Proiect PAOO
Nume: Baschir Calin
Grupa: 1209B

Trapped

Povestea jocului:

Legenda spune ca după ascensiunea regelui Minos la tronul insulei Creta, acesta se ruga la zeul marii, Poseidon, să-i trimită un taur alb ca zapada, ca semn al acceptarii divine. După ce i-a fost indeplinita dorinta, deși regele trebuia să sacrifice taurul, datorită frumuseții sale, Minos a dorit sa il pastreze. Conducatorul credea ca Poseidon avea sa accepte si un alt animal drept sacrificiu. Ca pedeapsa zeul a facut pe sotia regelui sa se indragostească de taur. A aparut, astfel, Minotaurul (Taurul lui Mino). Poetul Roman Ovid descrie aceasta creatura ca fiind “parte om, parte taur”. Pe masura ce a crescut Minotaurul nu mai putea fi controlat și devenea din ce în ce mai agresiv. În urma sfatului oracolului din Delphi, Minos i-a angajat pe Daedalus și pe fiul sau, Icar, să construiască un labirint subteran din care creatura sa nu poată evada cu usurinta.

Incercand, totusi, sa scape, Minotaurul se lupta cu tot felul de alte concepții ale imaginatiei, dar cea mai mare dificultate care i se impune este chiar propriul temperament. Dacă este atins de prea mulți inamici sau capcane, protagonistul își pierde controlul, și poate provoca un cutremur devastator pentru întreaga insula, blocandu-si, totodata, și singura posibilitate de scăpare.

Regulile Jocului:

Player-ul controlează Minotaurul în încercarea lui de a găsi ieșirea către suprafață. Daca acesta va fi atins de alti minotauri va provoca un cutremur ce va face intregul labirint sa se prabusească. Trebuie sa se fereasca pe cat posibil de inamici.

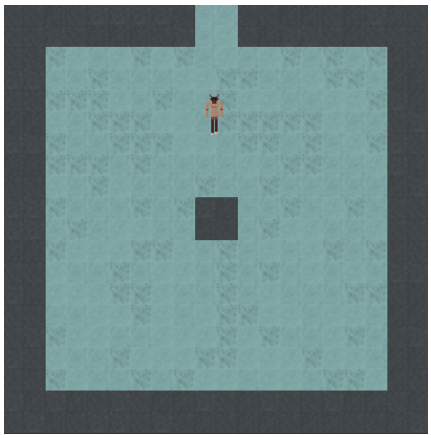
Caractere:

Minotaurul (controlat de player) și inamicii acestuia.

Meniul jocului:

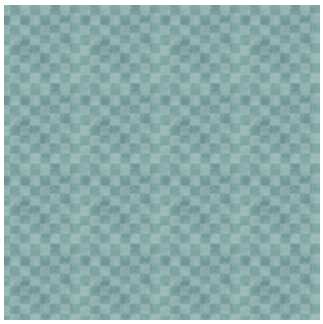
Meniul jocului este compus doar dintr-un buton de play, care va da drumul unei noi sesiuni din joc.

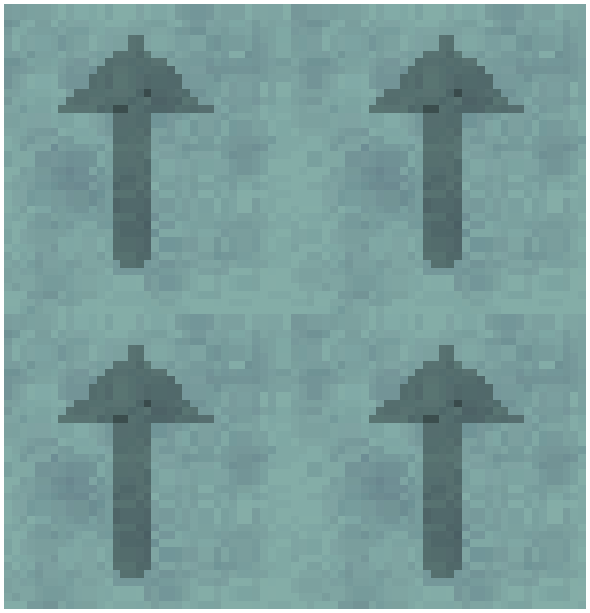
Exemplu de camera aparținând primului nivel:



Mecanica jocului:

Scopul jucătorului este cel de a duce caracterul în ultima camera a jocului, unde se afla tile-urile de sfarsit.





Bineinteles, in drumul sau catre victorie, playerul trebuie sa se ferească de alti minotauri care il urmaresc, folosindu-se de propria sa agilitate, de teren, dar si ajutandu-se de upgrade-urile de viteza oferite de tile-urile marcate ca atare.

Tabla de joc:

- Componente pasive
    - Podea
    - Zid
- Componente active
    - Tile de power up
    - Tile de finish

Jucătorul este **invins** în momentul în care un inamic il atinge. Pe de cealalta parte, protagonistul poate castiga în momentul în care atinge tile-urile specifice.

Structura și funcționalitatea claselor:

- **Main**
    - Se inițializează atat fereastra jocului, cat și jocul în sine

- **GamePanel**

- Este clasa în care se conecteaza toate elementele jocului precum **TileManager**, **MouseHandler** (Singleton), **KeyHandler**, **Player**, **Collison Checker**, **LevelManager**, **Menu**, **Thread**
- Public void run()
  - Controlează numărul de update-uri care sa se dea (60 pe secunda)
- Public void update()
  - Ajută la incrementarea scorului, o data cu trecerea secundelor
  - Da update la player si la level
- Public void paintComponent()
  - Stabilește în funcție de stadiul jocului ce trebuie desenat pe ecran

- **TileManager**
  - Contine un vector de Tile-uri, caracterizate de imaginea pe care o au, si trei valori de tip boolean, una pentru victorie, una pentru coliziune și una pentru speedUp
  - O matrice de întregi în care se reține ce fel de Tile este pe fiecare poziție
  - Public void getTileImage()
    - Încarcă imaginea pentru toate tile-urile utilizate
  - Public void loadMap(Graphics, String)
    - Depinzand de un fișier text în care se afla o matrice 20x20 se încarcă în fiecare “coordonata” tile-ul corespunzător. De asemenea în acest fișier se mai afla și numărul de mobi pozitia de spawn și viteza acestora
  - Funcțiile checkSpeedUp & checkVictory verifica dacă tile ul de sub player are vreo caracteristică specială
- **Player**

  - Functiile precum resurrect, isDead, setDead, setSpawnPoint, getPlayerImage,update, draw fac ceea ce spune și numele
  - checkTileAttributes verifica tile urile cu autorul funcțiilor checkSpeedUp & checkVictory mai sus prezentate
  - Derivata din clasa **Entity**
- **CollisionChecker**
  - În funcție de orientarea playerului verifica dacă următorul tile în care urmează sa se afle are sau nu coliziune. Dacă are îl va arunca înapoi în spate pentru a nu ajunge sa se blocheze.

- **LevelManager**
  - Cu ajutorul unui contor a nivelului curent, aceasta functie încărca fișierul text potrivit. De asemenea da update la Mobi, printeaza scorul cand se ajunge în ultima camera și verificam constant daca trebuie sa se treacă la un nivel precedent sau următor.
- **Menu**
  - În aceasta clasa este meniul și cuprinde încărcarea lui pe ecran și butonul de play.

Bibliografie:

Pana la coliziuni m-am inspirat destul de puternic din tutorialul urmator:

How to Make a 2D Game in Java #1 - The Mechanism of 2D Games