

# Proiect Programare Independență de Platformă

# Cave of the Necromancer




# Cave of the Necromancer



1

ORGANIZAREA ECHIPEI



3

DEMO ȘI MODURI DE  
DEZVOLTARE



5



CONCLUZIE + EXTINDERI  
POSIBILE ÎN VIITOR



2

SCOPUL PROIECTULUI



4

TESTARE ȘI GITHUB



# PREZENTAREA ECHIPEI



**Nicolae  
Taradaciuc**

---

**Șerban - Ilie  
Vicol**

---

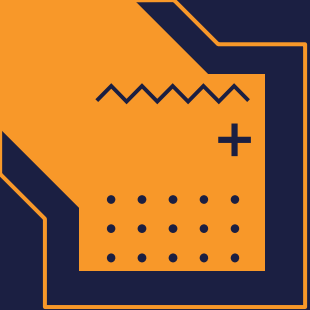


**Alexandru  
Munteanu**

---

**Ovidiu - Florin  
Berea**

---



# Modul de comunicare/întâlniri

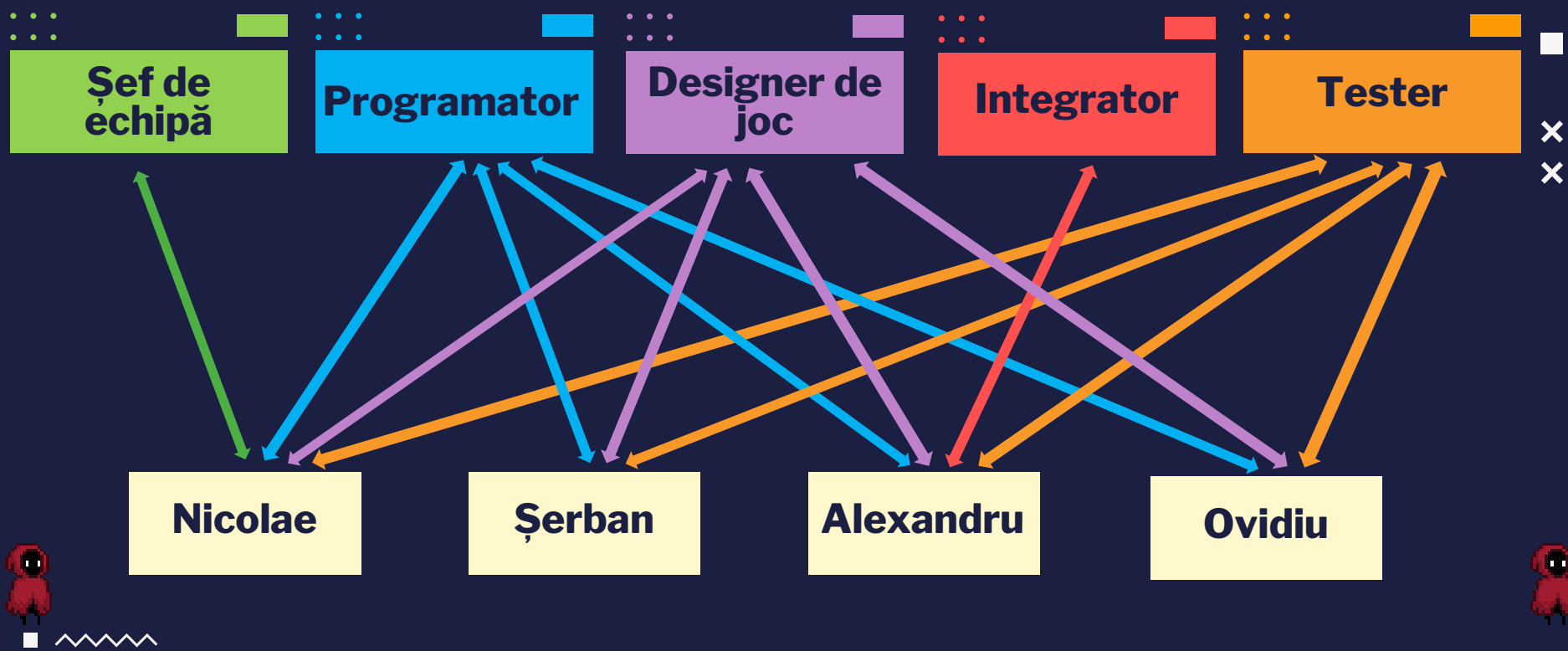


## Echipa noastră :

- **A avut între 1 și 2 întâlniri pe săptămână.**
- **A folosit platforma Discord.**

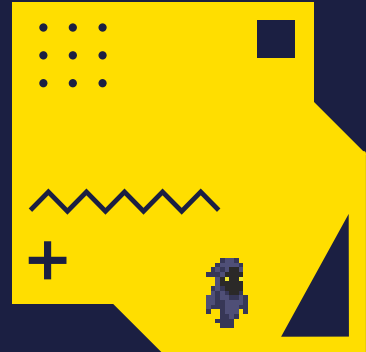
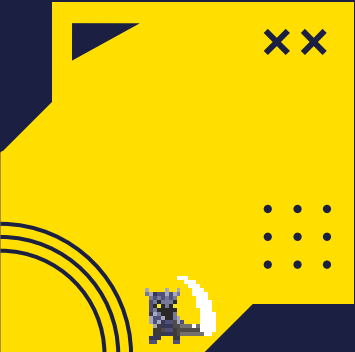


# Roluri asignate în echipă



# Mod de propuneri și stabilire obiective generale / specifice

- Prin evaluarea cerințelor proiectului și înțelegerea generală a jocului
- Stabilirea obiectivelor generale în obiective specifice și măsurabile
  - Propuneri creative și brainstorming
    - Colaborare și comunicare
    - Flexibilitate și ajustare





■  
⋮ **Mod de  
asignare  
task-uri și  
estimări** ×  
×

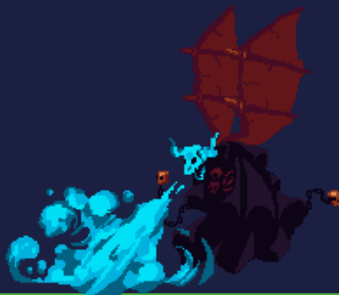
**Task-urile au avut  
niveluri de dificultate diferite  
în funcție de stadiul în care se  
afla jocul la momentul respectiv.**

**Acestea au fost împărțite realist  
pe baza ideilor fiecăruia și modul  
propus de rezolvare.**





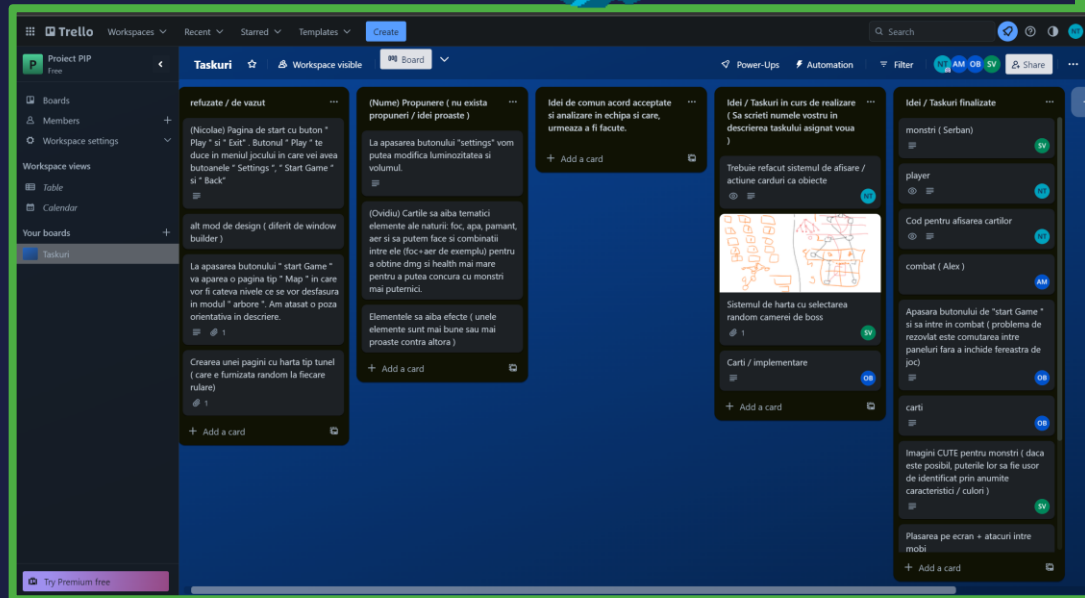
# Task-urile din echipă au fost asigurate folosind Trello, dar și spontan în cadrul întâlnirilor de pe discord.



+

Aceste întâlniri au fost organizate în ziua de joi în cadrul laboratorului de PIP și în zilele de duminică, momente la care prezentam fiecare dintre noi ceea ce am lucrat sau ce nu am reușit să rezolvăm cautând apoi împreună soluții sau redistribuind taskul către cineva cu o abordare mai bună.

×





În timpul procesului de dezvoltare s-au întâlnit unele întârzieri în atingerea anumitor obiective sau deadline-uri.

De obicei aceste obstacole au fost cauzate de provocări tehnice neprevăzute.

La nivel de echipă, au aparut probleme legate de neînțelegeri privind implementările unor metode din joc, existând conflicte legate de integrarea acestora în versiunea comună de joc.

# Probleme aparute la nivel de echipă



De asemenea, au existat numeroase probleme ce tin de partea de versionare cu git, dar si legate de libraria libgdx.



# Cave of the Necromancer

Aspectele cheie ale unui joc roguelike este generarea aleatorie de nivele, gestionarea resurselor (cărților) și supraviețuirea în fața amenințărilor. Fiecare nouă încercare de joc aduce o hartă diferită și unicitate în fiecare sesiune de joc

2

## SCOPUL PROIECTULUI



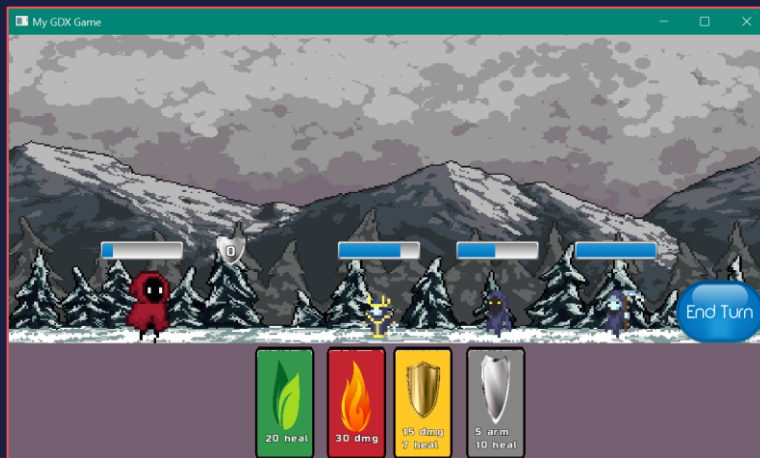
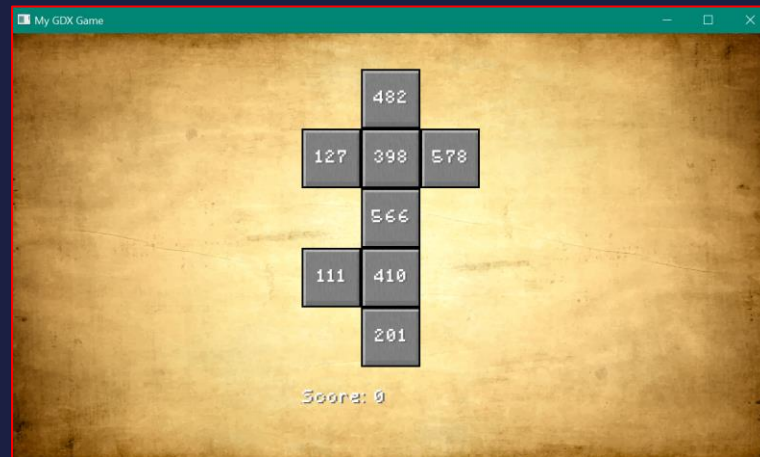
Jocul este bazat pe 3 mari zone de luptă, câmpie, munte și peșteră, semnificând călătoria îndelungată a magicianului nostru, în căutarea monstrului din peșteră. Jucatorul nu va ști exact la ce să se aștepte datorită sistemului roguelike.

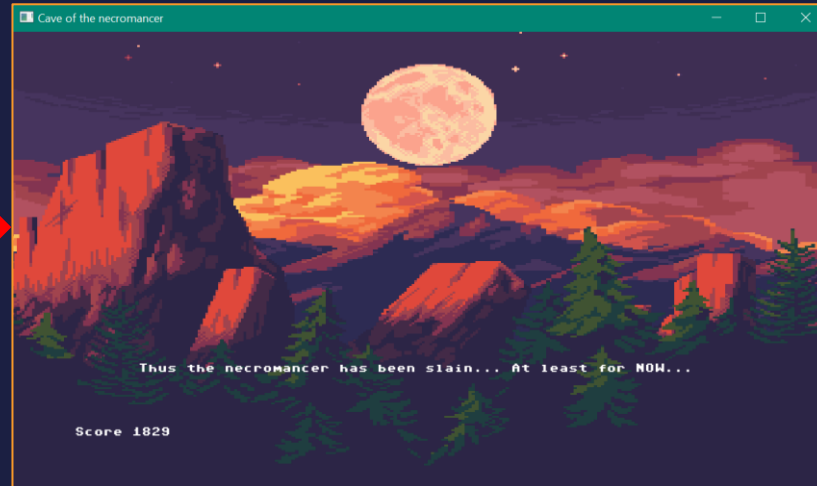
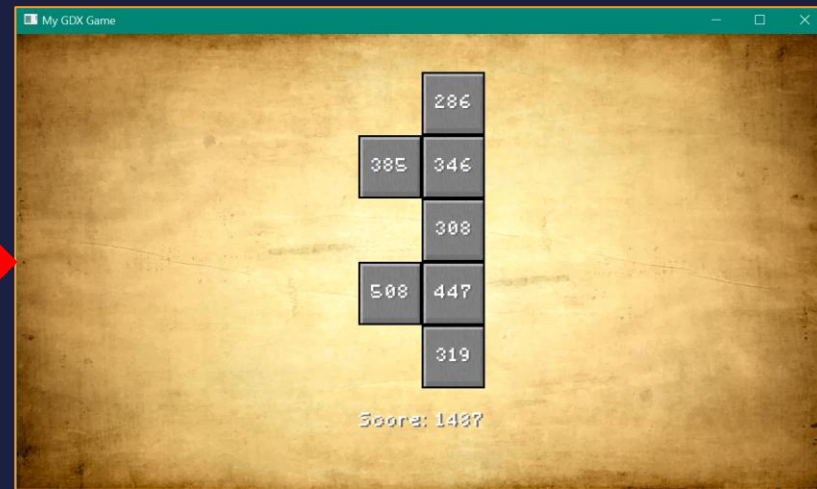
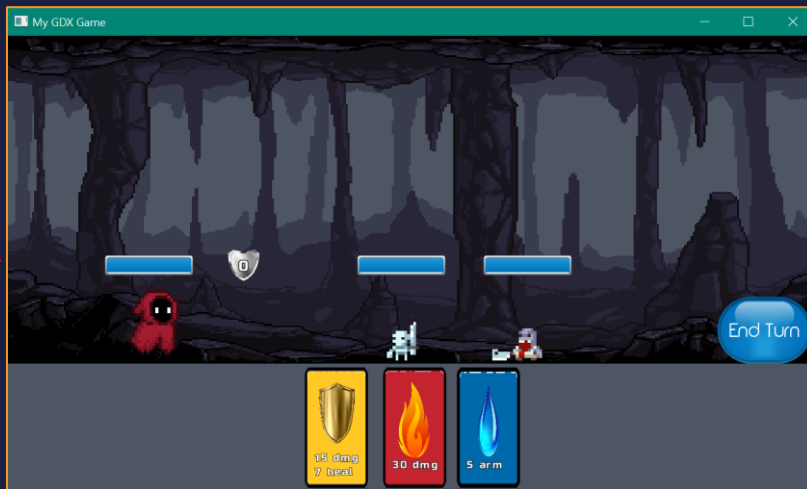
Magicianul nostru va întâmpina o serie de obstacole ce îi pot sau nu încheia călătoria, în funcție de modul său de gestionare eficientă al cărților

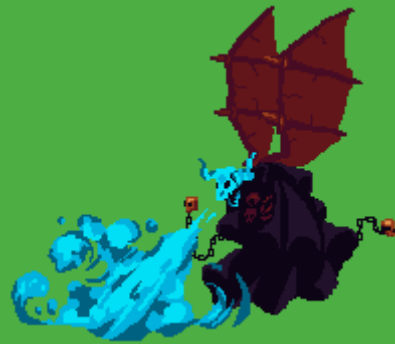
3

# DEMO ȘI PREZENTAREA APLICAȚIEI









✕ ✕





În secțiunea de testare a jocului "Cave of the Necromancer", prin validarea cu teste, au fost urmărite aspecte precum:

- Testarea diverselor acțiuni și combinații de acțiuni pentru a asigura că rezultatele sunt conforme cu așteptările
- Validarea că nivelele sunt generate aleatoriu în fiecare încercare de joc, asigurându-se că hărțile sunt variate și oferă experiențe unice pentru jucător.
- Verificarea că toate cărțile și resursele funcționează corect și pot fi utilizate în mod corespunzător în timpul jocului.
- Testarea pentru interacțiuni corecte între cărți și inamici sau alte mecanici specifice legate de sistemul de cărți.
- Testarea interfeței utilizator pentru a se asigura că este intuitivă și ușor de utilizat, cu elemente de navigare, butoane și informații prezentate în mod clar.





# Utilizare Github



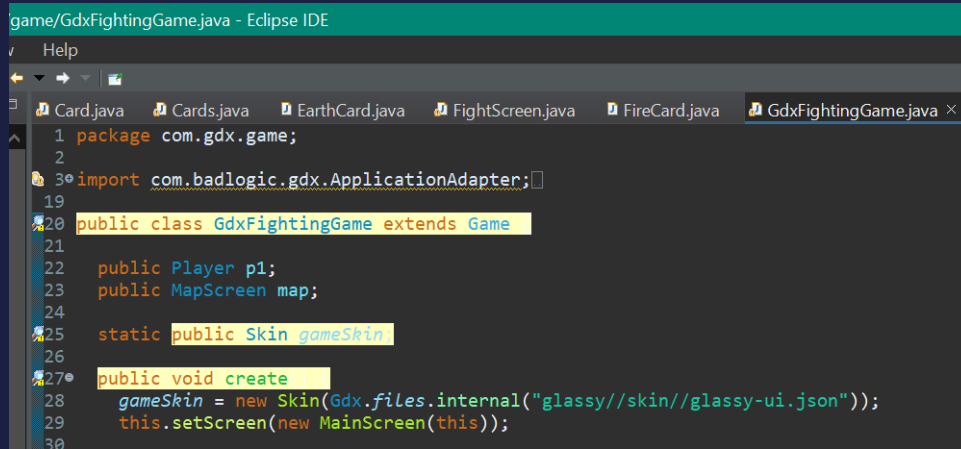
Message	Author	Authored Date	Committer
<b>GameV1.0</b> origin/GameV1.0 <b>HEAD</b> Javadoc Created	TaradaciucNicolae	2023-05-24 23:40:39	TaradaciucNicolae
Game UML Diagram + small changes	Ovidiu-Florin Berea	2023-05-24 22:17:55	Ovidiu-Florin Berea
Solving Checkstyle errors.	TaradaciucNicolae	2023-05-24 19:18:14	TaradaciucNicolae
Merge branch 'GameV1.0' of https://github.com/TaradaciucNicolae/libGDX-Fighting-Card-Game	TaradaciucNicolae	2023-05-24 19:14:15	TaradaciucNicolae
Solving Checkstyle Errors	TaradaciucNicolae	2023-05-24 19:13:38	TaradaciucNicolae
Game title changed.	Vicol Serban-Ilie	2023-05-24 18:18:31	Vicol Serban-Ilie
Possible new end turn button style.	Vicol Serban-Ilie	2023-05-24 18:13:10	Vicol Serban-Ilie
Now armour sets correctly through rooms.	Vicol Serban-Ilie	2023-05-24 18:06:13	Vicol Serban-Ilie
Removed Debug lines and changed the damage for FireCards.	TaradaciucNicolae	2023-05-24 17:46:53	TaradaciucNicolae
Removed Settings Btn/Added skins for Btms/ Repositioned the Volume Btn	TaradaciucNicolae	2023-05-24 17:40:29	TaradaciucNicolae
Assets added for MainScreen	Vicol Serban-Ilie	2023-05-24 17:26:12	Vicol Serban-Ilie
New background for mainscreen.	Vicol Serban-Ilie	2023-05-24 17:25:02	Vicol Serban-Ilie
Start zone and zone3 connection to endscreen fix.	Vicol Serban-Ilie	2023-05-24 16:24:27	Vicol Serban-Ilie
Alex done again	axel_alex	2023-05-24 14:14:52	axel_alex
Alex done	axel_alex	2023-05-24 01:04:47	axel_alex
Javadoc and checkstyle on all classes.	Vicol Serban-Ilie	2023-05-22 12:42:04	Vicol Serban-Ilie
Merge branch 'GameV1.0' of https://github.com/TaradaciucNicolae/libGDX-Fighting-Card-Game	TaradaciucNicolae	2023-05-22 10:04:35	TaradaciucNicolae
Player Test	TaradaciucNicolae	2023-05-22 10:03:34	TaradaciucNicolae
Merge branch 'GameV1.0' of https://github.com/TaradaciucNicolae/libGDX-Fighting-Card-Game	Gitea	2023-05-22 09:48:33	Gitea
Merge branch 'GameV1.0' of https://github.com/TaradaciucNicolae/libGDX-Fighting-Card-Game	Gitea	2023-05-22 09:47:42	Gitea
MovesTest and MyAnimationTest	Vicol Serban-Ilie	2023-05-22 09:47:26	Vicol Serban-Ilie
Cards Test Implement	Ovidiu-Florin Berea	2023-05-22 09:39:25	Ovidiu-Florin Berea
Cards Test Implement	Ovidiu-Florin Berea	2023-05-22 09:31:21	Ovidiu-Florin Berea
Merge branch 'GameV1.0' of https://github.com/TaradaciucNicolae/libGDX-Fighting-Card-Game	Vicol Serban-Ilie	2023-05-22 09:12:22	Vicol Serban-Ilie
hpBars + fight 3 monster	axel_alex	2023-05-22 01:11:58	axel_alex
BossTest	Vicol Serban-Ilie	2023-05-21 23:56:29	Vicol Serban-Ilie
fix for card fit	axel_alex	2023-05-21 23:48:08	axel_alex
Merge branch 'OvidiuB222' into GameV1.02222	axel_alex	2023-05-21 22:55:50	axel_alex
Integrated Nicu , added controlled sound from main menu	axel_alex	2023-05-21 22:25:48	axel_alex
origin/OvidiuB Card Design Update Final	Ovidiu-Florin Berea	2023-05-21 22:12:37	Ovidiu-Florin Berea
Number of monsters dependent on room score	axel_alex	2023-05-21 21:50:21	axel_alex
added monsters tests	axel_alex	2023-05-21 21:41:29	axel_alex
Integration Junit+ Serban code	axel_alex	2023-05-21 21:35:33	axel_alex
origin/VSerban More than one monster added and score in MapScreen	Vicol Serban-Ilie	2023-05-21 21:29:45	Vicol Serban-Ilie
Javadoc for EndScreen	Vicol Serban-Ilie	2023-05-21 19:55:12	Vicol Serban-Ilie
Cards Design Update	Ovidiu-Florin Berea	2023-05-21 19:39:19	Ovidiu-Florin Berea
Sound for EndScreen.	Vicol Serban-Ilie	2023-05-20 16:07:29	Vicol Serban-Ilie
Boss class and sprites added.	Vicol Serban-Ilie	2023-05-20 15:58:24	Vicol Serban-Ilie
MapScreen,FightScreen,Zone2,Zone3,EndScreen,new Assets	Vicol Serban-Ilie	2023-05-20 14:44:23	Vicol Serban-Ilie

Early implementation of a Zone 2. Changed:FightScreen,MapScreen and Game	Vicol Serban-Ilie	2023-05-19 08:42:44	Vicol Serban-Ilie
Combining basegame with latest mapscreen updates.	Vicol Serban-Ilie	2023-05-18 21:34:11	Vicol Serban-Ilie
First version of a score system.	Vicol Serban-Ilie	2023-05-18 21:00:33	Vicol Serban-Ilie
MapScreen Javadoc.	Vicol Serban-Ilie	2023-05-17 22:31:09	Vicol Serban-Ilie
Javadoc for MapScreen class.	Vicol Serban-Ilie	2023-05-17 22:26:55	Vicol Serban-Ilie
OvidiuB Cards Class Optimization + Image + Documentation + JUnit Test Getters/Setters (fail)	Ovidiu-Florin Berea	2023-05-16 23:26:35	Ovidiu-Florin Berea
Cards added element update	Ovidiu-Florin Berea	2023-05-16 17:08:32	Ovidiu-Florin Berea
Background map, random monster texture method and base boss class added	Vicol Serban-Ilie	2023-05-14 20:21:05	Vicol Serban-Ilie
Class card implement	Ovidiu-Florin Berea	2023-05-11 15:31:29	Ovidiu-Florin Berea
fight screen update	axel_alex	2023-05-11 14:12:13	axel_alex
image	axel_alex	2023-05-11 00:46:03	axel_alex
reparat carti	axel_alex	2023-05-11 00:45:52	axel_alex
Merge branch 'BOvidiu' into Integrate	axel_alex	2023-05-10 19:01:17	axel_alex
Merge branch 'TNicolae' into Integrate	axel_alex	2023-05-10 18:56:59	axel_alex
integrate serban	axel_alex	2023-05-10 18:45:10	axel_alex
Card component update v2	Ovidiu-Florin Berea	2023-05-10 17:55:48	Ovidiu-Florin Berea
The MapScreen class implementation is done.	Vicol Serban-Ilie	2023-05-10 17:08:08	Vicol Serban-Ilie
First batch of modifications	Vicol Serban-Ilie	2023-05-10 16:48:46	Vicol Serban-Ilie
Card component v1	Gitea	2023-05-10 16:25:50	Gitea
Card component	Ovidiu-Florin Berea	2023-05-10 00:32:37	Ovidiu-Florin Berea
I created a chance(20 / 30% / 50%) of winning a card after each end turn	TaradaciucNicolae	2023-05-10 00:28:09	TaradaciucNicolae
I added card assets	TaradaciucNicolae	2023-05-09 23:14:02	TaradaciucNicolae
implementation of pack, discarded cards, end turn and visuals of cards	TaradaciucNicolae	2023-05-09 23:14:02	TaradaciucNicolae
added minimonster pack	axel_alex	2023-05-08 22:51:04	axel_alex
MapScreen - almost done	Vicol Serban-Ilie	2023-05-08 22:12:44	Vicol Serban-Ilie
background assets	axel_alex	2023-05-08 20:47:24	axel_alex
modificari pentru clasele vechi	axel_alex	2023-05-08 20:47:07	axel_alex
Merge remote-tracking branch 'origin/Alex2' into BOvidiu	Gitea	2023-05-08 19:07:04	Gitea
Pull update	Ovidiu-Florin Berea	2023-05-08 19:02:26	Ovidiu-Florin Berea
fix	axel_alex	2023-05-03 23:04:07	axel_alex
+music	axel_alex	2023-05-03 22:51:49	axel_alex
added background music and background images	axel_alex	2023-05-03 22:51:27	axel_alex
Animation test	Vicol Serban-Ilie	2023-04-27 14:58:33	Vicol Serban-Ilie
Creating new branch based on Alex branch	Vicol Serban-Ilie	2023-04-26 14:59:07	Vicol Serban-Ilie
master origin/master GDX Project Creation	TaradaciucNicolae	2023-04-21 13:08:22	TaradaciucNicolae
Initial commit	TaradaciucNicolae	2023-04-21 12:39:24	GitHub

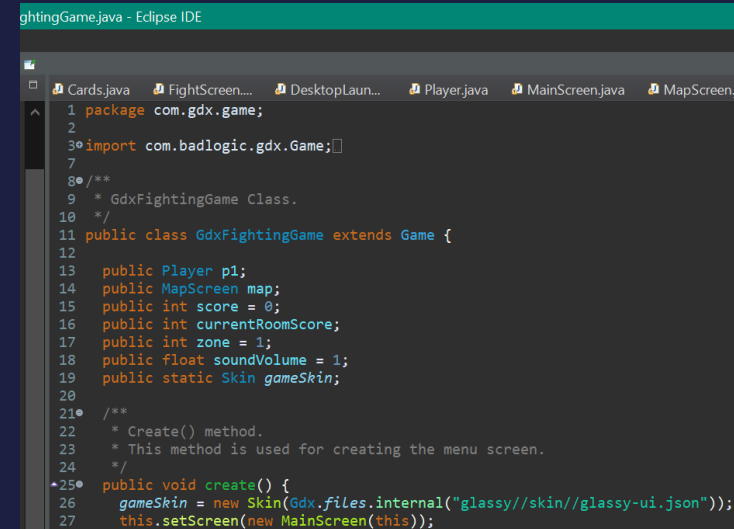
# Check style

Secțiunea Checkstyle a proiectului "Cave of the Necromancer" are ca scop asigurarea unui stil de codificare coerent și aplicarea unor standarde de calitate a codului.

Aceasta include alinierea corectă a codului, utilizarea de denumiri semnificative pentru variabile și funcții, gestionarea adecvată a spațiilor și indentării



```
game/GdxFightingGame.java - Eclipse IDE
Help
Card.java Cards.java EarthCard.java FightScreen.java FireCard.java GdxFightingGame.java x
1 package com.gdx.game;
2
3 import com.badlogic.gdx.ApplicationAdapter;
4
5 public class GdxFightingGame extends Game
6
7     public Player p1;
8     public MapScreen map;
9
10 static public Skin gameSkin
11
12 public void create
13     gameSkin = new Skin(Gdx.files.internal("glassy//skin//glassy-ui.json"));
14     this.setScreen(new MainScreen(this));
15
```



```
ghtingGame.java - Eclipse IDE
Cards.java FightScreen... DesktopLaun... Player.java MainScreen.java MapScreen...
1 package com.gdx.game;
2
3 import com.badlogic.gdx.Game;
4
5 /**
6  * GdxFightingGame Class.
7  */
8 public class GdxFightingGame extends Game {
9
10     public Player p1;
11     public MapScreen map;
12     public int score = 0;
13     public int currentRoomScore;
14     public int zone = 1;
15     public float soundVolume = 1;
16     public static Skin gameSkin;
17
18     /**
19      * Create() method.
20      * This method is used for creating the menu screen.
21      */
22     public void create() {
23         gameSkin = new Skin(Gdx.files.internal("glassy//skin//glassy-ui.json"));
24         this.setScreen(new MainScreen(this));
25     }
26 }
27
```

```

y/game/MainScreen.java - Eclipse IDE
w Help
+ + +
  Card.java  Cards.java  EarthCard.java  FightScreen.java  FireCard.java  GdxFightingGame.java  MainScreen.java
1 package com.gdx.game;
2
3 import com.badlogic.gdx.Gdx;
4
5 public class MainScreen implements Screen {
6     private Skin skin;
7     private Skin skin2;
8     private Skin skin3;
9     private Skin skin4;
10    private TextButton button;
11    private TextButton button2;
12    private TextButton button3;
13
14    private Label nameLabel;
15    private TextField nameText;
16    private Label addressLabel;
17    private TextField addressText;
18    private Table table;
19    private Table tableForVolume;
20
21    private GdxFightingGame game;
22    private Stage stage;
23
24    Music sound;
25
26    TextureRegion
27    Texture back;
28    private SpriteBatch batch;
29
30    public MainScreen(GdxFightingGame game) {
31        game = game;
32        game.map = new MapScreen(game);
33        // TODO: Auto-generated constructor stub
34    }
35
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```



```

green.java - Eclipse IDE
+ + +
  Cards.java  FightScreen...  DesktopLaun...  Player.java
1 package com.gdx.game;
2
3 import com.badlogic.gdx.Gdx;
4
5 /**
6  * Class MainScreen.
7  */
8 public class MainScreen implements Screen {
9     private Skin skin;
10    private Skin skin2;
11    private Skin skin4;
12    private TextButton button;
13    private TextButton button2;
14    private Label nameLabel;
15    private Table table;
16    private Table tableForVolume;
17    private GdxFightingGame game;
18    private Stage stage;
19    Music sound;
20    TextureRegion e;
21    Texture back;
22    private SpriteBatch batch;
23
24    /**
25     * MainScreen constructor.
26     */
27    public MainScreen(GdxFightingGame game) {
28        game = game;
29        game.map = new MapScreen(game);
30    }
31
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

y/game/MapScreen.java - Eclipse IDE
w Help
+ + +
  Card.java  Cards.java  EarthCard.java  FightScreen.java  FireCard.java  GdxFightingGame.java  MainScreen.java  MapScreen.java
1 package com.gdx.game;
2
3 import java.util.Random;
4
5 public class MapScreen implements Screen {
6     private GdxFightingGame game;
7     private Stage stage;
8     private int[][] fields;
9     private TextButton[][] buttons;
10    private Random random;
11    private Skin skin;
12    private Table table;
13    private int currentrow = 1;
14
15    /**
16     * @param game - GdxFightingGame instance
17     * @param fields - fields that need to be populated on the matrix of
18     * @param buttons - A matrix which contains all of the buttons we have to
19     * @param random - A parameter that we use for generating random numbers
20     */
21    public MapScreen(GdxFightingGame game) {
22        this.game = game;
23        stage = new Stage();
24        fields = new int[5][5];
25        buttons = new TextButton[5][5];
26        random = new Random();
27        skin = new Skin(Gdx.files.internal("craftacular/skin/craftacular-ui.json"));
28        table = new Table();
29        table.setFillParent(true);
30        fieldPopulation();
31        stage.addActor(table);
32        table.debug();
33    }
34
35    /**
36     * Return the current row where we are in the matrix of buttons
37     */
38    public int currentrow {
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```



```

green.java - Eclipse IDE
+ + +
  Card.java  FightScreen...  DesktopLaun...  Player.java  MainScreen...  MapScreen.java  Zone2.java  Zone3.java
1 package com.gdx.game;
2
3 import java.util.Random;
4
5 /**
6  * MapScreen Class.
7  */
8 public class MapScreen implements Screen {
9     private GdxFightingGame game;
10    private Stage stage;
11    private int[][] fields;
12    private TextButton[][] buttons;
13    private Random random;
14    private Skin skin;
15    private Table table;
16    private int currentrow = 0;
17    private Label score;
18    private Table tableScore;
19    /**
20     * MapScreen constructor.
21     */
22    public MapScreen(GdxFightingGame game) {
23        this.game = game;
24        stage = new Stage();
25        fields = new int[5][5];
26        buttons = new TextButton[5][5];
27        random = new Random();
28        skin = new Skin(Gdx.files.internal("craftacular/skin/craftacular-ui.json"));
29        table = new Table();
30        tableScore = new Table();
31        score = new Label("score");
32        table.setFillParent(true);
33        tableScore.setFillParent(true);
34        background = new TextureRegionDrawable(new TextureRegion(new Texture("Map/MapBackground.png")));
35        background.setFillParent(true);
36        background.setZIndex(1);
37        fieldPopulation();
38        stage.addActor(table);
39    }
40
41    /**
42     * CurrentRow method.
43     */
44    public int CurrentRow() {
45        return currentrow;
46    }
47
48 }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

x/game/MetalCard.java - Eclipse IDE
package com.gdx.game;

import com.badlogic.gdx.Gdx;

public class MetalCard extends Cards {

    MetalCard(int armour, int health) {
        this.table = new Table();
        Texture idlesheet = new Texture(Gdx.files.internal("badlogic.jpg"));
        TextureRegion tmp = TextureRegion.split(idlesheet, idlesheet.getWidth() / 5, idlesheet.getHeight());
        table.setBackground(new TextureRegionDrawable(tmp[0][1]));
        Skin gameSkin = new Skin(Gdx.files.internal("glassy/skin/glassy-ui.json"));
        this.armour = armour;
        this.health = health;
        Label l1 = new Label(this.armour + "arm", gameSkin);
        Label l2 = new Label(this.health + "health", gameSkin);
        table.add(new Image(l1).width(70).height(120).right().top());
        table.add(new Image(l2).width(70).height(120).right().top());
        table.row();
        table.add(l1).width(60);
        table.add(l2).width(60);
        table.debug();
        l1.setFontScale(0.7f);
        l2.setFontScale(0.7f);
    }
}

```



```

Cardjava - Eclipse IDE
package com.gdx.game;

import com.badlogic.gdx.Gdx;

/**
 * MetalCard class.
 */
public class MetalCard extends Cards {

    /**
     * MetalCard constructor.
     *
     * @param health - amount of heal the MetalCard gives
     * @param armour - amount of armour the MetalCard gives
     */
    MetalCard(int armour, int health) {
        this.table = new Table();
        Texture im = new Texture(Gdx.files.internal("cardselement/metal.png"));
        Texture idlesheet = new Texture(Gdx.files.internal("cards/pixelCardAsset.png"));
        TextureRegion[] tmp = TextureRegion.split(idlesheet, idlesheet.getWidth() / 5, idlesheet.getHeight());
        table.setBackground(new TextureRegionDrawable(tmp[0][1]));
        Skin gameSkin = new Skin(Gdx.files.internal("arcade/skin/arcade-ui.json"));
        this.armour = armour;
        this.health = health;
        Label l1 = new Label(this.armour + " arm", gameSkin);
        Label l2 = new Label(this.health + " heal", gameSkin);
        table.add(new Image(im).width(60).height(120).right().top());
        table.add(l1).width(60);
        l1.setFontScale(0.7f);
        table.add(l2).width(60);
        l2.setFontScale(0.7f);
    }
}

```

```

x/game/Player.java - Eclipse IDE
package com.gdx.game;

import java.util.ArrayList;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.TextureRegion;
import com.badlogic.gdx.scenes.scene2d.ui.Label;
import com.badlogic.gdx.scenes.scene2d.ui.Table;

public class Player {

    /**
     * @param idlesheet - Armura curenta a jucatorului
     * @param armour - Acest parametru este conditia principala in a determina
     * @param alive - daca jucatorul a murit sau nu
     * @param health - Viata curenta a jucatorului (Initializata cu 100)
     * @param nrCards - Numarul de Cards pe care jucatorul le poate utiliza
     */
    public Texture idlesheet;
    private int armour;
    public boolean alive = true;
    private int maxHealth = 100;
    private int health;
    private int nrCards = -1;

    /**
     * @param listaCardsInMana - Toate Cartile pe care jucatorul de tine in mana in
     * @param listaCardsTotal - Toate Cartile detinute de jucator in momentul
     * @param listaCardsInMana - Toate Cartile ce au fost utilizate in combat
     */
    ArrayList<Cards> listaCardsInMana = new ArrayList<>();
    ArrayList<Cards> listaCardsTotal = new ArrayList<>();
    ArrayList<Cards> listaCardsDiscarded = new ArrayList<>();
}

```



```

Java - Eclipse IDE
package com.gdx.game;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.Texture;
import java.util.ArrayList;
import java.util.ListIterator;

/**
 * Player Class.
 */
public class Player {

    /**
     * Parameters.
     *
     * @param idlesheet - idle animation texture.
     * @param armour - amount of armour the player has.
     * @param alive - it tells if the player is alive or not
     * @param health - amount of health the player has.
     * @param nrCards - amount of cards the player can use.
     * @param listaCardsInMana - the hand of cards that the player has.
     * @param listaCardsTotal - all the cards that the player has in deck.
     * @param listaDiscarded - all the cards that have been used.
     */
    public Texture idlesheet;
    private int armour;
    public boolean alive = true;
    private int maxHealth = 100;
    private int health;
    private int nrCards = -1;
    int mana=3;
    ArrayList<Cards> listaCardsInMana = new ArrayList<>();
    ArrayList<Cards> listaCardsTotal = new ArrayList<>();
    ArrayList<Cards> listaCardsDiscarded = new ArrayList<>();
}

```





5

# CONCLUZII ȘI EXTINDERI POSIBILE ÎN VIITOR

Implementarea de animații pentru acțiuni precum atac sau apărare

Crearea și implementarea mai multor tipuri de cărți

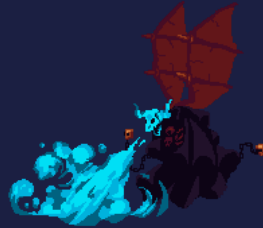
Crearea unui sistem de monștri specifici fiecărei zone.

Abilitatea de a vizualiza cartile ramase in pachet si cartile deja utilizate in lupta

Crearea unui sistem de mana pentru utilizarea abilităților/cărților.



# CONCLUZII



Acest proiect s-a dovedit a fi mai greu decat ne-am așteptat, din cauza timpului limitat pe care l-a avut fiecare membru al echipei.

Suntem convinși că această experiență a contribuit la dezvoltarea lucrului în echipa. Deși nu am reușit să facem tot ce ne-am propus, a fost o experiență interesantă datorită faptului că am reușit să învățăm și să utilizăm o nouă librerie precum LibGDX, cât și dezvoltarea abilităților noastre de a soluționa problemele apărute pe parcurs.

În concluzie, a fost o placere și o provocare de a crea un proiect atât de interesant, iar acest lucru nu ne poate face decat sa fim mândri de realizarea noastră.