

Programozás Alapjai 3 – Házi Feladat

Dokumentáció

Jankó András – NHVU6N

1.Választott Feladat Specifikációja:

Aknakereső: Egy klasszikus aknakereső játék, ahol egy négyzetrácsos pályán kell megtalálni az összes bombát, amelyek elvannak rejtve a felhasználó előtt. A user először egy menüt fog látni, ahol elindíthat egy új játékot vagy betöltheti egy régi mentését, és egyes beállításokat is el tud érni. A pálya állása tehát kimenthető, és visszatölthető lesz. Miután elindítottunk a játékot alapesetben minden rács állapota ismeretlen lesz, de ha rákattint egyre a cella tartalma kiderül, ha a cella maga egy bomba volt akkor a játéknak vége és a mosolygós fej megnyomásával egy új játék indítható. Ha a cellában nem volt bomba a játék kiírja mennyi bomba van a szomszédságában. Mikor egy celláról úgy véli a játékos, hogy ott bomba van megjelölheti egy „flaggel”, ami később ottmarad jelezve azt a játékosnak. Úgy lehet nyerni a játékban, hogy az összes bombát nem tartalmazó cellát felfedi a játékos. A program grafikus felületét SWING-el valósítom meg.

2.Use Case leírások

A főmenüben

Play Game:

A felhasználó a play game menüpontra kattintva egy játékot kezd el, de ha van mentett játék akkor az töltődik be.

Scoreboard:

A scoreboard menüpont alatt megtekinthetők a régebbi elért eredmények.

Options

Az options menüpont alatt állítható be a zene, és a hardmode.

Exit Game:

Az exit gammel bezáródik a játék.

A játék közben

Click:

Ha játék közben a felhasználó egy cellára kattint azzal felfedi annak a tartalmát a fentebb leírtaknak megfelelően.

Flagged Click:

A zászló ikonra kattintva, és ezután egy cellára ráclikkelve a játék megjelöli azt a cellát egy flaggel amire nem lehet majd kattintani amíg le nem szedtük a flaget róla. Ezt egy újabb kattintással lehet elérni.

Click on Emoji:

Ha rákattintunk a középen lévő mosolygós fejecskére a játék újraindul.

Set Spacing:

A set spacing menüponttal a cellák között lévő távolságot tudjuk beállítani.

3.Osztályok

1. Main

Felelősségek

Beállítja a Nimbus black theme-t, és elindítja egy külön threaden a Menu-t.

Attribútumok

-

Metódusok

-

2. Menu

Felelősségek

A user több lehetőség közül tud választani benne, eltudja indítani a játékot, beállításokat tud kezelni, és megtudja tekinteni a ranglistát.

Attribútumok

jlab: JLabel	Az ablak tetején lévő kis menu.
Jfrm: JFrame	Maga az ablak.
optionsMenu: OptionsMenu	A beállításokat kezelő osztály.
scoreboard: Scoreboard	A ranglistát kezelő osztály.

Metódusok

+run()	Elindít egy audio file-t, és megnézi hogy leállították e.
+actionPerformed(ae: Actionevent)	Egy switch case-ben vizsgálja hogy mit választott a user.

3. OptionsMenu

Felelősségek

Az elérhető beállításokat kezeli.

Attribútumok

jfrm: JFrame	Az ablak.
hardmode: boolean	Nehéz fokozatot jelzi.
music: boolean	A zene ki/be kapcsolható vele.

Metódusok

+makeVisible	Láthatóvá teszi az ablakot.
--------------	-----------------------------

4. Scoreboard

Felelősségek

A ranglistát kezeli.

Attribútumok

Data: UserData	A lista ahol a userek eredményeit tároljuk.
newName: JTextfield	Textfield a névnek.
newPoint: JTextfield	Textfield a pontnak.

Metódusok

+ initComponents()	Beállítja az ablak komponenseit.
+ saveAll()	Elmenti a ranglista állapotát.
+ getData()	Visszatér a userek listájával.

5. UserData

Felelősségek

Ebben az osztályban tároljuk a különböző usereket.

Attribútumok

Users: List<User>	A userek listája.
-------------------	-------------------

Metódusok

+ getRowCount()	Visszaadja a users méretét.
+ getColumnCount()	Visszaadja mennyi oszlop van.
+ getValueAt(rowIndex: int, columnIndex: int)	Visszaadja vagy a user nevét vagy a pontját.
+ addTableModelListener(tl: TableModelListener)	Listenert ad a userdata-hoz.
+ setValueAt(o: Object, i: int, j: int)	Egy adott indexre beállít egy objektumot.
+ addUser(name: String, point: int)	Hozzáad egy usert a listánkhoz.

6. User

Felelősségek

Egy usert modellez.

Attribútumok

Name: String	A user neve.
Point: int	A user pontja

Metódusok

Csak getter, és setter metódusai vannak.

7. GUI

Felelősségek

Magáért a játékmenetért felel.

Attribútumok

timeX: int	Idő kijelző x koordináta.
timeY: int	Idő kijelző y koordináta.
Sec: int	Idő kijelzőhöz egyre növekvő változó.
randomStuff: Boolean	A hardmode-ot jelzi.
Board: Board	Belső osztály, ezzel valósítom meg a grafikus dolgokat.
Click: click	Az egér lenyomását figyelő osztály.
Scoreboard: Scoreboard	A ranglista.
Popup: PopupUsername	Felugró ablak a user bejelentkezéséhez.

Metódusok

+run()	Futás közben időközönként repainteljük az ablakot, és megnézzük hogy nyert-e a játékos..
+saveResult()	Elmenti a scoreboardba a user eredményét.
+checkVictoryStatus()	Megnézi nyert vagy veszített a user.
+totalMines(): int	Az összes bomba számosságával tér vissza.
+totalBoxesRevealed(): int	Az összes felfedett cella száma.

8. Click

Felelősségek

Az elérhető beállításokat kezeli.

Attribútumok

startDate:Date	A számlálóhoz kell.
endDate: Date	Mikor megáll a számláló.
Rand: Random	A bombákat randomizálja.
Move: Move	Az egér mozgásáért felel.
Level: Level	A pálya objektuma.
Spacing: int	A cellák között lévő távolság.
Flagger: Boolean	A zászló kijelölését jelzi.
Happiness: Boolean	A smiley hangulatát modellezi.
Victory: Boolean	A játék végét jelzi.
Defeat: Boolean	A vereséget jelzi.

Metódusok

+inBoxX(): int	Megadja hogy az eger x pozicioja egy cellaban van-e.
+inBoxY(): int	Ugyanez y-ra.
+mouseClicked(e: MouseEvent)	A clickelésre történő interakciókat állítja be.
+resetAll()	Újrakezdi a játékot.
+inFlagger(): boolean	Megadja hogy az egér a flaggerben van-e.

9. Move

Felelősségek

Az egér mozgását követi.

Attribútumok

Mx: int	Egér x koordinátája.
My: int	Egér y koordinátája.

Metódusok

+mouseMoved(e: MouseEvent)	A fenti változókbá kerülnek az egér koordinátái.
----------------------------	--

10. Level

Felelősségek

A pálya állapotát tartalmazza

Attribútumok

Neighbours: int[][]	Egy matrix ami tárolja a szomszédokat.
Revealed: int[][]	Egy matrix ami számontartja melyik cellákat fedték már fel.
Flagged: int[][]	Egy zászlójelzős matrix.
Mines: int[][]	A bombákat tartja számon

Metódusok

+IsN(mx,my,cx,cy): boolean	Megmondja hogy egy adott cella körül egy adott ciklusban van-e bomba.
+saveLevel()	Elmenti a pálya állapotát.

4.UML Diagram:

