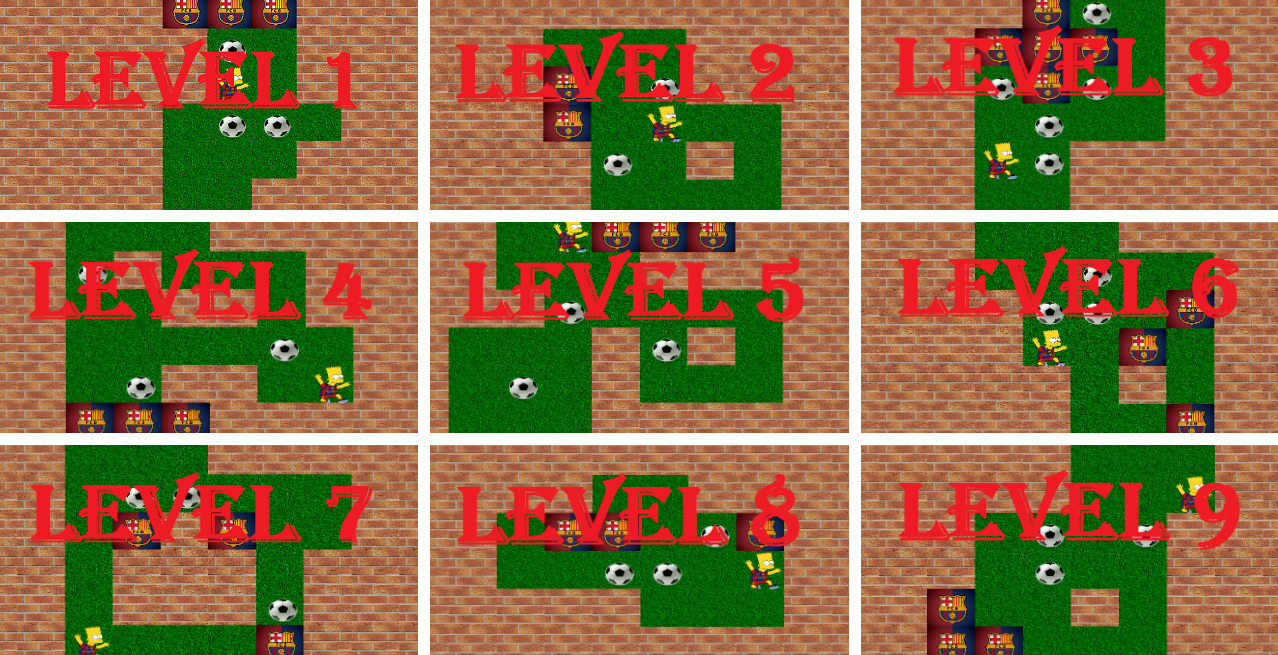


După cum se observă din diagrama de mai sus, utilizatorul va avea următoarele alegeri în meniul principal.Optînd pentru prima alegere,el se poate informa cu regulile jocului și cerința pentru a ieși învingator.Dacă alege “Select Level”,îi apare o nouă pagină,de unde poate să inceapă jocul cu orice nivel dorește.Dacă cumva îi pare jocul plictisitor,sau e dezămagit ca nu poate trece un nivel,sau cazul fericit că a fost atît de bun și a cîștigat la fiecare din nivele,jucătorul poate da click pe Exit.  
Pentru o înțelegere mai bună a funcționării proiectului, mai jos este reprezentat un flow-chart.



*2.2.Proiectare JAVA*

Primul pas a fost crearea clasei Model,unde este descrisă toată logică jocului.Fiecare nivel este reprezentat de o matrice de tip char,care are 4 elemente distincte.  
’+’ -> personajul principal,Simson  
’-‘ ->spațiu liber  
‘$’ -> perete  
‘#’ -> mingea  
Locurile unde trebuie duse mingele sunt date de 2 vectori,unul cu pozițiile rindurilor,iar altul cu pozițiile coloanelor celulei ce reprezintă destinația.  
 Sunt create 4 metode(MoveUp,MoveDown,MoveLeft și MoveRight )pentru deplasarea lui Simson,care fac modificări în matricea cu nivelul actual.Aici sunt incluse toate cazurile posibile de mișcare și sunt interzise cele iregulamentare.Mai este implementată metoda Win de tip boolean,care verifică dacă nu cumva toate mingele sunt pe destinație,în caz ca răspunsul este afirmativ,este returnat TRUE.O altă clasă care este în strînsă legătură cu clasa Model,este Levels.Aici sunt declarate matriciil,pozițiile lui Simson,vectorii cu destinațiile pentru fiecare nivel,care sunt selectate pentru nivelul corespunzator printr-un switch.  
 Fiind creat creirul proiectului,în continuare urmează doar să îi dăm un aspect exterior atrăgator,adică sa facem clasele care țin de grafică. Mai întîi a fost a creat un resource folder, unde au fost puse toate imaginele care vor fi folosite în proiect.Apoi a fost creată prima clasă ce ține de grafică,care reprezinta meniul principal,care este compus dintr-un JFrame și două JPanel.În primul sunt 3 butoane,care au fost descrise și în Use Case Diagram,iar in al dolea JPanel este o imagine. Următoarea clasă care ține de grafică este SelectLevel,care reprezintă un JFrame de tip GridLayout(3 x 3).Sunt incluse 9 butoane,fiecare reprezentînd unul din nivele.Pe buton este afișată imaginea cu nivelul respectiv,ceea ce puteți vedea în imaginea de mai jos.



A treia clasă este Level, unde matricea de tip char este transformată în grafică,fiecarui simbol ($,#,-,+) îi este atribuită o imagine.Pentru fiecare imagine care va fi desenată,este făcută o clasa separată care desenează imaginea atunci cînd este necesitatea. Clasa Level este alcătuită dintr-un JFrame, alcătuit din 2 JPanel.Primul este de tip GridLayout(8 x 10),unde va fi desenată matricea,si partea care reprezintă jocul,iar in al doilea va fi afișat Nivelul,numărul de mișcări rămase,timpul în secunde care se va scurge și butoanele pentru Restart,MainMenu,Exit și NextLevel,dacă Nivelul este cîștigat.Cum arată tot ce am descris pîna acum, se poate observa în imaginea de mai jos.



**3.Implementare**

Pentru că sa puteți încerca jocul pe calculatorul dumneavoastră,există doua posibilități.  
 1.Sa instalați JDK și Eclipse.Cum se face acest lucru corect puteți vedea în urmatorul tutorial :  
<https://www.youtube.com/watch?v=fAj9qYFKONc> .  
După să creați toate clasele care au fost descrise în proiectare, și să rulați de pe butonul RUN clasă principală Menu, care conține metoda main.

2.O alta metodă mai simpla este să instalați Java de pe linkul următor <http://www.java.com/en/download/> .Iar apoi sa deschideți direct aplicația de tip jar,care a fost exportată din Eclipse.

**4.Posibilități de dezvoltare ulterioară**

Proiectul dat ar putea fi îmbunătățit prin următoarele metode:

1.Sa fie create mai multe nivele  
2.Matricea sa fie facută mai mare și să fie desenată mai repede.  
3.Să se facă schimbări de grafică,care ar atrage mai multă lume.  
4.Ceasul să fie afișat mai frumos,spre exemplu ca o clipsidră, sau ca un cronometru.  
5.Sa fie introduse sunete,spre exemplu cînd eroul principal se mișcă,cînd jocul este cîșigat,s-au pierdut să se emite diverse sunete.  
Deci, fiecare avînd preferințele și imaginația sa,poate schimba proiectul în așa fel ca să arate ideal după parerea lui.

**5.Referințe**

1.Cursurile și laboratoarele de la facultate  
2. <https://www.oracle.com/java/index.html>  
3. <http://stackoverflow.com/>   
4.,,Core Java Volume I-Fundamentals “ by Cay S.Horstmann, Gary Cornell.