Házi feladat - 2048

Programozás alapjai 3.

Specifikáció, dokumentáció, felhasználói kézkönyv

Kerek Dominik
ZZSE8C

2021. 12. 05.

Tartalom

1.Feladat

Az ismert 2048 játék implementálása lesz a feladat. Ebben a játékban a 2 hatványainak összeadásával kell elérni a 2048-as mezőt. A játéktér egy 4x4-es felület. A nyilak segítségével mozgathatjuk a mezőket, és ha egyformák, akkor azok összeadódnak. Minden mező pontosan addig csúszik az adott irányba, amíg falat, vagy egy másik különböző mezőt nem ér, egyébként összeadódnak és lesz egy nagyobb értékű mező. Az új mezők folyamatosan jelennek meg random a pályán, az üres játékterületen, minden lépés után.

2.Feladatspecifikáció / use-case-ek

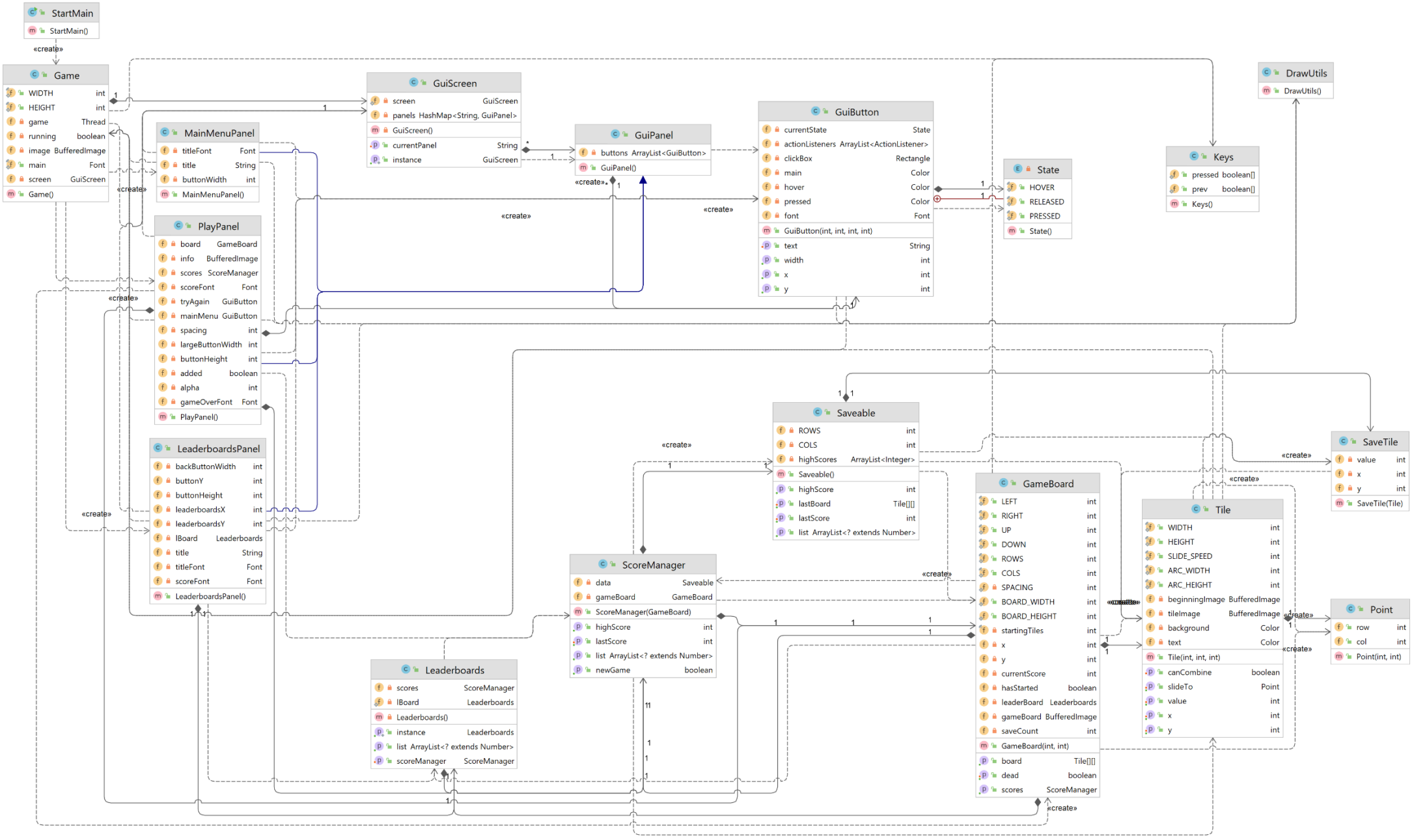
A játékos egy grafikus felülettel találkozik. Indulást követően egy menü fogadja a felhasználót, itt 3 menüpont lesz elérhető:

- *Play*
- *Scores*
- *Quit*

Play: Ha van egy félbe maradt játék, akkor azt tudjuk folytatni, ha nincsen, akkor egy teljesen új játék indul. Amennyiben a játéktér megtelik, egy Gameover feliratot látunk, ekkor választhatunk, hogy egy új játékot indítunk vagy kilépünk a menübe. A pontszámunkat és a legjobb elért pontszámot látjuk a játék közben.

Scores: Itt látható a legjobb elért pontszámok sorrendben kiírva.

Quit: Kilép a játékból.


StartMain
StartMain()
«create»
Game
WIDTH int
HEIGHT int
game Thread
running boolean
image BufferedImage
main Font
screen GuiScreen
Game()
GuiScreen
screen GuiScreen
panels HashMap<String, GuiPanel>
GuiScreen()
currentPanel String
instance GuiScreen
GuiPanel
buttons ArrayList<GuiButton>
GuiPanel()
GuiButton
currentState State
actionListeners ArrayList<ActionListener>
clickBox Rectangle
main Color
hover Color
pressed Color
font Font
GuiButton(int, int, int, int)
text String
width int
x int
y int
State
HOVER
RELEASED
PRESSED
State()
DrawUtils
DrawUtils()
Keys
pressed boolean[]
prev boolean[]
Keys()
MainMenuPanel
titleFont Font
title String
buttonWidth int
MainMenuPanel()
PlayPanel
board GameBoard
info BufferedImage
scores ScoreManager
scoreFont Font
tryAgain GuiButton
mainMenu GuiButton
spacing int
largeButtonWidth int
buttonHeight int
added boolean
alpha int
gameOverFont Font
PlayPanel()
LeaderboardsPanel
backButtonWidth int
buttonY int
buttonHeight int
leaderboardsX int
leaderboardsY int
lBoard Leaderboards
title String
titleFont Font
scoreFont Font
LeaderboardsPanel()
Saveable
ROWS int
COLS int
highScores ArrayList<Integer>
Saveable()
highScore int
lastBoard Tile[][]
lastScore int
list ArrayList<? extends Number>
SaveTile
value int
x int
y int
SaveTile(Tile)
GameBoard
LEFT int
RIGHT int
UP int
DOWN int
ROWS int
COLS int
SPACING int
BOARD_WIDTH int
BOARD_HEIGHT int
startingTiles int
x int
y int
currentScore int
hasStarted boolean
leaderBoard Leaderboards
gameBoard BufferedImage
saveCount int
GameBoard(int, int)
board Tile[][]
dead boolean
scores ScoreManager
Tile
WIDTH int
HEIGHT int
SLIDE_SPEED int
ARC_WIDTH int
ARC_HEIGHT int
beginningImage BufferedImage
tileImage BufferedImage
background Color
text Color
Tile(int, int, int)
canCombine boolean
slideTo Point
value int
x int
y int
Point
row int
col int
Point(int, int)
ScoreManager
data Saveable
gameBoard GameBoard
ScoreManager(GameBoard)
highScore int
lastScore int
list ArrayList<? extends Number>
newGame boolean
Leaderboards
scores ScoreManager
lBoard Leaderboards
Leaderboards()
instance Leaderboards
list ArrayList<? extends Number>
scoreManager ScoreManager
«create»

3.Osztályok

StartMain: Ez tartalmazza a main-t, példányosít egy Game-t és létrehozza az ablakot. Majd elindítja a játékot.

control package:

Keys: A billentyűleütések kezeléséért felelős.

gameCore package:

Game: Ez a class tartalmazza a game loop-ot, egy külön szálon fut. A JPanel leszármazottja, itt jönnek létre a különböző menüpontok példányai (Menu, Play, Leaderboards). Ezenfelül implementálja a KeyListenert, MouseListenert és a MouseMotionListenert, ezek megvalósításáért is felel.

run(): Ez a függvény fut le a szál elindításakor. Ez a gameloop, ami a *render()* és a *update()* függvényeket másodpercenként 60-szor hívja meg.

start(): Szálbiztos, ha már fut a szál, visszatér, ha még nem létrehoz egy újat és elindítja azt.

render(): Kirajzolja a fehér hátteret, majd meghívja a GuiScreen *render()* függvényét, ami az aktuális panelt rajzolja ki.

update(): Meghívja a GuiScreen és a Keys *update()* függvényét.

GameBoard: Ez az osztály több mindenért is felel. Elsősorban a játéktér kirajzolásáért és a csempék mozgatásáért. De itt számolja a játék az aktuális pontszámot, amit továbbít a ScoreManagernek.

GameBoard(): A konstruktorban létrejön a tábla képe, kirajzolódik, majd ha mentés nélkül indult el a játék, akkor létrehoz egy úgy játékot, ha volt mentés, akkor visszatölti a régit.

setDead(): Ha véget ért a játék, akkor megnézi, hogy az aktuális pontszám nagyobb-e az előző rekordnál, ha igen, akkor beállítja új rekordnak, ezen kívül beállítja, hogy a játéknak vége.

reset(): Alaphelyzetbe állítja a játékterületet, az aktuális pontot lenullázza és elindítja a játékot újból.

start(): Új játék esetén 2 új csempét generál random helyre.

createBoardImage(): Kirajzolja a játékteret, egy 4x4-es tábla.

resetPosition(): A csempék mozgatásakor használatos, új koordinátákat a csempéknek.

move(): A mozgások lehetőségét vizsgálja meg minden csempénél, amint megtelik a játéktér és nincs több mozgási lehetőségünk, egy *false* jelzést ad vissza.

moveTiles(): A csempék újrarendezéséért felelős, irányok alapján amíg olyan jelzést kapunk a fentebb említett *move()* függvénytől, hogy tudunk mozdulni, addig mozdulnak a csempék és ezzel egy időben újak is generálódnak random helyen. Lépésenként menti az állást program.

spawnRandom(): Generál egy véletlenszerű helyen egy új csempét 2 vagy 4 értékkel. Addig keresi az új koordinátákat, amíg üres helyet nem talál, és oda generál egy újat.

checkOutOfBounds(): Eldönti, hogy elérte-e már a játéktér határát az adott irányba mozduló csempe.

checkSurroundingTiles(): True-t ad vissza, ha az adott csempe mellett üres hely van, vagy vele azonos értékű csempe. Mindkét eset azt jelenti, hogy tud arrébb csúszni az adott csempe. False-t ad ha elérte a játéktér határát, vagy egy olyan csempét, aminek az értéke különböző.

checkKeys(): Az irányokat ellenőrzi és állítja be. Ha még nem kezdődött el a játék, akkor átállítja.

checkDead(): Megvizsgálja, hogy van-e még lépési lehetőségünk a *checkSurroundingTiles()* segítségével, ha nincs true-t ad vissza, ami a játék végét jelenti.

update(): Ellenőrzi a billentyű leütéseket és frissíti a csempék helyét a *checkKeys()* és *resetPosition()* függvényekkel.

Leaderboards: A GameBoardtól átvett ScoreManagerből nyert highscore adatokat kezeli. A LeaderBoardsPanel miatt van rá szükség.

ScoreManager: A pontok és rekordok kezelését oldja meg. A fájlkezelés és az aktuális játék közötti kapcsolatot teremti meg.

ScoreManager(): Beállítja a GameBoardot és a Saveable-t.

saveLastState(): Odaadja mentésre a Saveable-nek a GameBoard aktuális állását.

loadLastState(): Betölti a Saveable-ből a legutolsó mentett állást.

isNewGame(): Megvizsgálja, hogy létezik-e mentett fájl, ha nem akkor true-t ad vissza.

saveGame(): Meghívja a Saveable mentés függvényét, menti a játékot fájlba.

getList(): Visszaadja a rekordok mentet listáját.

Tile: Egy csempe, aminek értékétől függően változik a színe. Van koordinátája, ami alapján kirajzolódik.

convertTileToSaveTile(): Egy kapott Tile típusú csempét átalakít egy SaveTile-ra, ami egy fájlba menthető verziója a Tile-nak, csak a lényeges informicókat tartalmazza.

drawImage(): Kirajzolja a csempe értékétől függő színnel az adott csempét.

gui package:

GuiButton: A menüben található gombok, amik érzékenyek az egérmozgásra is, és ilyenkor más színűek lesznek.

GuiPanel: A JPanel leszármazottja, GuiButton-okat tárol és jelenít meg.

GuiScreen: Nevekhez rendelve paneleket lehet állítgatni, hogy éppen melyik az aktuális panel, amit ki kell rajzolni és amikre éppen aktuálisan működnie kell a vezérlésnek.

LeaderBoardsPanel: Kirajzolja a highScore listát sorba rendezve egy Leaderboards cím alá.

drawLeaderboards(): Kirajzolja a mentett rekordok listáját egymás alá.

convertToStrings(): A kapott int típusú listát átalakítja String típusúvá.

MainMenuPanel: Három gombot tartalmaz, Play, Score és Quit GuiButton-ök, ezeket megnyomva kirajzolódik a kívánt panel.

MainMenuPanel(): Hozzáadja a gombokat és ahhoz a listenereket lambda segítségével.

PlayPanel: Ez a class felelős a játéktér folyamatos kirajzolásáért, amit a GameBoard segítségével valósít meg. Ezenfelül, ha a játék véget ért, mert betelt a szabad terület, ez rajzolja ki a GameOver menüt, ahol választhatunk, hogy új játékot kezdünk, vagy visszalépünk a menübe.

drawGui(): Kirajzolja az aktuális pontszámot és a legjobb pontszámot.

drawGameOver(): Ha a játék véget ért kirajzolja a Game Over feliratot és 2 gombot, amiket megnyomva vagy egy új játékot kezdhetünk vagy visszatérhetünk a menübe.

other package:

Point: Egy segédosztály, ami egy sort és egy oszlopot tárol. A csempék átmozgatásánál segít.

DrawUtils: Egy segédosztály, amely 2 funkciót tartalmaz, az egyik a szélességet a másik a magasságot adja vissza, a megadott szöveg hossza alapján. A szövegek kirajzolásánál hasznos.

getMessageWidth(): Visszaadja egy megadott szöveg szélességét.

getMessageHeight(): Visszaadja egy megadott szöveg magasságát.

Saveable: Ez az az osztály, ami mentésre kerül, ezért implementálja a Serializable-t. Mentésre kerül a játéktér legutolsó állása, tehát a csempék, az ehhez az álláshoz tartozó pontszám és a rekordok listája.

getLastBoard(): Visszaadja az utolsó mentett játék állást, egy két dimenziós csempék tömbjét, miután a *ConvertSaveTileToTile()* függvény segítségével visszalakította kirajzolható, normál Tile típussá a mentett csempéket.

setLastBoard(): Beállítja az aktuális játékállást a Tile tipúsú tömböt SaveTile típusúvá alakítva.

saveGame(): Az egész osztályt kimenti egy fájlba.

loadGame(): A kimentett fájl visszaolvassa és visszatér a betöltött Saveable-el.

addNewHighScore(): Az új rekordot beálltja a lista elejére.

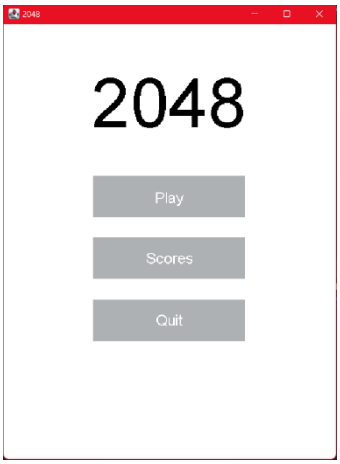
getList(): Visszaadja a rekordok listáját.

SaveTile: Egy segédosztály, ami a csempék mentését segíti, csak a lényeges információkat tartalmazza egy-egy csempéről, amik alapján vissza lehet állítani a játékot oda, ahol abba hagytuk.

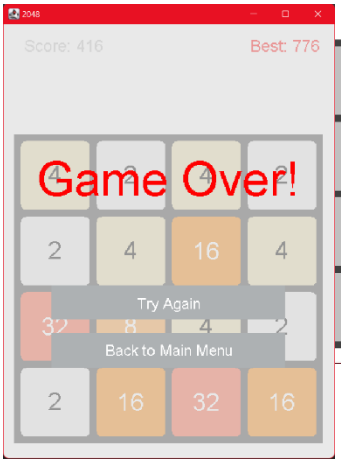
convertSaveTileToTile(): Átalakít egy SaveTile-t egy Tile tipusúvá, ezzel tér vissza.

4. Felhasználói kézikönyv

A programot elindítva a felhasználó a menüvel találkozik. Itt választhat a *Play, Score* és a *Quit* menüpontok közül.



A *Play* elindítja a játékot, ha korábban már játszottunk, akkor a legutolsó játékmenet töltődik be. Ha még nem játszottunk egy teljesen új játék indul. A játék menet a NYÍL billentyűkkel irányítható, a lényege, hogy az egyforma értékű csempéket összeolvasszuk, így egy nagyobb csempét létrehozva elérjük a 2048-as értékűt. A játék nem áll meg ha ezt elértük, megy tovább és számolja tovább a pontokat.



A *Scores* menüpontra lépve ez a képernyő fogad minket. Itt látható legfelül a legjobb pontszámunk, amit sikerült elérnünk. Alatta az előző pontszámok felsorolva.