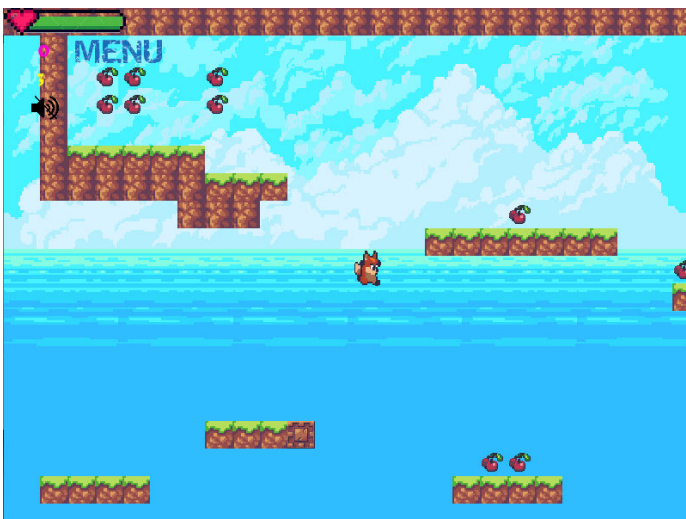
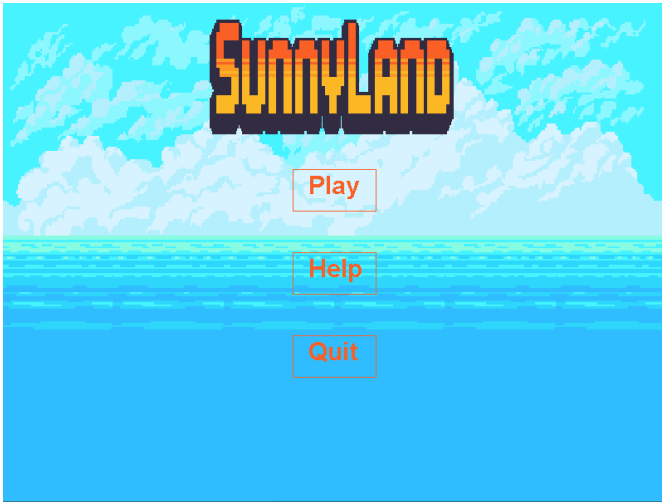


SunnyLand

Pavăl Mihaela-Irina

affiliation: <Faculty of Automatic Control and Computer Engineering, Iasi>,
email: <mihaelapaval35@gmail.com>,
academic year: <2019 - 2020>



- **Gameplay:** Jucătorul trebuie să colecteze toate cireșele pentru a avansa spre următorul nivel. Poți folosi diamantele pentru a te apăra de inamici, iar inimioarele îți vor crește durata de viață. Jocul va fi câștigat atunci când toți inamicii sunt înfrânți.
- **Plot:** Ești o vulpe plecată în căutare de hrană. Scopul tău este de a ajunge în siguranță la puii tăi în timp ce înfrânți toți inamicii întâlniți în drum spre casă.
- **Characters :**
 - i. *Foxie* este protagonistul jocului. Este o vulpiță simpatică ce dorește să ajungă în siguranță acasă la puii săi. Are abilitatea de a se apăra de pericole emițând stele luminoase din mainile sale.



- ii. *Opossum* este inamicul acestui joc.



• **Mecanics :**

- i. **Levels:** jocul are 5 niveluri de dificultate. Un nivel este stocat într-un fișier cu extensia .png a cărui adresă este stocată în baza de date. Nivelul de dificultate este determinat de numărul de inamici și poziționarea dalelor pentru a facilita acesul la cireșe, respective diamante. Fiecare nivel are propria hartă.
- ii. **Game points:** pentru accesarea nivelului următor trebuie colectat un numar de 19 cireșe pentru fiecare nivel. Nu exista posibilitatea trecerii la un nivel superior daca aceasta conditie nu este satisfacută. Scopul jocului este de a omorî toți inamicii de pe parcursul întregului joc.
- iii. **User interaction:** pe parcursul jocului player-ul intră în interacțiune cu entități de tip cireașă, diamant, inamic și block. În momentul în care se realizează coliziunea cu o cireașă/diamant contorul aflat în partea stângă sus crește până la un număr egal cu 19. Acțiunea de atac se realizează doar dacă contorul responsabil de numărarea diamantelor este diferit de 0. Dacă se realizează coliziunea cu un inamic (oposum) viața player-ului scade, acest lucru putând fi vizibil, de asemenea, în colțul din stânga. Utilizatorul are posibilitatea de a reveni în orice moment la meniul principal apăsând butonul de meniu ce este vizibil pe întreaga durată a jocului. De asemenea există și opțiunea de a dezactiva sunetul prin apăsarea butonului disponibil în partea stângă.
- iv. **The control keys:**

Action	Player
Press to move up	UP Arrow
Press to move right	RIGHT Arrow
Press to move left	LEFT Arrow
Press to shoot	SPACE

- **Data base: *data.db*** - în aceasta sunt stocate datele specific fiecărui nivel

Coloana Level – ID-ul nivelului

Coloana Background – adresa fișierului ce conține mapa nivelului

Coloana Sound – adresa fișierului ce conține coloana sonoră a nivelului

	Level	Background	Sound
	Filter	Filter	Filter
1	1	/level/level.png	/sound/platformer_level03_loop....
2	2	/level/level2.png	/sound/platformer_level03_loop....
3	3	/level/level3.png	/sound/platformer_level03_loop....
4	4	/level/level4.png	/sound/platformer_level03_loop....
5	5	/level/level5.png	/sound/platformer_level03_loop....

```

public static String getLevelPath(int level){
    Connection c = null;
    Statement stmt = null;
    String sql = null;
    String path = "";
    try {
        Class.forName("org.sqlite.JDBC");
        c = DriverManager.getConnection("jdbc:sqlite:data.db");
        c.setAutoCommit(false);
        stmt = c.createStatement();
        //      sql = "CREATE TABLE LevelPath " +
        //          "(Level INT, " +
        //          " Background TEXT )";
        //      stmt.executeUpdate(sql);
        //      sql = "INSERT INTO LevelPath (Level, Background) " +
        //          "VALUES (1, '/level/level.png');";
        //      stmt.executeUpdate(sql);

        sql = "ALTER TABLE LevelPath " + " ADD COLUMN Sound TXT; ";
        //      stmt.executeUpdate(sql);
        //
        sql = "INSERT INTO LevelPath (Sound) " +
            "VALUES ('/sound/platformer_level03_loop.wav');";
        //      stmt.executeUpdate(sql);

        ResultSet rs = stmt.executeQuery("SELECT * FROM LevelPath WHERE Level=" + level + " ");
        path = rs.getString("Background");

        rs.close();
        stmt.close();
        c.close();
    } catch (Exception e) {
        System.err.println( e.getClass().getName() + ": " + e.getMessage() );
        System.exit(0);
    }
    System.out.println("Operation done successfully");
    return path;
}

```

```

public static String getAudioPath(int level){
    Connection c = null;
    Statement stmt = null;
    String path = "";
    try {
        Class.forName("org.sqlite.JDBC");
        c = DriverManager.getConnection("jdbc:sqlite:data.db");
        c.setAutoCommit(false);
        stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM LevelPath WHERE Level=" + level + " ");
        path = rs.getString("Sound");
        System.out.println(path);

        rs.close();
        stmt.close();
        c.close();
    } catch (Exception e) {
        System.err.println( e.getClass().getName() + ": " + e.getMessage() );
        System.exit(0);
    }
    System.out.println("Operation done successfully");
    return path;
}

```

- **Diagrama de clase:**

Diagrama de clase este anexată în fișierul SunnyLand.png și în fișierul SunnyLand.uml ce poate fi accesat folosind IntelliJ IDEA Ultimate.