

Fungorium

68 – nullpointerexception

Konzulens:

Simon Balázs

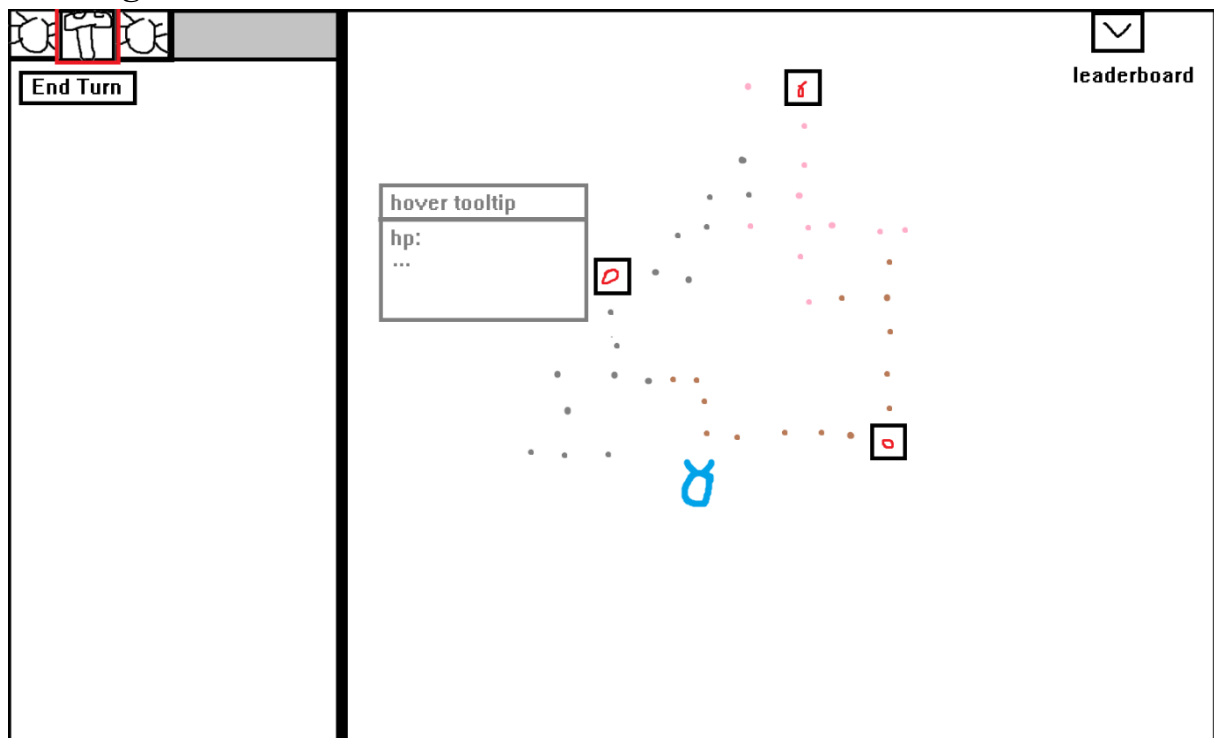
10. Grafika tervezés

Csapattagok

Fórián-Szabó Bernát	ABIVEP	foriansziga@gmail.com
Zsigmond		
Gyárfás Réka	AFVLHM	gyarfas.reka@gmail.com
Kemecsei Kornél	HDB6X9	kornel.kemecsei@edu.bme.hu
Kuzmin Iván	U0725D	ikuzmin@edu.bme.hu
Georgijevics		
Tóth Mihály Balázs	OAYOF1	toth.misi05@gmail.com

2025.05.05.

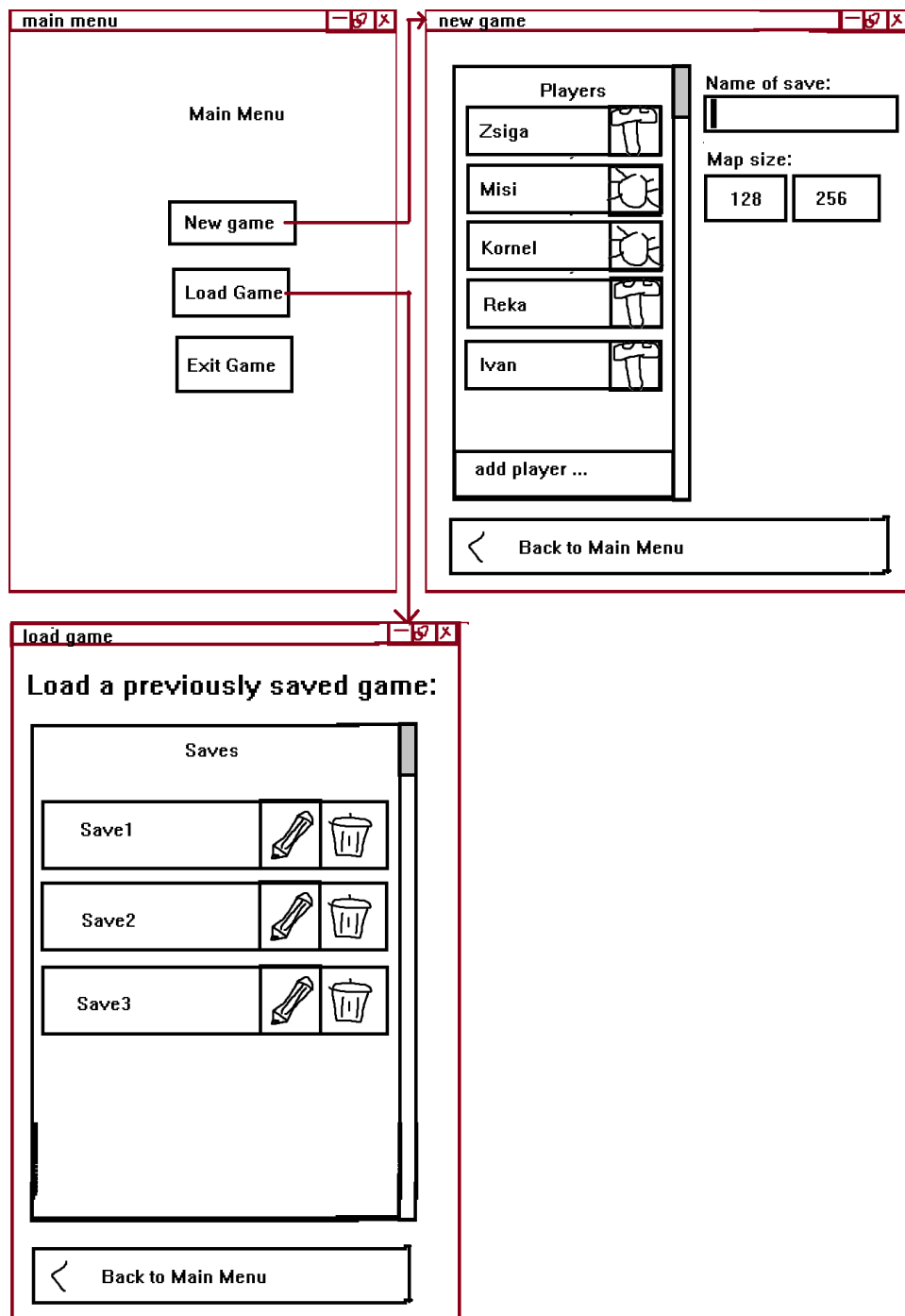
10.1 A grafikus interfész



In-Game HUD

Evaluate Game Now

Save & Exit game



10.2 A grafikus rendszer architektúrája

10.2.1 A felület működési elve

10.2.1.1 Általános Felépítés

A fő ablak egy PanelRenderer panel, amely dinamikusan váltja a megjelenítendő tartalmat (stratégiát). Különböző "képernyők" vagy "jelenetek" léteznek (stratégiák):

- Főmenü (MainMenuStrategy)
- Új Játék Beállításai (NewGameSetupStrategy)
- Játék Képernyő (InGameStrategy)

Az interaktív elemek (gombok, szövegmezők) vizuális visszajelzést adnak (színváltozás egér fölött, lenyomáskor).

10.2.1.1.1 Főmenü (MainMenuStrategy)

Megjeleníti a játék címét ("Fungorium").

Egy animált grafikát játszik le középen.

Gombok:

- "Start Game" (Új játék indítása)
- "Open Console" (Teszt konzol megnyitása)
- "Exit" (Kilépés a játékból)

Alul megjeleníti a készítők nevét ("Made by NullPointerException").

10.2.1.1.2 Új Játék Beállításai (NewGameSetupStrategy)

Megjeleníti a képernyő címét ("New Game Setup").

- Bal oldali panel:
 - Játékosok listája:
 - Minden játékos mellett egy ikon jelzi a típusát (Gomba/Rovar). Az ikonra kattintva váltható a típus.
 - A játékos neve egy szerkeszthető szövegmező (CustomTextField).
 - A lista alatt egy "Add player..." gomb új játékos hozzáadásához.
- Jobb oldali panel:
 - "Name of save:" címke és egy szövegmező (CustomTextField) a mentés nevének megadásához.
 - "Map size:" címke és gombok a térkép méretének kiválasztásához ("32", "64"). Csak a "32"es érhető el alapból.
- Alul:
 - "Confirm Setup" gomb: Csak akkor aktív, ha van legalább egy Gomba és egy Rovar játékos, és a mentés neve meg van adva. Elindítja a játékot a beállításokkal.
 - "< Back" gomb: Visszalép a Főmenübe.

10.2.1.1.3 Játék Képernyő (InGameStrategy)

Két fő részre osztott nézet: bal oldali információs sáv (HUD) és jobb oldali játéktérkép.

Bal oldali HUD:

- Aktuális kör száma.
- Aktuálisan lépő játékos neve.
- Játékosok ikonjai listázva:

- Az aktuálisan lépő játékos ikonja piros kerettel ki van emelve.
- Az ikonok alatt a játékosok nevei láthatók.

"Next Turn" gomb:

- A következő játékosra léptet. (Elhelyezés fázisában inaktív).

Jobb oldali Térkép:

- Háttérkép (vagy zöld négyzet, ha a kép nem töltődik be).
- A térképen a Gomba (FungusBody) és Rovar (Insect) egységek a saját ikonjaikkal jelennek meg a megfelelő mezőkön (Tile).

Elhelyezés Fázis:

Amikor egy játékosnak egységet kell letennie, az egérkurzor helyén áttetszően megjelenik a lehelyezendő egység ikonja. A térképre kattintva lehet lehelyezni az egységet.

10.2.1.2 Technikai Részletek (Osztályok és Működés)

10.2.1.2.1 Alap Struktúra és Megjelenítés

10.2.1.2.1.1 PanelRenderer

- Egy JPanel leszármazott, amely a központi rajzolófelület.
- Felelős az aktuális IRenderStrategy tárolásáért és annak render metódusának meghívásáért a paintComponentben.
- Kezeli az egér (kattintás, mozgatás) és billentyűzet eseményeket, amelyeket továbbít az aktuális stratégiának.
- Egy Timer használ a MainMenuStrategy animációjának folyamatos újrarajzolásához.
- Fókuszálható a billentyűzet események fogadásához (szövegmezők miatt).

10.2.1.2.1.2 IRenderStrategy

- Interfész, amely meghatározza a megjelenítési stratégiák közös metódusait:
- render(Graphics2D g2d, Dimension dimension): A képernyő kirajzolása.
- updateHover(int mouseX, int mouseY): Az egér pozíciója alapján frissíti az interaktív elemek hover állapotát.
- handlePress(int mouseX, int mouseY): Egérle nyomás kezelése.
- handleRelease(int mouseX, int mouseY): Egérfelengedés kezelése (általában itt történik a gombnyomás akciója).
- getTextFields(): Visszaadja a stratégiához tartozó szövegmezőket (billentyűzetkezeléshez).
- AbstractRenderStrategy: Absztrakt osztály, amely implementálja az IRenderStrategyt.
- Közös funkcionalitást biztosít a gombok (buttons) és szövegmezők (textFields) listájának kezelésére.
- Implementálja az updateHover, handlePress, handleRelease metódusokat úgy, hogy azok végigiterálnak az elemeken és beállítják az állapotukat (hovered, pressed, focused).
- Az onClicked(CustomButton btn) absztrakt metódust a leszármazottaknak kell implementálniuk a gombnyomások kezelésére.

10.2.1.2.2 *Konkrét Stratégiák*

10.2.1.2.2.1 **MainMenuStrategy**

- Kirajzolja a háttér, címet, animációt (képkockák betöltése static blokkban, időzített váltás), gombokat és a lábléceket.
- Az `onButtonClicked` metódus a `MainMenuPresenter`-en keresztül jelzi a felhasználó szándékát (Start, Console, Exit).
- `NewGameSetupStrategy`:
- Kirajzolja a felületet: cím, játékoslista (`CustomPlayerList` segítségével), gombok, szövegmezők.
- Kezeli a játékosok hozzáadását (`addPlayerButton`), a típusváltást (ikon gombok), a mentés nevének és térképméretnek beállítását.
- Az `onButtonClicked` és a szövegmezők eseményei a `NewGameSetupPresenter`-t hívják meg.
- A `syncPlayerTextFields` szinkronizálja a játékosok listáját a megjelenített szövegmezőkkel és ikon gombokkal.
- `handleKeyPressEvent`: Továbbítja a billentyűleütéseket a fókuszált szövegmezőnek.
- `InGameStrategy`:
- Kirajzolja a HUDot (kör, játékos infók, ikonok, "Next Turn" gomb) és a játéktérképet (háttér, egységek).
- Az `onButtonClicked` a "Next Turn" gomb esetén növeli a kör számát a `GameCoordinator`-ban.
- Kezeli az elhelyezési fázist (`placementPhase`):
- `updateHover`: Beállítja a `placementHover` pontot az egér helyzetének megfelelően.
- `handlePress`: Ha elhelyezési fázis van és a kattintás a térképen történt, létrehozza a megfelelő egységet (`FungusBody` vagy `Insect`) a `GameCoordinator`-ban lévő adatok alapján, hozzáadja a mezőhöz (`Tile`), majd növeli a kör számát és lépteti az elhelyező játékost, vagy kilép az elhelyezési fázisból.

10.2.1.2.2.2 **Prezentálók (Presenters MVP Minta)**

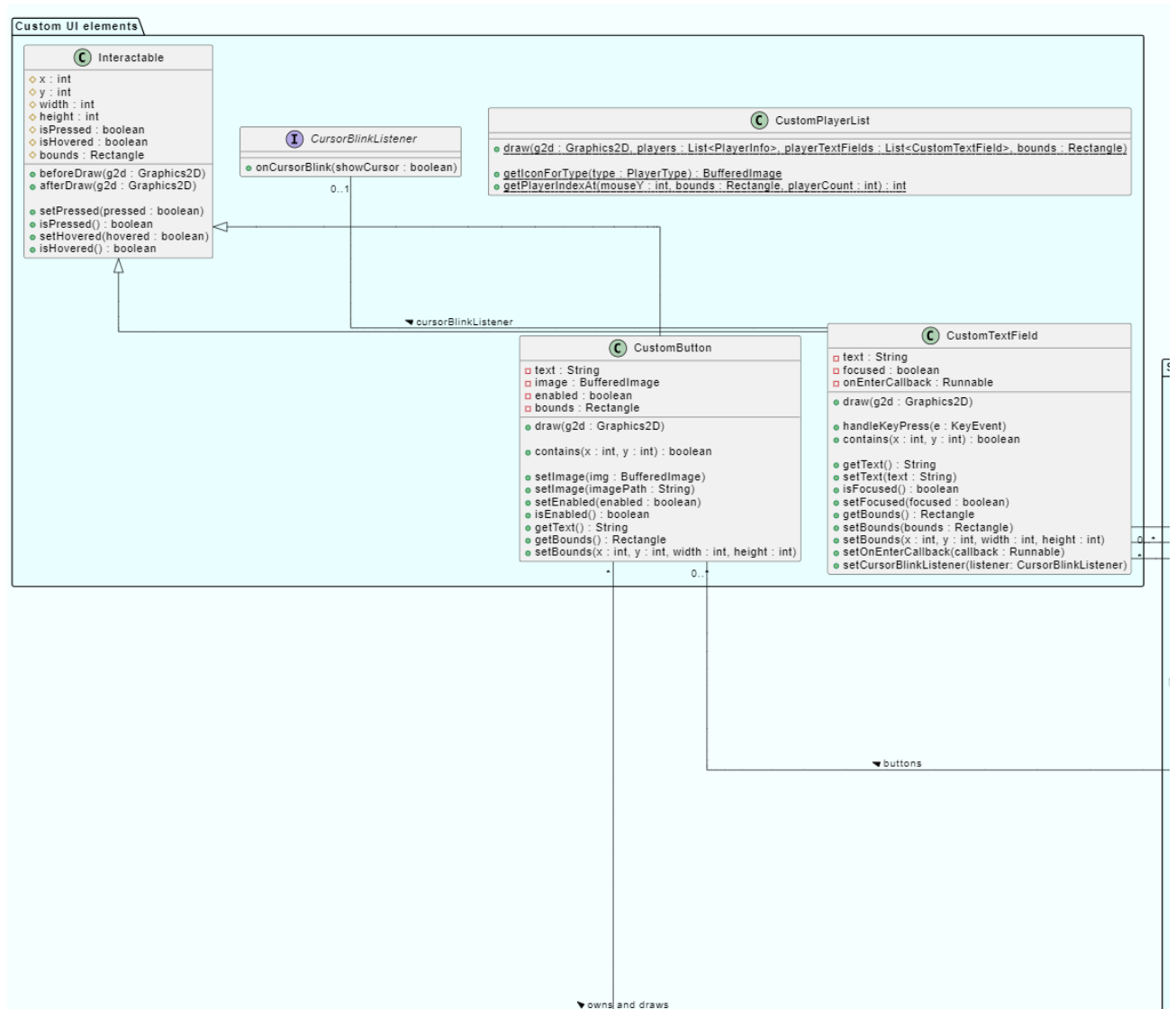
- Céljuk a Nézet (Stratégia) és a Modell/Logika (`GameCoordinator`) szétválasztása. A Nézet a `Presenter`-nek szól, a `Presenter` pedig a `GameCoordinator`-nak.
- `MainMenuPresenter`: Fogadja a `MainMenuStrategy`-tól érkező gombnyomásjelzéseket (`onStartGameClicked`, `onTesterClicked`, `onExitClicked`) és továbbítja azokat a `GameCoordinator`-nak (pl. képernyőváltás, kilépés).
- `NewGameSetupPresenter`:
- Tárolja az új játék beállításait (játékoslista, mentésnév, térképméret), amelyeket a `NewGameSetupStrategy` módosít a metódusain keresztül (`addPlayerRequested`, `updatePlayerName`, `togglePlayerType`, `setMapSize`, `setSaveName`).
- A "Confirm" gomb lenyomásakor (`onConfirmSetupClicked`) átadja a beállításokat a `GameCoordinator`-nak a játék indításához.
- A "Back" gombnál (`onBackToMainMenuClicked`) jelzi a `GameCoordinator`-nak, hogy térjen vissza a főmenübe.
- Adatokat szolgáltat a `NewGameSetupStrategy`-nak a megjelenítéshez (`getPlayers`, `getMapSize`, stb.).
- `InGamePresenter`:
- Adatokat közvetít a `GameCoordinator`-tól az `InGameStrategy` felé (játékosok, térképméret, HUD szélesség, aktuális kör, mezők lekérdezése).

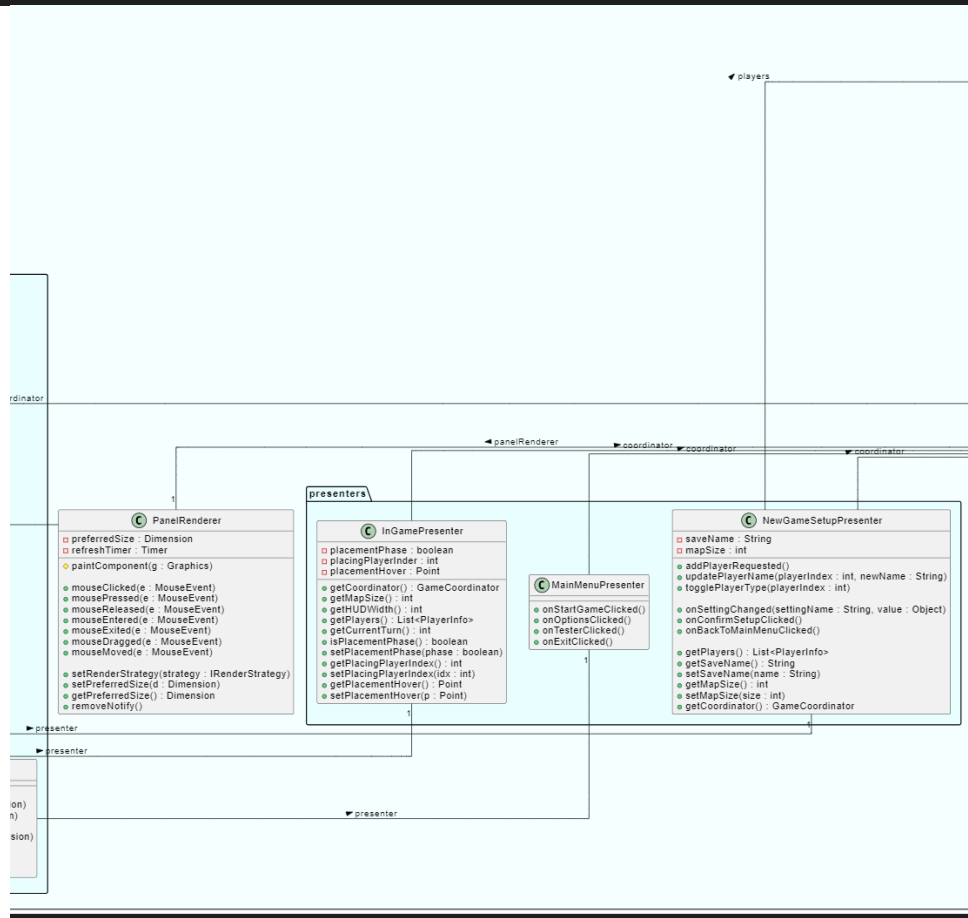
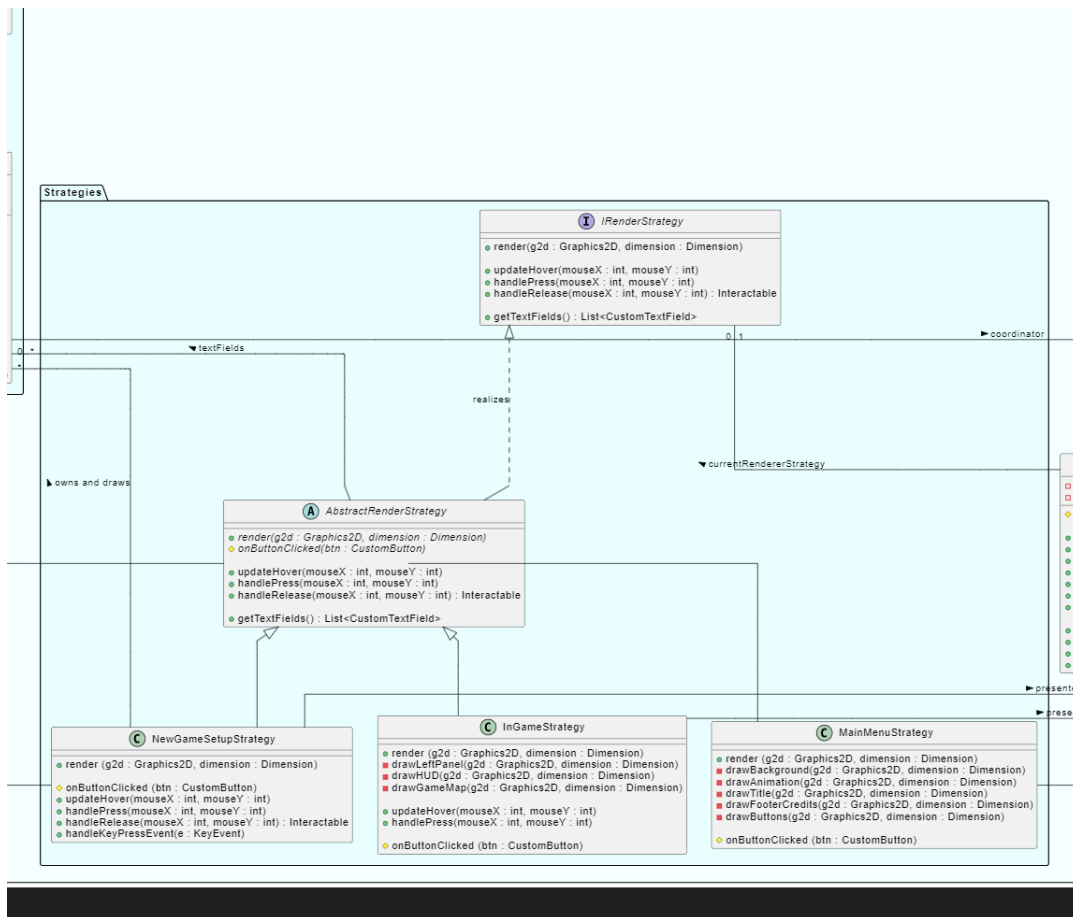
- Kezeli az elhelyezési fázis állapotát (placementPhase, placingPlayerIndex, placementHover).
- *Megjegyzés: Az InGameStrategy néhány helyen (pl. kör léptetés) közvetlenül is hozzáfér a GameCoordinatorhoz a Presenteren keresztül (presenter.getCoordinator()), ami kissé eltér a szigorú MVPTől, de a kód így működik.*

10.2.1.2.2.3 Egyedi UI Elemek (graphics.customUIElements)

- Interactable: Absztrakt ősszotály az interaktív elemekhez.
- Tárolja a pozíciót (x, y), méretet (width, height), bounds (téglalap), és az állapotokat (isHovered, isPressed).
- beforeDraw/afterDraw: Segédfüggvények a Graphics2D állapotának mentésére/visszaállítására rajzolás előtt/után.
- CustomButton: Interactable leszármazott.
- Megjelenít egy gombot szöveggel és/vagy képpel.
- Különböző színeket használ a normál, hover, lenyomott és inaktív (enabled) állapotokhoz.
- A contains metódus ellenőrzi, hogy egy pont a gombon belül vane (csak ha enabled).
- CustomTextField: Interactable leszármazott.
- Egy egyszerű szövegbeviteli mező.
- Kezeli a szöveg (text) tárolását, a fókusz (focused), a kurzor pozícióját (cursorPosition) és a kurzor villogását (showCursor, Timer).
- A handleKeyPress metódus kezeli a billentyűleütéseket (karakterbevitel, törlés, backspace, nyilak, home/end, enter).
- Az Enter lenomásakor meghívhat egy onEnterCallbacket.
- Szüksége van a GameCoordinatorra a villogás és gépelés közbeni újrarajzolás kéréséhez.
- CustomPlayerList: Statikus segédosztály.
- draw: Kirajzol egy játékoslistát egy adott Rectangle területre. Felhasználja a Player objektumokat az ikonokhoz (getIconForType segítségével) és a CustomTextField listát a nevek megjelenítéséhez és szerkesztéséhez. Kiszámolja az elemek elhelyezkedését.
- getIconForType: Visszaadja a megfelelő ikont (BufferedImage) a játékos típusa alapján.
- getPlayerIndexAt: Megállapítja, hogy egy adott Y koordináta a lista melyik játékosára esik.

10.2.2 A felület osztály-struktúrája







10.3 A grafikus objektumok felsorolása

10.3.1 PanelRenderer

Felelősség

A PanelRenderer osztály egy egyedi JPanel, amely a stratégiai tervezési minta (IRenderStrategy) segítségével felel a grafikus felület dinamikus kirajzolásáért. Feladata, hogy a hozzá rendelt megjelenítési stratégia szerint jelenítse meg a felületet, valamint kezelje az egér és billentyűzet eseményeit. Különös figyelmet fordít a főmenü stratégia (MainMenuStrategy) esetén a rendszeres újrarajzolásra (30 FPS frissítési ráta). Továbbá lehetővé teszi a felhasználói interakciókat (pl. kattintás, fókusz kezelés), így szoros integrációban működik az UI komponensekkel.

Össztályok

java.lang.Object → javax.swing.JComponent → javax.swing.JPanel →
graphics.PanelRenderer

Interfészek

- java.awt.event.MouseListener
- java.awt.event.MouseMotionListener
- **Attribútumok**
- - currentRenderStrategy : IRenderStrategy
A jelenleg beállított renderelési stratégia, amely a panel kirajzolását vezérli.
Láthatóság: privát (-).
- - refreshTimer : Timer
Egy Swing-es időzítő, amely periodikusan újrarajzolja a panelt bizonyos stratégiák (pl. főmenü) esetén. Láthatóság: privát (-).
- - preferredSize : Dimension
A panel preferált mérete. Az initPanel() során állítódik be, és később módosítható.
Láthatóság: privát (-).

Metódusok

- + PanelRenderer(IRenderStrategy initialStrategy)
Konstruktor, amely beállítja az adott stratégiát és elindítja az időzítőt. Láthatóság: publikus (+).
- + PanelRenderer()
Konstruktor, amely null stratégiával hívja meg a fenti konstruktort. Láthatóság: publikus (+).
- - void initPanel(IRenderStrategy strategy)
Inicializálja a panelt, beállítja az eseményfigyelőket, méretet, fókuszt stb. Láthatóság: privát (-).
- - void startRefreshTimer()
Elindít egy 32 ms-os időzítőt, amely újrarajzolja a panelt a MainMenuStrategy esetén. Láthatóság: privát (-).
- - KeyAdapter createKeyAdapter()
Létrehoz egy KeyAdapter példányt, amely billentyűlenyomásokat kezel. Láthatóság: privát (-).
- + void setRenderStrategy(IRenderStrategy strategy)
Beállítja az aktuális megjelenítési stratégiát. Ha null, hibaüzenetet ír ki. Automatikus újrarajzolást kér. Láthatóság: publikus (+).
- # void paintComponent(Graphics g)
Felüldefiniálja az alapértelmezett kirajzolást. A kirajzolást delegálja a stratégia objektumnak. Láthatóság: protected (#).

- + void setPreferredSize(Dimension d)
Beállítja a panel preferált méretét. Láthatóság: publikus (+).
- + Dimension getPreferredSize()
Visszaadja a preferált méretet. Láthatóság: publikus (+).
- # void removeNotify()
Törléskor leállítja a frissítő időzítőt. Láthatóság: protected (#).

MouseListener metódusok:

- + void mousePressed(MouseEvent e)
Delegálja a nyomás eseményt a stratégiának, majd újrarajzol. Láthatóság: publikus (+).
- + void mouseReleased(MouseEvent e)
Delegálja az elengedés eseményt, majd kiírja, hogy melyik Interactable aktiválódott. Láthatóság: publikus (+).
- + void mouseExited(MouseEvent e)
Visszaállítja a hover állapotot, ha az egér elhagyja a panelt. Láthatóság: publikus (+).
- + void mouseEntered(MouseEvent e)
Nem végez műveletet. Láthatóság: publikus (+).
- + void mouseClicked(MouseEvent e)
Nem végez műveletet. Láthatóság: publikus (+).

MouseMotionListener metódusok:

- + void mouseDragged(MouseEvent e)
Hover állapot frissítése húzás közben. Láthatóság: publikus (+).
- + void mouseMoved(MouseEvent e)
Hover állapot frissítése mozgás közben. Láthatóság: publikus (+).

10.3.2 AbstractRenderStrategy

Felelősség

Az AbstractRenderStrategy osztály absztrakt bázisosztályként szolgál minden olyan renderelési stratégia számára, amely grafikus komponenseket (gombokat, szövegmezőket) kezel. Felelős az interaktív elemek – például gombok és szövegmezők – egységes hover, lenyomás és felengedés alapú kezeléséért. Emellett sablont biztosít a leszármazott osztályoknak a gombkattintások feldolgozásához (onButtonClicked). Ez az osztály biztosítja az egérműveletek kezelését és az interaktív komponensek állapotának frissítését.

Ősosztályok

AbstractRenderStrategy

Interfészek

IRenderStrategy

Ez az interfész definiálja az egységes renderelési és eseménykezelési metódusokat, amelyeket az osztály megvalósít.

Attribútumok

- - buttons: List<CustomButton>

A renderelendő és kezelendő gombokat tároló lista. Láthatóság: protected (-), típus: List<CustomButton>

- - textFields: List<CustomTextField>

A szövegbeviteli mezőket tároló lista. Ezeken lehet fókusz, valamint hover és kattintás eseményeket is figyel. Láthatóság: protected (-), típus: List<CustomTextField>

Metódusok

- + abstract void render(Graphics2D g2d, Dimension dimension)

Absztrakt metódus, amelyet minden konkrét stratégia implementál. A grafikus komponensek kirajzolásáért felelős. Láthatóság: public (+)

- + void updateHover(int mouseX, int mouseY)

Frissíti a gombok és szövegmezők hovered állapotát az egér aktuális pozíciója alapján. Ha a kurzor egy komponens felett van, annak hovered állapota igaz lesz. Láthatóság: public (+)

- + void handlePress(int mouseX, int mouseY)

Beállítja a pressed állapotot a gombon, amelyre az egér kattintott, valamint fókuszt ad a megfelelő szövegmezőnek. Egyszerre csak egy gomb vagy szövegmező lehet aktív. Láthatóság: public (+)

- + Interactable handleRelease(int mouseX, int mouseY)

Felengedés esemény kezelése. Visszaadja az aktuálisan elengedett Interactable elemet (ha volt ilyen), és meghívja az onClicked metódust a releváns gombon. Minden gomb pressed állapotát visszaállítja. Láthatóság: public (+)

- + List<CustomTextField> getTextFields()

Visszaadja az aktuális szövegmező listát. Hasznos lehet külső osztályok számára, például billentyűleütések delegálásához. Láthatóság: public (+)

- # abstract void onClicked(CustomButton btn)

Absztrakt metódus, amelyet a leszármazott osztályoknak kell implementálniuk. Akkor hívódik meg, amikor egy gombra kattintás történt. Lehetővé teszi a gomb-specifikus logika kezelését. Láthatóság: protected (#)

10.3.3 InGameStrategy

Felelősség

Az InGameStrategy osztály a játék futás közbeni (in-game) grafikus megjelenítéséért felel. A renderelési stratégia részeként biztosítja a háttér megjelenítését, a bal oldali kezelőpanel (HUD) kirajzolását, a játékosikonok elhelyezését, valamint a körváltó gomb és az elhelyezési fázis vizuális elemeinek megjelenítését. Kezeli a felhasználói interakciókat, például az egérkattintást és -mozgást is. Fontos szerepe van a játék körváltásának és entítaselhelyezésének irányításában a presenter segítségével.

Össztályok

java.lang.Object → graphics.strategies.AbstractRenderStrategy →
graphics.strategies.InGameStrategy

Interfészek

Nem valósít meg közvetlenül interfészeket, de absztrakt őssztyán keresztül örökölhet implementációkat.

Attribútumok

- - presenter: InGamePresenter
A grafikus stratégia mögötti adatok és műveletek eléréséhez szükséges presenter objektum. Privát láthatóságú.
- - backgroundImage: BufferedImage
A játék háttérként megjelenített kép. Privát láthatóságú.
- - nextTurnButton: CustomButton
A „Next Turn” gomb, amely a játékosváltást indítja. Privát láthatóságú.
- - TILE_SIZE: int
A térkép csempéinek mérete. Privát, static final.
- - PLAYER_ICON_SIZE: int
A játékosikonok mérete a HUD-on. Privát, static final.
- - PLAYER_ICON_GAP: int
Az ikonok közötti függőleges távolság. Privát, static final.
- - PLAYER_ICON_START_Y: int
A játékosikonok kezdő y-koordinátája a bal oldali panelen. Privát, static final.
- - FUNGUS_ICON: BufferedImage
A gombás játékos ikonja. Privát, static final.
- - INSECT_ICON: BufferedImage
A rovaros játékos ikonja. Privát, static final.

Metódusok

- + InGameStrategy(InGamePresenter presenter)
Konstruktor, beállítja a presenter objektumot, betölti a háttérét és inicializálja a gombokat. Nyilvános (+).
- + void render(Graphics2D g2d, Dimension dimension)
A fő renderelő metódus. Kirajzolja a HUD-ot, térképet, valamint az elhelyezési ikont, ha szükséges. Nyilvános (+).
- - void drawLeftPanel(Graphics2D g2d, Dimension dimension)
A bal oldali HUD panel rajzolását végzi: játékosikonok, aktuális kör, "Next Turn" gomb. Privát (-).
- - void drawGameMap(Graphics2D g2d, Dimension dimension)
A térkép háttérképének kirajzolását végzi. Privát (-).
- # void onButtonClicked(CustomButton btn)
A CustomButton eseményének kezelése. Ha a „Next Turn” gombot nyomták meg, akkor növeli az aktuális kört és újrarajzoltatja a képernyőt a koordinátorral. Védett (#).
- + void updateHover(int mouseX, int mouseY)
A hover pozíció frissítése, ha elhelyezési fázisban vagyunk. Felülírja az ős metódust. Nyilvános (+).
- + void handlePress(int mouseX, int mouseY)
Egérkattintás eseményének kezelése. Ha elhelyezési fázis van, akkor a megfelelő mezőre helyezi a játékos entitását, majd kezeli a körváltást vagy a fázis végét. Egyébként az ős metódus hívódik meg. Nyilvános (+).

10.3.4 IRenderStrategy

Felelősség

Az `IRenderStrategy` interfész meghatározza azokat a műveleteket, amelyeket minden renderelési stratégia (pl. játékmenet, főmenü, beállítások képernyő stb.) implementációjának kötelező megvalósítania. Ezzel biztosítja, hogy az alkalmazás különböző nézetei egységes módon legyenek kezelhetők – különösen a megjelenítés (render) és az interakciók (egérkezelés) tekintetében. Ez az interfész alapot nyújt az ún. **stratégia tervezési mintához**, lehetővé téve a nézetek dinamikus cseréjét egy központi kezelő osztály által.

Össztályok

- `java.lang.Object` → Interfészként nincs szuperosztálya.

Interfészek

- Ez maga egy interfész, így nem valósít meg más interfészeket.

Attribútumok

- Az interfész nem tartalmaz attribútumokat (mezőket), kizárólag metódusdefiníciókat.

Metódusok

- `+ void render(Graphics2D g2d, Dimension dimension)`
A teljes képernyő vagy panel kirajzolásáért felelős metódus. Minden implementációnak meg kell határoznia, hogyan jelenik meg az adott képernyőn a tartalom. Általában a `JPanel.paintComponent()` metódusból hívódik.
Láthatóság: + (publikus, absztrakt)
- `+ void updateHover(int mouseX, int mouseY)`
Frissíti az egér pozíciójának megfelelő elemek „hover” állapotát (pl. ha egy gomb fölé visszük az egeret, kiemelődik).
Láthatóság: + (publikus, absztrakt)
- `+ void handlePress(int mouseX, int mouseY)`
Kezeli az egér lenyomásának eseményét. Használható például gomb nyomva tartásának effektusához.
Láthatóság: + (publikus, absztrakt)
- `+ Interactable handleRelease(int mouseX, int mouseY)`
Kezeli az egér gombjának felengedését, és visszaadja azt az interaktív elemet (pl. `CustomButton`), amelyre a kattintás történt. Ez segíti a logikai kezelés leválasztását a vizuális megjelenítésről.
Láthatóság: + (publikus, absztrakt)
- `+ List<CustomTextField> getTextFields()`
Az adott képernyőhöz tartozó egyedi szövegmezők (`CustomTextField`) listáját adja vissza. Például egy bejelentkező képernyőn ez tartalmazhatja a felhasználónév és jelszó mezőket.
Láthatóság: + (publikus, absztrakt)

10.3.5 MainMenuStrategy**Felelősség**

A `MainMenuStrategy` osztály a **`IRenderStrategy`** interfész konkrét implementációja, amely a játék főmenüjének megjelenítését és interakcióit kezeli. Ez az osztály felelős a háttér, a cím, a gombok, az animáció és a lábléc (credit) kirajzolásáért a főmenü képernyőn, valamint az egyes gombokhoz tartozó kattintási események kezeléséért. Az osztály ezen kívül animációs kereteket is kezel, amelyek a menü részeként dinamikusán frissülnek.

Össztályok

- `java.lang.Object` → `AbstractRenderStrategy`

Interfészek

- **IRenderStrategy:** Az osztály implementálja a IRenderStrategy interfészt, így köteles megvalósítani annak minden metódusát, mint például a render, updateHover, handlePress, handleRelease és getTextFields.

Attribútumok

- **presenter:** A MainMenuPresenter típusú referencia, amely az események kezelését végzi, például a gombok kattintásainak feldolgozása.
Láthatóság: private, típusa: MainMenuPresenter
- **ANIMATION_FRAMES:** Statikus lista, amely az animációs képkockákat tárolja. Ezek a képek a főmenü animációjához szükségesek.
Láthatóság: private static, típusa: List<BufferedImage>
- **ANIMATION_FRAME_DELAY:** A képkockák közötti késleltetés milliszekundumban.
Láthatóság: private static, típusa: int
- **lastFrameTime:** Az utolsó képkocka megjelenítésének ideje, amely az animáció sebességét irányítja.
Láthatóság: private, típusa: long
- **currentFrameIndex:** Az aktuális képkocka indexe az animációs lista alapján.
Láthatóság: private, típusa: int
- **Metódusok**
- **+ void render(Graphics2D g2d, Dimension dimension)**
A főmenü megjelenítéséért felelős metódus, amely különböző segédmetódusok segítségével rajzolja ki a háttér, a címet, a gombokat, az animációt és a lábléceket.
Láthatóság: + (publikus)
- **+ void drawBackground(Graphics2D g2d, Dimension dimension)**
Az osztály háttérének kirajzolásáért felelős metódus. A háttér színét és méretét határozza meg a dimension alapján.
Láthatóság: + (privát)
- **+ void drawTitle(Graphics2D g2d, Dimension dimension)**
A főmenü címének (pl. "Fungorium") kirajzolásáért felelős metódus.
Láthatóság: + (privát)
- **+ void drawFooterCredits(Graphics2D g2d, Dimension dimension)**
A lábléc (pl. "Made by NullPointerException") kirajzolásáért felelős metódus.
Láthatóság: + (privát)
- **+ void drawButtons(Graphics2D g2d)**
A főmenü gombjainak kirajzolásáért felelős metódus, amely végigiterál a gombokon és meghívja a draw metódust.
Láthatóság: + (privát)
- **+ void drawAnimation(Graphics2D g2d, Dimension dimension)**
Az animációs képkockák kirajzolásáért felelős metódus, amely a képernyő közepére rajzolja az animációt a ANIMATION_FRAMES listából.
Láthatóság: + (privát)
- **+ void onButtonClicked(CustomButton btn)**
Az eseménykezelő metódus, amely a gombok kattintásait kezeli, és a megfelelő műveletet hívja meg a presenter példányon.
Láthatóság: + (publikus)

10.3.6 NewGameSetupStrategy

Felelősség:

A NewGameSetupStrategy osztály felelőssége, hogy kezelje az új játék beállításait, beleértve a játékosok hozzáadását, a térkép méretének kiválasztását, és az új játék indítását. Az osztály a felhasználói interakciókat kezeli, például a gombnyomásokat és az input mezők frissítését, hogy lehetővé tegye a felhasználónak az új játék konfigurálását.

Össztályok:

- **AbstractRenderStrategy:** A NewGameSetupStrategy közvetlenül az AbstractRenderStrategy osztályból származik, így örökli annak funkcionalitását a grafikus felület megjelenítésére.

Interfészek:

- A NewGameSetupStrategy nem implementál közvetlenül interfészeket, de az osztály által használt egyéb osztályok és komponensek (pl. CustomButton, CustomTextField, Interactable) interfészeket implementálhatnak, hogy támogassák az interaktív elemek kezelését.

Attribútumok:

- **presenter:** A NewGameSetupPresenter típusú attribútum, amely a játékosok kezelését végző presenter-t tárolja.
- **addPlayerButton:** A játékos hozzáadására szolgáló CustomButton.
- **mapSize32Button, mapSize64Button:** A térkép méretének választására szolgáló gombok.
- **confirmButton:** A beállítások megerősítésére szolgáló gomb.
- **backButton:** A visszalépést biztosító gomb.
- **saveNameField:** A játék mentésének nevére szolgáló szövegdoboz.
- **playerTextFields:** A játékosok nevének beírására szolgáló szövegdobozok listája.
- **playerIconButton:** A játékosok ikonjának kiválasztására szolgáló gombok listája.
- **focusedPlayerIndex:** A fókuszált játékos indexe.
- **playerListBounds:** A játékosok listájának területi határai.

Metódusok:

- **syncPlayerTextFields():** A játékosok szövegdobozait frissíti a presenter-ben lévő játékosokkal.
- **updatePlayerNameFromField(int index):** A játékos nevének frissítése a szövegdobozból.
- **render(Graphics2D g2d, Dimension dimension):** A grafikai elemek megjelenítése, beleértve a játékosok listáját, gombokat és szövegdobozokat.
- **onButtonClicked(CustomButton btn):** A gombnyomás eseményeinek kezelése, mint például játékos hozzáadása vagy térkép méretének kiválasztása.
- **updateHover(int mouseX, int mouseY):** Az egér mozgásának figyelése, hogy a gombok és szövegdobozok kiemelése megtörténjen.
- **handlePress(int mouseX, int mouseY):** Az egér kattintás kezelésére szolgáló metódus.
- **handleRelease(int mouseX, int mouseY):** Az egér gomb felengedése utáni események kezelése.
- **handleKeyPressEvent(KeyEvent e):** A billentyűleütés eseményeinek kezelése, hogy a szövegdobozokban történő bevétel frissüljön.
- **unfocusAllTextFields():** Az összes szövegdoboz fókuszaltságának eltávolítása.

10.3.7 InGamePresenter

Felelősség

Az InGamePresenter az MVP (Model-View-Presenter) minta Presenter rétege, amely közvetítő szerepet játszik a játék logika (model) és a játék képernyőjét megjelenítő InGameStrategy (view) között. Felelőssége az aktuális játékállapot lekérése és a felhasználói interakciók közvetítése a megfelelő modellekhez. A játék jelenlegi állapota, mint a térkép mérete, a játékosok listája és a HUD (képernyő alatti információs panel) szélessége, mind a Presenter-en keresztül érhetőek el. Emellett kezeli a játék elhelyezési fázisát és a játékosok elhelyezési indexét is.

Össztályok

- Nincs össztálya.

Interfészek

- Az osztály nem valósít meg interfészeket, de kapcsolatban áll a GameCoordinator osztállyal.

Attribútumok

- coordinator: A GameCoordinator típusú attribútum, amely a játékot vezérlő objektumot tárolja. Láthatósága: private, típusa: GameCoordinator.
- placementPhase: boolean típusú attribútum, amely a játék elhelyezési fázisának állapotát jelzi. Láthatósága: private.
- placingPlayerIndex: int típusú attribútum, amely a jelenleg elhelyező játékos indexét tárolja. Láthatósága: private.
- placementHover: Point típusú attribútum, amely a felhasználó által az elhelyezési fázisban kiemelt helyet jelöli. Láthatósága: private.

Metódusok

- InGamePresenter(GameCoordinator coordinator): A konstruktor felelőssége, hogy inicializálja az InGamePresenter objektumot a megfelelő GameCoordinator példánnyal.
- getCoordinator(): Publikus metódus, amely visszaadja a coordinator attribútumot.
- getMapSize(): Publikus metódus, amely visszaadja a térkép méretét a GameCoordinator-ból.
- getHUDWidth(): Publikus metódus, amely visszaadja a HUD szélességét a GameCoordinator-ból.
- getPlayers(): Publikus metódus, amely visszaadja a játékosok listáját.
- getCurrentTurn(): Publikus metódus, amely visszaadja a jelenlegi kör számát.
- isPlacementPhase(): Publikus metódus, amely visszaadja, hogy a játék jelenleg az elhelyezési fázisban van-e.
- setPlacementPhase(boolean phase): Publikus metódus, amely beállítja a játék elhelyezési fázisát.
- getPlacingPlayerIndex(): Publikus metódus, amely visszaadja a jelenleg elhelyező játékos indexét.
- setPlacingPlayerIndex(int idx): Publikus metódus, amely beállítja a jelenleg elhelyező játékos indexét.
- getPlacementHover(): Publikus metódus, amely visszaadja a jelenleg kiemelt elhelyezési helyet.
- setPlacementHover(Point p): Publikus metódus, amely beállítja a jelenlegi elhelyezési helyet.

10.3.7.1 MainMenuPresenter

Felelősség

A MainMenuPresenter osztály az MVP (Model-View-Presenter) minta Presenter komponense, amely közvetítő szerepet játszik a felhasználói felület (View) és az alkalmazás logikája (Model) között. Feladata a felhasználói interakciók (például gombnyomások) kezelésének delegálása a megfelelő logikai műveletekhez, amelyeket a GameCoordinator végez el. Ezáltal biztosítja a GUI és az alkalmazás logikájának elválasztását, lehetővé téve a könnyű tesztelhetőséget és karbantartást. A Presenter ezen kívül biztosítja, hogy a felhasználói interakciók helyes alkalmazás logikai műveletekhez kapcsolódjanak.

Össztályok

- Nincs őssztálya, közvetlenül a Object osztályból származik.

Interfészek

- Az osztály nem valósít meg interfészeket, de kapcsolatban áll a GameCoordinator osztállyal, amely a logikai műveleteket végzi.

Attribútumok

- coordinator: A GameCoordinator típusú attribútum, amely a játékot vezérlő objektumot tárolja. Láthatósága: private, típusa: GameCoordinator.
Leírás: A játék logikai vezérléséért felelős objektum, amely a Presenter által végrehajtott műveletekhez kapcsolódó funkcionalitást biztosít.

Metódusok

- MainMenuPresenter(GameCoordinator coordinator): Konstruktor, amely inicializálja a MainMenuPresenter objektumot és ellenőrzi, hogy a coordinator nem null értékű.
Láthatóság: public
Leírás: A konstruktor felelős a Presenter létrehozásáért, és biztosítja, hogy a GameCoordinator nem null értékű.
- onStartGameClicked(): Publikus metódus, amely kezelni hívja a játék indításához szükséges logikát, delegálva azt a coordinator-nek.
Láthatóság: public
Leírás: A "Start Game" gomb megnyomásakor aktiválódik, és a coordinator-t arra kéri, hogy mutassa meg az új játék beállítási képernyőt.
- onOptionsClicked(): Publikus metódus, amely kezeli az "Options" gombnyomást és a beállítások képernyőjét mutatja meg.
Láthatóság: public
Leírás: A "Options" gomb megnyomásakor aktiválódik, és a coordinator-t arra kéri, hogy mutassa meg az alkalmazás beállításait.
- onTesterClicked(): Publikus metódus, amely a tesztelő mód indításához szükséges logikát végzi el.
Láthatóság: public
Leírás: A "Tester" gomb megnyomásakor egy új App objektumot hoz létre és elindítja azt.
- onExitClicked(): Publikus metódus, amely kezeli az alkalmazás kilépését.
Láthatóság: public
Leírás: Az "Exit" gomb megnyomásakor a coordinator-t arra kéri, hogy lépjen ki az alkalmazásból.

10.3.7.2 NewGameSetupPresenter

Felelősség

A NewGameSetupPresenter osztály az MVP (Model-View-Presenter) minta Presenter komponense, amely közvetítő szerepet játszik az új játék beállításainak kezelése és a játék logikájának vezérlése között. A Presenter felelős a felhasználói interakciók (például új játékos hozzáadása, mentési név beállítása, térkép méret megadása) kezeléséért, valamint a játék indításához szükséges beállítások érvényesítéséért és továbbításáért a GameCoordinator-nak. Az osztály lehetővé teszi az új játék konfigurálását a felhasználói felületen keresztül.

Össztályok

- Nincs össztálya, közvetlenül a Object osztályból származik.

Interfészek

- Az osztály nem valósít meg interfészeket, de kapcsolatban áll a GameCoordinator osztállyal, amely végrehajtja a játék indításához szükséges műveleteket.

Attribútumok

- coordinator: A GameCoordinator típusú attribútum, amely a játékot vezérlő objektumot tárolja. Láthatósága: private, típusa: GameCoordinator.
Leírás: A játék logikai vezérléséért felelős objektum, amely a Presenter által végrehajtott műveletekhez kapcsolódó funkcionalitást biztosít.
- players: Egy lista (List<PlayerInfo>) a játékosok tárolására. Láthatósága: private, típusa: List<PlayerInfo>.
Leírás: A játékban résztvevő játékosok tárolására szolgáló lista, amely a PlayerInfo objektumokat tartalmazza.
- saveName: A játék mentési neve, alapértelmezett értéke "MyGame". Láthatósága: private, típusa: String.
Leírás: A játék mentésének neve, amelyet a felhasználó konfigurálhat.
- mapSize: A térkép mérete, alapértelmezett értéke 32. Láthatósága: private, típusa: int.
Leírás: A játék térképének mérete, amelyet a felhasználó választhat, és csak érvényes méretek engedélyezettek.

Metódusok

- NewGameSetupPresenter(GameCoordinator coordinator): Konstruktor, amely inicializálja a Presenter-t és beállítja az alapértelmezett játékosokat.
Láthatóság: public
Leírás: A konstruktor felelős a Presenter létrehozásáért, és biztosítja, hogy a GameCoordinator és a játékos lista inicializálva legyen.
- onSettingChanged(String settingName, Object value): Publikus metódus, amely kezeli a beállítások módosításait, és a változásoknak megfelelően frissíti a beállításokat.
Láthatóság: public
Leírás: A felhasználói felület által jelzett beállításváltozások kezelésére szolgál. A settingName alapján frissíti az aktuális beállításokat.
- addPlayerRequested(): Publikus metódus, amely új játékos hozzáadását kéri.
Láthatóság: public
Leírás: Új játékos hozzáadásakor váltogatja a játékos típusát (INSECT vagy FUNGUS), majd hozzáadja a játékos listához.
- setSaveName(String name): Publikus metódus, amely beállítja a játék mentési nevét.
Láthatóság: public
Leírás: Frissíti a játék mentési nevét, ha a felhasználó módosítja azt.
- updatePlayerName(int playerIndex, String newName): Publikus metódus, amely frissíti egy adott játékos nevét a listában.
Láthatóság: public

Leírás: A játékos nevét frissíti, ha az új név eltér az aktuális névtől. Hibát jelez, ha a megadott index érvénytelen.

- `togglePlayerType(int playerIndex)`: Publikus metódus, amely váltogatja egy játékos típusát (FUNGUS <-> INSECT).

Láthatóság: public

Leírás: A játékos típusát váltja, és a listában frissíti az adott játékost.

- `setMapSize(int size)`: Publikus metódus, amely beállítja a térkép méretét, ha a megadott érték érvényes (32 vagy 64).

Láthatóság: public

Leírás: A térkép méretének beállítását végzi, és hibaüzenetet ad, ha a megadott méret érvénytelen.

- `onConfirmSetupClicked()`: Publikus metódus, amely elindítja a játékot a beállított paraméterekkel.

Láthatóság: public

Leírás: A "Confirm Setup" gomb megnyomásakor elindítja a játékot a coordinator segítségével a beállított térkép mérettel, játékosokkal és mentési névvel.

- `onBackToMainMenuClicked()`: Publikus metódus, amely visszatér a főmenübe.

Láthatóság: public

Leírás: A "Back to Main Menu" gomb megnyomásakor visszatér a főmenübe a coordinator segítségével.

- `getPlayers()`: Publikus metódus, amely visszaadja a játékosok listáját.

Láthatóság: public

Leírás: Visszaadja a játékosok listáját. Védekezés céljából, ha szükséges, egy másolatot adhat vissza.

- `getSaveName()`: Publikus metódus, amely visszaadja a mentési nevet.

Láthatóság: public

Leírás: Visszaadja a játék mentési nevét.

- `getMapSize()`: Publikus metódus, amely visszaadja a térkép méretét.

Láthatóság: public

Leírás: Visszaadja a beállított térkép méretét.

- `getCoordinator()`: Publikus metódus, amely visszaadja a GameCoordinator példányt.

Láthatóság: public

Leírás: Visszaadja a játék koordinátort, amely a logikai műveleteket végzi.

10.3.8 CustomButton Osztály

Felelősség

A CustomButton osztály felelős egy testreszabott gomb létrehozásáért és megjelenítéséért, amely képes különböző állapotokat (normál, hover, pressed, disabled) kezelni, valamint képet és szöveget is megjeleníteni. Ezen kívül a gomb interakcióit is képes kezelni, például kattintás és egérmozgás.

Öszoztályok

- **Interactable:** A CustomButton osztály az Interactable osztályból származik, amely a gomb helyét és méretét kezeli, valamint az interakciók alapvető logikáját biztosítja.

Interfészek

- Nincs meghatározott interfész, amit a CustomButton osztály implementál.

Attribútumok

- **text:** A gomb feliratát tárolja, ami a felhasználó számára megjelenik. (láthatóság: private, típus: String)
- **normalColor:** A gomb alap színe, ha nincs semmilyen interakció (láthatóság: private, típus: Color)

- **hoverColor**: A gomb színe, amikor az egér fölötté van (láthatóság: private, típus: Color)
- **pressedColor**: A gomb színe, amikor az egér rákattintott (láthatóság: private, típus: Color)
- **textColor**: A gomb szövegének színe, ha a gomb aktív (láthatóság: private, típus: Color)
- **disabledTextColor**: A gomb szövegének színe, ha a gomb le van tiltva (láthatóság: private, típus: Color)
- **textFont**: A gomb szövegének betűtípusa (láthatóság: private, típus: Font)
- **image**: A gombra beállított képet tárolja (láthatóság: private, típus: BufferedImage)
- **enabled**: A gomb engedélyezett állapotát tárolja (láthatóság: private, típus: boolean)
- **Metódusok**
- **setImage(String imagePath)**: Beállítja a gomb képét a megadott fájl útvonal alapján. (láthatóság: public, típus: void)
 - Leírás: A fájl elérési útját használja a kép beolvasásához, és eltárolja a gomb számára. Ha a fájl nem található, akkor hibaüzenetet ad.
- **setImage(BufferedImage img)**: Beállítja a gomb képét egy már meglévő BufferedImage objektum alapján. (láthatóság: public, típus: void)
 - Leírás: A már betöltött képet állítja be a gomb számára.
- **setEnabled(boolean enabled)**: Beállítja a gomb engedélyezett állapotát. (láthatóság: public, típus: void)
 - Leírás: Ha a gomb engedélyezett, akkor az interakciók lehetségesek, különben nem.
- **isEnabled()**: Ellenőrzi, hogy a gomb engedélyezett-e. (láthatóság: public, típus: boolean)
 - Leírás: Ha a gomb engedélyezett, akkor igaz értéket ad vissza, különben hamisat.
- **draw(Graphics2D g2d)**: A gombot kirajzolja a megadott Graphics2D kontextusra. (láthatóság: public, típus: void)
 - Leírás: A gomb különböző állapotainak megfelelően (normál, hover, pressed, disabled) különböző háttérszínekkel és szövegekkel rajzolja meg a gombot. Ha van kép beállítva, azt is megjeleníti.
- **contains(int mouseX, int mouseY)**: Ellenőrzi, hogy a megadott egérpozíció a gomb területén belül van-e. (láthatóság: public, típus: boolean)
 - Leírás: Ha az egér pozíciója a gomb területén belül van és a gomb engedélyezett, akkor igaz értéket ad vissza, különben hamisat.
- **getText()**: Visszaadja a gomb feliratát. (láthatóság: public, típus: String)
 - Leírás: A gomb feliratát adja vissza, ami a gomb szövege.
- **getBounds()**: Visszaadja a gomb határait. (láthatóság: public, típus: Rectangle)
 - Leírás: A gomb elhelyezkedését és méretét adja vissza egy Rectangle objektum formájában.
- **setBounds(int x, int y, int width, int height)**: Beállítja a gomb terjedelmét és pozícióját. (láthatóság: public, típus: void)
 - Leírás: A gomb helyét és méretét állítja be a megadott értékek alapján.

10.3.9 CustomPlayerList

Felelősség

A CustomPlayerList osztály feladata egy játékosok listájának kirajzolása, ahol minden játékos neve szerkeszthető egy CustomTextField segítségével. Az osztály biztosítja a játékosok típusának megfelelő ikonok megjelenítését, valamint az interakciókhoz szükséges vizuális elemeket. Az ikonok és szövegek a listában való megjelenítését egy adott területen belül szabályozza, figyelembe véve a különböző grafikai elemek közötti elrendezést.

Ősosztályok

- Nincs közvetlen ősosztálya, mivel ez egy segédosztály, amely nem öröklődik más osztályoktól.

Interfészek

- Az osztály nem implementál interfészeket.

Attribútumok

- **FUNGUS_ICON**: A gombás játékos ikonja. Típusa: BufferedImage, statikus, és az osztály betöltésekor kerül inicializálásra.
- **INSECT_ICON**: A rovar típusú játékos ikonja. Típusa: BufferedImage, statikus, és az osztály betöltésekor kerül inicializálásra.
- **ITEM_HEIGHT**: Az egyes elemek (játékosok) magassága a listában. Típusa: int, alapértelmezett értéke 25.
- **PADDING**: A padding (kitöltés) az elemek és a lista között. Típusa: int, alapértelmezett értéke 5.
- **ICON_SIZE**: Az ikonkép mérete. Típusa: int, alapértelmezett értéke 15.
- **ICON_TEXT_GAP**: Az ikon és a szöveg közötti távolság. Típusa: int, alapértelmezett értéke 5.

Metódusok

- **getIconForType(PlayerType type)**:
 - **Leírás**: Visszaadja a megfelelő ikont a játékos típusának megfelelően (FUNGUS vagy INSECT).
 - **Láthatóság**: Publikus, statikus metódus.
- **draw(Graphics2D g2d, List<PlayerInfo> players, List<CustomTextField> playerTextFields, Rectangle bounds)**:
 - **Leírás**: A játékosok listáját rajzolja ki a megadott Graphics2D kontextusra. A játékosok nevei szerkeszthetők, és minden egyes játékoshoz tartozik egy ikon.
 - **Láthatóság**: Publikus, statikus metódus.
- **getPlayerIndexAt(int mouseY, Rectangle bounds, int playerCount)**:
 - **Leírás**: Meghatározza, hogy a kattintott Y koordináta alapján melyik játékosra kattintottak a listában.
 - **Láthatóság**: Publikus, statikus metódus.

10.3.10 CustomTextField

Felelősség

A CustomTextField osztály felelőssége, hogy egy egyedi szövegbeviteli mezőt reprezentáljon, amely képes kezelni a szöveget, a kurzort, a fókuszaltságot és a billentyűzetes bemeneteket. A szövegdoboz figyeli a kurzor villogását, támogatja az Enter billentyűhöz rendelt callback függvényeket, valamint biztosítja a karakterek megfelelő beszúrását, törlését és navigálását a kurzor pozíciójának segítségével.

Ősosztályok

Interactable → CustomTextField

Interfészek

A CustomTextField megvalósítja a következő interfészt:

- **CursorBlinkListener:** A kurzor villogásáért felelős eseménykezelő interfész.

Attribútumok

- **bounds:** A szövegbeviteli mező helyét és méretét meghatározó téglalap (láthatóság: private, típus: Rectangle).
- **text:** A beírt szöveget tároló karakterlánc (láthatóság: private, típus: String).
- **focused:** A mező fókuszált állapotát jelző logikai érték (láthatóság: private, típus: boolean).
- **cursorPosition:** A kurzor aktuális pozíciója a szövegben (láthatóság: private, típus: int).
- **showCursor:** A kurzor láthatóságát szabályozó logikai érték (láthatóság: private, típus: boolean).
- **onEnterCallback:** Az Enter billentyű lenyomásakor meghívott opcionális callback (láthatóság: private, típus: Runnable).
- **blinkTimer:** A kurzor villogásának időzítője (láthatóság: private, típus: Timer).
- **coordinator:** A játék koordinátorát reprezentáló objektum (láthatóság: private, típus: GameCoordinator).
- **cursorBlinkListener:** A kurzor villogásának eseménykezelője (láthatóság: private, típus: CursorBlinkListener).

Metódusok

- **CustomTextField(int x, int y, int width, int height, GameCoordinator coordinator):** Konstruktor, amely inicializálja a szövegbeviteli mezőt a megadott pozícióval, mérettel és koordinátorral.
- **String getText():** A szöveg visszaadása (láthatóság: public).
- **void setText(String text):** A szöveg beállítása, és a kurzor pozíciójának frissítése (láthatóság: public).
- **boolean isFocused():** A mező fókuszáltságának lekérdezése (láthatóság: public).
- **void setFocused(boolean focused):** A mező fókuszáltságának beállítása és a kurzor villogásának kezelése (láthatóság: public).
- **Rectangle getBounds():** A mező elhelyezkedésének és méretének lekérdezése (láthatóság: public).
- **void setBounds(int x, int y, int width, int height):** A mező pozíciójának és méretének frissítése (láthatóság: public).
- **boolean contains(int x, int y):** A mező tartalmazza-e a megadott koordinátákat (láthatóság: public).
- **void setOnEnterCallback(Runnable callback):** Az Enter billentyűhöz tartozó callback függvény beállítása (láthatóság: public).
- **boolean handleKeyPress(KeyEvent e):** A billentyűzetes események kezelése, beleértve a karakterek beírását és navigálást a kurzorral (láthatóság: public).
- **void draw(Graphics2D g2d):** A szövegbeviteli mező kirajzolása, beleértve a szöveget, a kurzort és a mező határait (láthatóság: public).

Ezek a metódusok lehetővé teszik a szövegbeviteli mező dinamikus kezelését, beleértve az események reagálását és a vizuális megjelenítést.

10.3.11 Interactable

Felelősség

Az **Interactable** osztály felelőssége a felhasználói interakciók kezelésére vonatkozik, különösen az egér mutatósávjának és nyomógombjának vizuális visszajelzéseire. Az osztály tartalmazza az interakciós állapotokat, mint például az "hovered" (egérrel fölé vitt) és "pressed" (lenyomott) állapotokat, amelyek lehetővé teszik a felhasználói visszajelzések

megjelenítését. Az osztály emellett biztosítja a vizuális elemek előtti és utáni állapotok kezelését is.

Ősosztályok

Legősebb osztály → Object

Az osztály nem örököl más osztályoktól közvetlenül, ezért az Object osztály a legősebb ősosztály.

Interfészek

Az osztály nem implementál semmilyen interfészt.

Attribútumok

- x: int – A komponens bal felső sarkának vízszintes koordinátája. Láthatóság: protected.
- y: int – A komponens bal felső sarkának függőleges koordinátája. Láthatóság: protected.
- width: int – A komponens szélessége. Láthatóság: protected.
- height: int – A komponens magassága. Láthatóság: protected.
- isHovered: boolean – Az egérrel történő rámutatás állapota. Láthatóság: protected.
- isPressed: boolean – A komponens lenyomott állapotát jelzi. Láthatóság: protected.
- bounds: Rectangle – A komponens körüli téglalap, amely tartalmazza a pozícióját és a méretét. Láthatóság: protected.
- originalColor: Color – A grafikai elem színének mentése a rajzolás előtt. Láthatóság: private.
- originalFont: Font – A grafikai elem betűtípusa a rajzolás előtt. Láthatóság: private.
- originalClip: Shape – A grafikai elem vágási területe a rajzolás előtt. Láthatóság: private.
- originalStroke: Stroke – A grafikai elem körvonalának vastagsága a rajzolás előtt. Láthatóság: private.

Metódusok

- beforeDraw(Graphics2D g2d): void – Ment egyes grafikai beállításokat, mint a szín, font, vágási terület és vonalvastagság a rajzolás előtt, hogy később vissza lehessen állítani őket. Láthatóság: protected.
- afterDraw(Graphics2D g2d): void – Visszaállítja az előzőleg mentett grafikai beállításokat. Láthatóság: protected.
- setPressed(boolean pressed): void – Beállítja a komponens lenyomott állapotát. Láthatóság: public.
- isPressed(): boolean – Visszaadja a komponens lenyomott állapotát. Láthatóság: public.
- setHovered(boolean hovered): void – Beállítja, hogy az egér rá van-e húzva a komponensre. Láthatóság: public.
- isHovered(): boolean – Visszaadja, hogy az egér rá van-e húzva a komponensre. Láthatóság: public.

10.4 Kapcsolat az alkalmazói rendszerrel

[Szekvencia-diagramokon ábrázolni kell a grafikus rendszer működését. Konzisztens kell legyen az előző alfejezetekkel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában. Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon.]

10.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2025.04.30. 18:00	0,5 óra	Fórián Gyárfás Kemecsei Tóth	Megbeszélés: konzultációs információk, feladatok átbeszélése
2025.05.01. 14:00	8 óra	Fórián	Tevékenység: grafika implementálása
2025.05.3. 12:00	2 óra	Kuzmin	Tevékenység: osztálydiagramok elkészítése
2025.05.03. 17:00	3 óra	Kemecsei	Tevékenység: grafikus objektumok felsorolása
2025.05.04. 14:00	4 óra	Gyárfás	Tevékenység: szekvenciadiagrammok elkészítése
2025.05.05. 08:00	1 óra	Kemecsei	Tevékenység: dokumentum formázás, nyomtatás, verziókezelés, csapatkoordináció, grafikus felület ötletelés, tesztelés