Logo, company name

Description automatically generated

Sudoku bazat pe culori

Dragusanu Octavian

Grupa 30238

Anul 3

Cuprins

1. Cerinta proiectului
2. Instrumente folosite
3. Limbajul ales
4. Diagrama USE-CASE si de CLASE

1.Cerinta proiectului

Obiectivul acestei teme este familiarizarea cu sablonul architectural ***Model-View-Controller.***

Astfel, am avut de dezvoltat(analiza, proiectare si implementare) o aplicatie desktop pentru un joc de tip sudoku bazat pe culori. Jocul este dezvoltat pe mai multe niveluri (5x5, 6x6, 7x7 si 8x8), cu 5 pana la 8 culori. Pentru a realiza o interfata mai atractiva pentru copiii ce vor rezolva jocul, acesta este prezentat in mai multe variante: cu pucte, triunghiuri sau cerculete, conform figurei 1.

Jucatorii vor putea sa se autentifice si sa-si vizualizeze punctajul. Jucatorii vor avea posibilitatea, de asemenea, sa vizualizeze un graphic cu punctajul propriu raportat la punctajul celorlalti jucatori.

Interfata grafica a aplicatiei va fi disponibilia in cel putin 3 limbi de circulatie internationala(implpicit limba romana).

Shape

Description automatically generated

**Figura 1**. Interfete joc

2.Instrumente folosite si justificarea limbajului

Pentru dezvoltarea mini jocului, am ales sa folosesc limbajul de programare orientat pe obiecte JAVA, pentru ca se supune in totalitate cerintelor de rezolvare, si deoarece sunt mai familiarizat cu acest limbaj. Ca mediu de dezoltare am folosit **IntelliJ IDEA**, dezvoltat de JetBrains.

Pentru a realiza diagramele UML, dupa care m-am ghidat pe parcursul construirii aplicatiei, am folosit **STARUml**.

3.Diagramele UML

1. Diagrama USE-CASE

Diagram

Description automatically generated

**Figura 2**. Diagrama USE-CASE

In Figura 2 este reprezentata in mod grafic posiblitatea interactiunilor ale unui utilizator cu aplicatia. Diagrama cazurilor de utilizare arata diferitele cazuri de utilizare pe care sistemul le are.

Astfel, avem un **Player**, care are posibilitatea sa selecteze dificultatea jocului(5x5, 6x6, 7x7 sau 8x8), sa aleaga cum sa arate culorile(cercuri, patrate sau triunghiuri).

Dupa ce incepe jocul, **Player**-ul trebuie sa puna cate o culoare in fiecare casuta disponibila, prin utilizarea campurilor de text si al butonului INSERT.

In cazul in care **Player**-ul face o alegere gresita, acesta va fi atentionat de aplicatie.

Jocul se termina in momentul in care toate casutele sunt colorate.

1. Diagrama de CLASE

Diagram, engineering drawing

Description automatically generated

**Figura 3**. Diagrama de CLASE

1. Diagramele de pachete

Diagram

Description automatically generated

**Figura 4**. Pachetul Model

Diagram

Description automatically generated

**Figura 5**. Pachetul View

Diagram

Description automatically generated with medium confidence

**Figura 6**. Pachetul Controller

A picture containing diagram

Description automatically generated

**Figura 7.** Pachetul ConnectionPackage