Dinosaur Island
Ioana-Roxana Băcălie
1208A

1. **Povestea jocului**

Jocul este situat pe o insulă necunoscută, pe care a ajuns Cleo după ce i s-a defectat avionul. Pe această insulă se află dinozauri, despre care se credea că sunt animale dispărute. Pentru că insula este plină de pericole, Cleo vrea să scape cât mai repede de acolo. Pentru a pleca de pe insulă, are nevoie să adune anumite unelte căzute în timpul prăbușirii cu care să repare avionul. Trebuie să fie atentă la orice mișcare, căci orice întâlnire cu un dinozaur este fatală. De asemenea, Cloe nu știe să înoate, așa că trebuie să fie atentă să nu ajungă în apă și trebuie să se uite pe unde calcă pentru că sunt gropi împrăștiate pe toată insula.

2. **Prezentare joc**

Jucătorul trebuie să adune toate materialele necesare și să se ferească de dinozauri, pentru a trece la nivelul următor. De asemenea, trebuie să adune anumite unelte. Jocul este de tip single player și este construit pe niveluri.

3. **Regulile jocului**

Jocul implică deplasarea pe insulă pentru a lua obiectele necesare. Jucătorul este ucis dacă este mâncat de un dinozaur, dacă ajunge în apă sau într-o groapă. Trecerea la următorul nivel se face în momentul în care toate obiectele sunt adunate. Peste apă se poate trece doar prin zona unde sunt pietre, altfel personajul moare.

.

Metoda de victorie - fiecare nivel este câștigat atunci când toate obiectele de pe hartă sunt adunate. Scorul reprezintă timpul necesar rezolvării nivelelor (se adună timpii de la nivelele anterioare). Cu cât scorul este mai mic, cu atât este mai bine, fiindcă înseamnă că nivelurile au fost rezolvate mai rapid. Scorul este trecut în milisecunde pentru o precizie mai mare.

Metoda de înfrângere - personajul este înfrânt atunci când cade într-o groapă, cade în apă sau când este găsit de oricare din dinozauri. Personajul are o singură viață. Dacă personajul este învins într-un nivel superior, atunci când moare se întoarce înapoi la nivelul 1 (nu este salvat progresul)..

Mutări imposibile - acolo unde există coliziuni cu un copac/buștean la stânga, personajul nu se poate muta la stânga. La fel se întâmplă și la celelalte direcții. De asemenea, personajul nu poate trece de marginile hărții.

4. **Personajele jocului**

Cloe este personajul principal al jocului, fiind o fată curajoasă și determinată. Știe să piloteze avioane, însă nu știe să înoate, așa că e foarte periculos să ajungă în preajma apei. Cloe se ferește de dinozaurii de pe insulă, pentru că dacă o întâlnesc în cale, aceștia o vor considera hrană.

Dinozaurii de pe insulă, deși par prietenoși, sunt foarte periculoși, iar Cloe trebuie să se ferească de ei.

5. **Tabla de joc**

Componente pasive

- Iarba, florile, pietrele, pânza de păianjen - nu au proprietăți speciale
- Copacii, bușteni - personajul trebuie să le ocolească, cresc dificultatea jocului
- Pietrele de pe apă - personajul poate trece apa pe pietre fără a muri

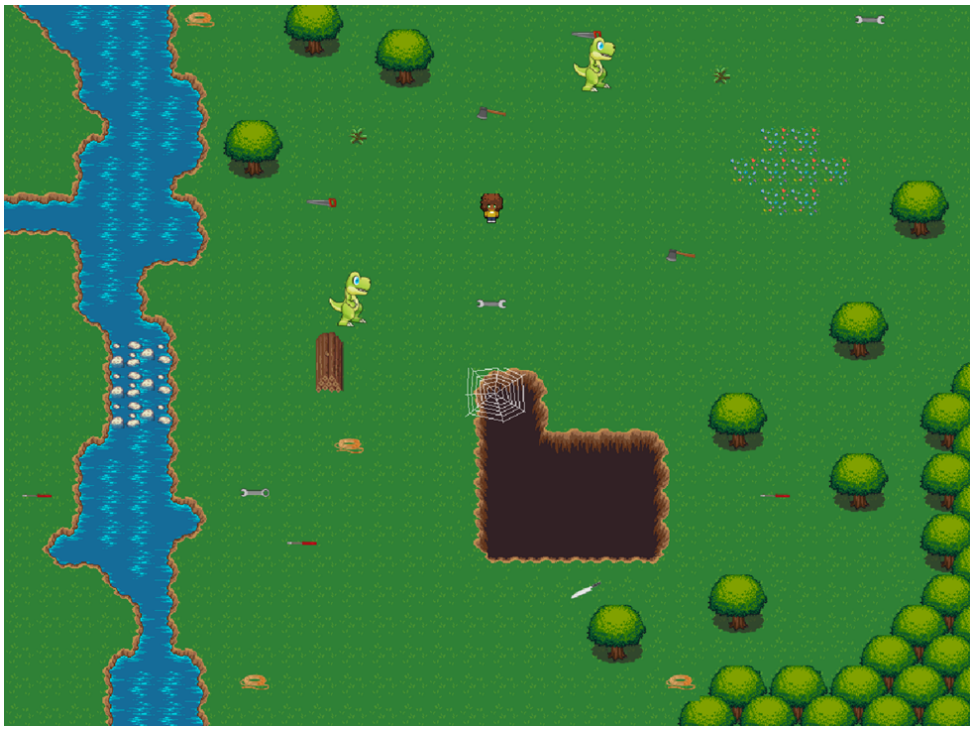
Componente active

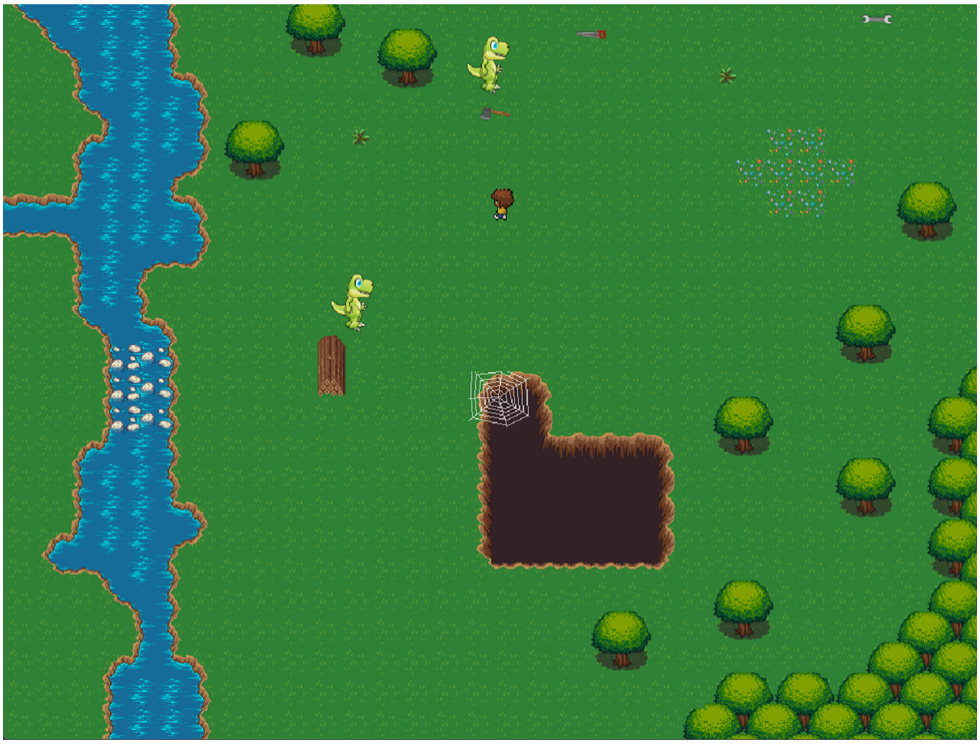
- Apa - dacă ajunge în apă, personajul moare
- Gropile - dacă ajunge în groapă, personajul moare
- Uneltele - când ajunge la acestea, personajul le adună - când sunt toate obiectele adunate, personajul trece la următorul nivel

Structura tablei de joc și modul în care este construită

- Dimensiunea tablei de joc este de **1024 x 768 pixeli**, fiind alcătuită din dale de dimensiunea **32 x 32 pixeli**.
- Sunt 3 niveluri, iar tablele diferă: la nivelul 1 sunt un râu și o groapă, dar la următoarele niveluri apar mai multe gropi/lacuri care cresc dificultatea pentru că personajul principal trebuie să le ocolească. De asemenea, crește numărul de dinozauri odată cu creșterea nivelului. Mai jos sunt reprezentate cele 3 table de joc pentru cele 3 niveluri (imaginile sunt screenshot-uri din joc):

6. **Screenshot-uri din timpul jocului**
    - **Nivelul 1 -** screenshot din timpul jocului, în două ipostaze - când niciun obiect nu este adunat și când majoritatea obiectelor sunt deja adunate de pe hartă





- **Nivelul 2**

- **Nivelul 3**

7. **Mecanica jocului**

- Personajul este controlat folosind săgețile sus / jos / stânga / dreapta.
- Săgeată stânga - personajul se mișcă la stânga
- Săgeată dreapta - personajul se mișcă la dreapta
- Săgeată sus - personajul se mișcă în sus
- Săgeată jos - personajul se mișcă în jos
- Tasta Space - personajul adună obiectul din proximitatea sa
- Meniul este controlat cu ajutorul mouse-ului și atunci când se apasă click stânga pe un buton acesta este activat.

8. **Game Sprite**

9. **Descriere fiecare nivel**

Dificultatea este crescută treptat în fiecare nivel, deoarece apar mai mulți dinozauri și sunt mai multe obiecte de adunat. De asemenea, după cum este explicat și mai sus, harta jocului este din ce în ce mai complicată, cu mai multe obstacole (apă, gropi) la fiecare nivel.

- **Nivelul 1 -** sunt 2 dinozauri pe insulă de care trebuie să se ferească Cloe și 15 unelte de adunat
- **Nivelul 2 -** sunt 5 dinozauri pe insulă de care trebuie să se ferească Cloe și 20 unelte de adunat
- **Nivelul 3 -** sunt 9 dinozauri pe insulă de care trebuie să se ferească Cloe și 30 de unelte de adunat

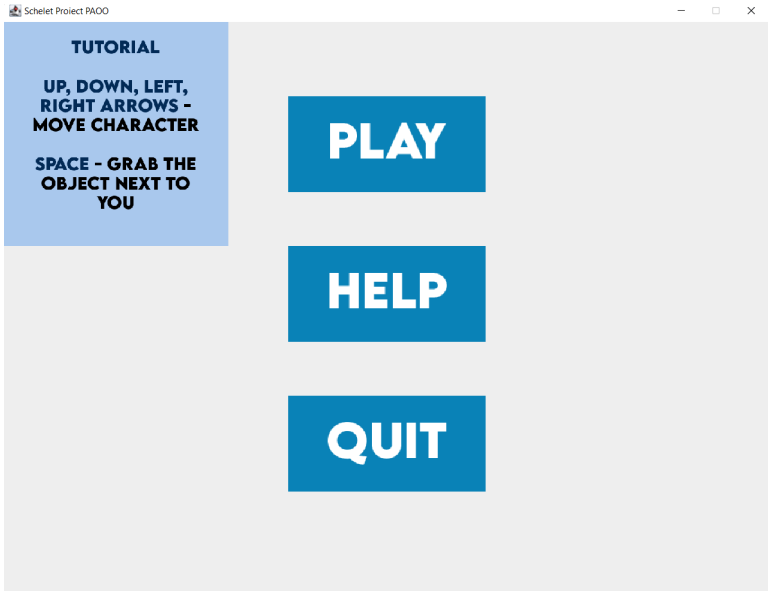
10. **Descrierea meniului jocului**

Meniul este alcătuit din următoarele butoane:

- **Buton Play** - prin apăsarea acestuia are loc începerea jocului
- **Buton Help** - prin apăsarea acestuia sunt arătate instrucțiuni despre ce trebuie făcut în joc și ce taste trebuie apăsate
- **Buton Quit** - prin apăsarea acestuia are loc ieșirea din fereastra de joc

Așa arată meniul:

Așa arată meniul după ce este apăsat butonul HELP.



Când cursorul este deasupra unui buton acesta își schimbă culoarea în felul următor:

11. Șabloane de proiectare - SINGLETON și OBSERVER

- Utilizarea șablonului de proiectare **Observer**
    - Am creat două interfețe - Observer și Subject
    - Clasa Score implementează interfața Observer și conține o metodă update(int level, long score), care schimbă scorul și nivelul și le introduce în baza de date atunci când un nivel este câștigat.
    - Clasa GameState implementează interfața Subject, iar atunci când un nivel este câștigat, observerul (Score) este notificat prin metoda notifyObservers(), care apelează metoda update din clasa Score.
- Utilizarea șablonului de proiectare **Singleton**
    - Pentru a restricționa numărul de instanțe al clasei MenuState la un singur obiect am folosit șablonul de proiectare Singleton.
    - În clasa MenuState este implementată metoda getInstance(), ce permite crearea unei noi instanțe a clasei dacă aceasta nu există deja. Daca instanța exista deja, atunci întoarce o referință către acel obiect.

12. Diagrame de clase

În imaginea următoare poate fi vizualizată diagrama de clase:

Imaginea cu diagrama de clase poate fi descărcată la link-ul acesta pentru o vizualizare mai ușoară. Se poate da zoom, pentru a observa fiecare clasă în parte.

Descrierea claselor:

- **Main**
    - După cum se poate observa din diagrama prezentată mai sus, această clasă creează o instanță de tipul Game, iar mai apoi pornește fluxul activităților prin apelul metodei StartGame() din Game.
- **Game**
    - Rolul acestei clase este de a menține interfața grafică la zi cu ceea ce se întâmplă în joc.
    - Această clasă implementează interfața Runnable pentru a avea comportamentul unui fir de execuție (thread).
    - În momentul apelului metodei **StartGame()** se instanțiază un obiect de tip Thread pe baza instanței curente a clasei Game. Orice obiect de tip Thread trebuie să implementeze metoda run() care este apelată atunci când firul de execuție este pornit (start()). Această metodă **run()** inițializează jocul, iar mai apoi controlează numărul de cadre pe secundă printr-o buclă while și “pregătește” noua scenă (Update()) pe care o va desena pe interfața grafică (Draw()).

    - Metoda **Update()** actualizează starea jocului (de exemplu: modifică poziția jucătorilor pe baza tastelor apăsate, schimbă poziția inamicilor, crează diferite tile-uri (dale), etc. Se apelează metoda Update() din clasa GameState/MenuState, care la rândul său apelează alte metode Update(). Metoda Draw() va desena pe interfața grafică modificările făcute de metoda Update().
    - Metoda **run()** a clasei Game crează o instanță a clasei GameWindow. Această clasă este responsabilă cu fereastra în care vor fi desenate obiectele pe un canvas.. De asemenea, avem un obiect JFrame care permite desenarea de butoane, controale, textbox-uri etc.
- **Tile**
    - Această clasă reține informații despre tile-urile (dalele) din joc. Clasa Tile reține câte o instanță pentru fiecare sub-tip de tile (GrassTile, MontainTile, TreeBottomTile, RoundTreeLeftTile, WaterTile etc.). Toate aceste clase extind clasa Tile. Constructorul acestora primește ca parametru un ID prin care se va putea identifica acel tile și apelează constructorul clasei de bază, adică al clasei Tile, care primește ca parametru un membru static al clasei Assets. Membrul static folosit este ales în funcție de tipul clasei respective, de exemplu pentru GrassTile se va folosi membrul static grass.
- **Assets**
    - Conține câte un membru static de tip BufferedImage pentru fiecare tile/dală și pentru jucători/inamici. In acești membri sunt stocate imaginile, adică texturile acestora care vor fi desenate prin apelarea metodelor Draw() din fiecare clasă. Pentru a decupa imaginile se folosește o instanță a clasei SpriteSheet în metoda Init(). Metoda folosită din SpriteSheet este crop(x, y) sau crop2(x,y), crop3(x,y) în funcție de cum dorim să fie decupată imaginea.
- **SpriteSheet**
    - Constructorul acestei clase primește imaginea. Metoda crop(x, y) (sau crop2, crop3, crop4) primeste doi parametrii, x specifică pe ce coloană se află textura pe care dorim să o decupăm în imaginea cu toate texturile, iar y specifică linia.
- **ImageLoader**
    - Înainte de a extrage fiecare textură, imaginea cu toate texturile trebuie să fie citită din memorie, iar pentru asta se folosește clasa ImageLoader cu metoda statică LoadImage(path) care primește ca parametru calea către imagine în memoria calculatorului.
- **EntityManager**
    - Conține un ArrayList denumit entities, în care se va adăuga fiecare entitate mobilă (dinozauri, player și nu entitățile statice precum Tool1, Rope etc.) de pe tablă.
    - Constructorul primește un obiect de tip Player, care se va adăuga în entities și va fi primul element (elementul cu indice 0).

    - Metodele Update() și Draw() apelează metodele Update() și, respectiv, Draw() din obiectele de tip Entity.
    - Metoda addEntity adaugă o entitate în entities, iar metoda getEntities returnează entities.
- **Entity**
    - Este o clasă abstractă ce va fi extinsă de clasele Player și Dinosaur.
    - Metoda getCollisionBounds() este utilizată pentru coliziuni, returnează un dreptunghi (Rectangle) ce reprezintă zona în care este mărginit personajul.
    - Metoda checkEntityCollision(Entity) va verifica dacă există coliziune cu o altă entitate
    - Metodele CollisionWithWater(int x, int y) și CollisionWithDirt(int x, int y) verifică dacă există coliziuni cu apa sau groapa (verifică dacă entitatea se află pe un tile de tip apă sau groapă, prin apelarea metodei getTile(x,y) din Game.
- **Player**
    - Extinde clasa Entity
    - Pentru animații folosim instanțele animUp, animDown, animLeft, animRight ale clasei Animation
    - Metoda **move()** apelează metodele **moveX**() și **moveY**() care verifică dacă există coliziuni cu obstacole precum copacii și buștenii și dacă personajul se poate deplasa în respectiva direcție sau nu.
    - Metoda CollisionWithTile(int x,int y) este apelată în moveX() și moveY() pentru verificarea coliziunii cu dalele de tip solid (precum copacii, buștenii). Pentru această verificare, există o metodă isSolid() în clasa Tile, care arată dacă tile-ul e solid sau nu.
    - În metoda **Update()** se verifică ce tastă este apăsată din săgețile sus/jos/stânga/dreapta și se modifică variabilele xMove și yMove.
    - Metoda **getCurrentAnimationFrame()** verifică dacă personajul se deplasează la stânga, la dreapta, sus, sau jos și returnează animația potrivită.
- **Dinosaur**
    - Extinde clasa Entity
    - Similar cu metoda din clasa Player, are o metoda **getCurrentAnimationFrame**(), dar doar pentru mișcare stânga/dreapta.
    - Metoda Update() realizează mișcarea între două coordonate ale lui x diferite, în funcție de variabila type a dinozaurului, care este inițializată atunci când este creată o instanță Dinozaur.
- **StaticEntity**
    - extinde la rândul său clasa Entity și este extinsă de clasele Knife, Tool1, Tool2, Tool3, Tool4, Tool5, Rope, folosite pentru crearea uneltelor de pe hartă.
- **Animation**
    - Clasă folosită pentru animații

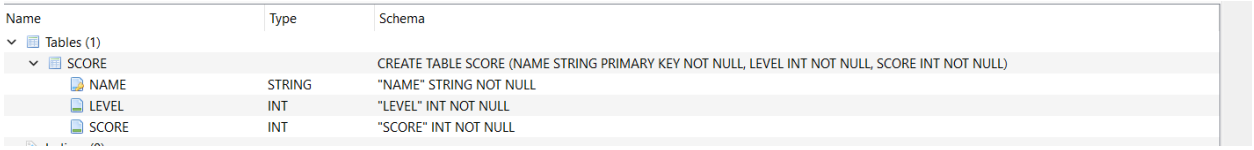
    - Conține un vector de tip BufferedImage (frames) iar prin metoda getCurrentFrame se returnează imaginea corespunzătoare momentului din animație în care se află personajul (adică se returnează frames[index], iar indexul este modificat în functia update în funcție de trecerea milisecundelor).
- **MenuState**
    - În constructor se apelează metoda addObject(UIObject) din UIManager pentru a se crea cele 3 butoane (Play, Quit, Help)
    - Funcția onClick() de la fiecare buton este diferită - atunci când se apăsa pe butonul Play începe jocul (State.setState(game.getGameState()), atunci când se apasă pe butonul Help apare/dispare imaginea cu instrucțiunile (se modifică variabila help care dacă este true atunci în metoda Draw este desenată și imaginea cu instrucțiunile, iar în caz contrar nu este desenată) , iar atunci când se apasă pe butonul Quit se iese din fereastra de joc (Runtime.getRuntime.exit(0))
- **GameState**
    - În level este stocat un întreg care reprezintă nivelul la care se află jucătorul
    - În score este stocat scorul care reprezintă timpul (calculat in milisecunde) de la începerea nivelului 1 până la câștigarea unui nivel
    - Este folosit șablonul de proiectare Observer, deoarece GameState implementează interfața Subiect și anunță observerii când scorul este modificat. Clasa Score implementează interfața Observer.
    - Metoda CreateEntities() crează entitățile corespunzătoare fiecărui nivel al jocului (se apelează entityManager.addEntity).
    - Metoda Update() apelează CreateEntities() și entityManager.update(), iar apoi se verifică dacă se apasă tasta space atunci când player-ul e în preajma unui obiect. Dacă da, atunci obiectul este scos de pe hartă prin apelarea metodei remove. După, se verifică dacă există coliziuni cu dinozaurii sau cu dalele de tip apă sau groapă și, dacă da, atunci player-ul a pierdut jocul și se ajunge din nou la Meniu. Dacă toate obiectele au fost adunate (adică dimensiunea entities din entityManager e egală cu numărul dinozaurilor + 1), atunci jocul este câștigat și se trece la nivelul următor, după ce sunt anunțați Observerii (Score).
    - Metoda Draw() - în funcție de numerele de pe un tablou bidimensional map declarat in clasa Map, se desenează Tile-urile potrivite pe hartă. Se apelează metoda getMap(int level) din clasa Map pentru a vedea care hartă este folosită (hărțile diferă la fiecare nivel).
- **Score**
    - Implementează interfața Observer și conține o metodă update(int level, int score), care inserează scorurile în baza de date. Metoda este apelată în metoda notifyObservers() din clasa GameState.

- **MouseManager, UIImageButton, UIManager, UIObject**
    - Sunt clase utilizate pentru realizarea butoanelor

13. Strategia de joc - Jocul este de tip single player.

14. Baza de date

- Baza de date conține 3 coloane (NAME, LEVEL, SCORE)
- **Coloana NAME** - conține numele jucătorului, introdus de acesta când câștigă nivelul - este cheia primară
- **Coloana LEVEL** - conține nivelul câștigat de fiecare jucător
- **Coloana SCORE** - conține scorul fiecărui jucător care reprezintă timpul, în milisecunde, de la începerea nivelului 1 până la terminarea unui nivel. Un scor mai mic indică un timp mai bun și astfel jucătorii cu timp mai mic sunt cei care s-au descurcat mai bine
- Sunt adăugate/updatate elemente în baza de date atunci când se câștigă un nivel.



Exemplu de cum arată baza de date după ce mai mulți jucători au caștigat niveluri.

	NAME	LEVEL	SCORE
	Filter	Filter	Filter
1	Paula	1	3455
2	Rebeca	3	273693
3	Alina	3	80360
4	Maria	1	5950
5	Ana	1	10131
6	Marian	1	8044
7	Flavia	3	435364
8	Casiana	2	245666
9	Camila	1	53693

15. Bibliografie

- https://opengameart.org/content/lpc-forest-tiles
- https://www.gameart2d.com/free-dino-sprites.html
- https://www.freepik.com/free-vector/toolbox-many-tools-illustration_1142108.htm#query=tools%20drawing&position=4&from_view=search
- https://www.pngall.com/blade-png/download/63678
- https://opengameart.org/content/top-down-player-sprite-sheet-julia

- **https://www.freepik.com/free-vector/set-isolated-gardening-tools_6830092.htm#query=rope%20drawing&position=10&from_view=search**