**11. Grafikus felület specifikációja**

26 – five\_guys

Konzulens:

Huszerl Gábor

Csapattagok

| Lekli Zsolt | JENKHQ | zsoltlekli@gmail.com |
| --- | --- | --- |
| Fejes Ábel | AF1Y1C | fejesabel@gmail.com |
| Zelch Csaba | LK0617 | zch03po@gmail.com |
| Szolár Soma | E05SC6 | somaszolar1@gmail.com |
| Toronyi Zsombor | S8F7DV | toronyizsombi@gmail.com |

2025. 05. 05.

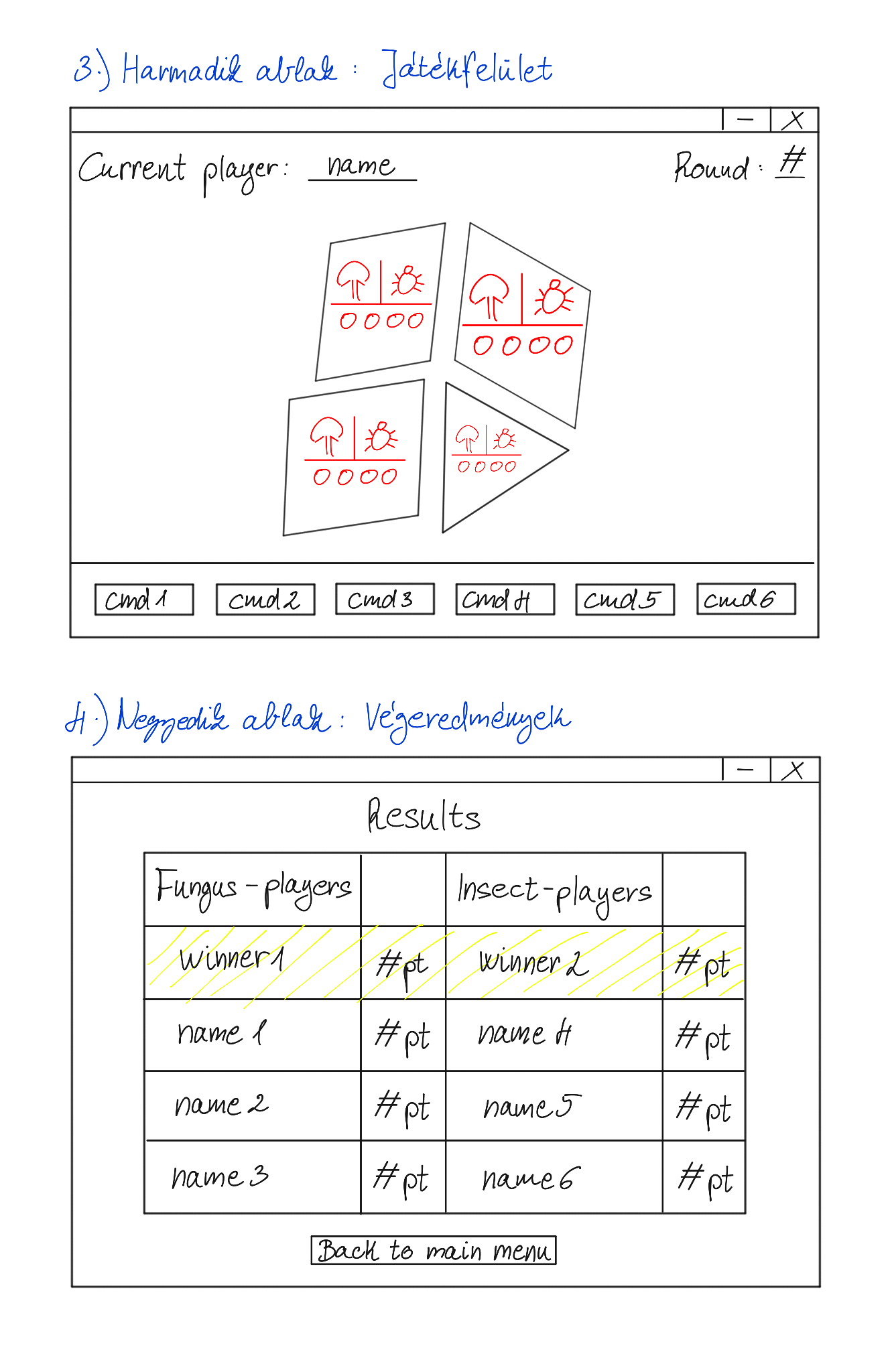
1. Grafikus felület specifikációja

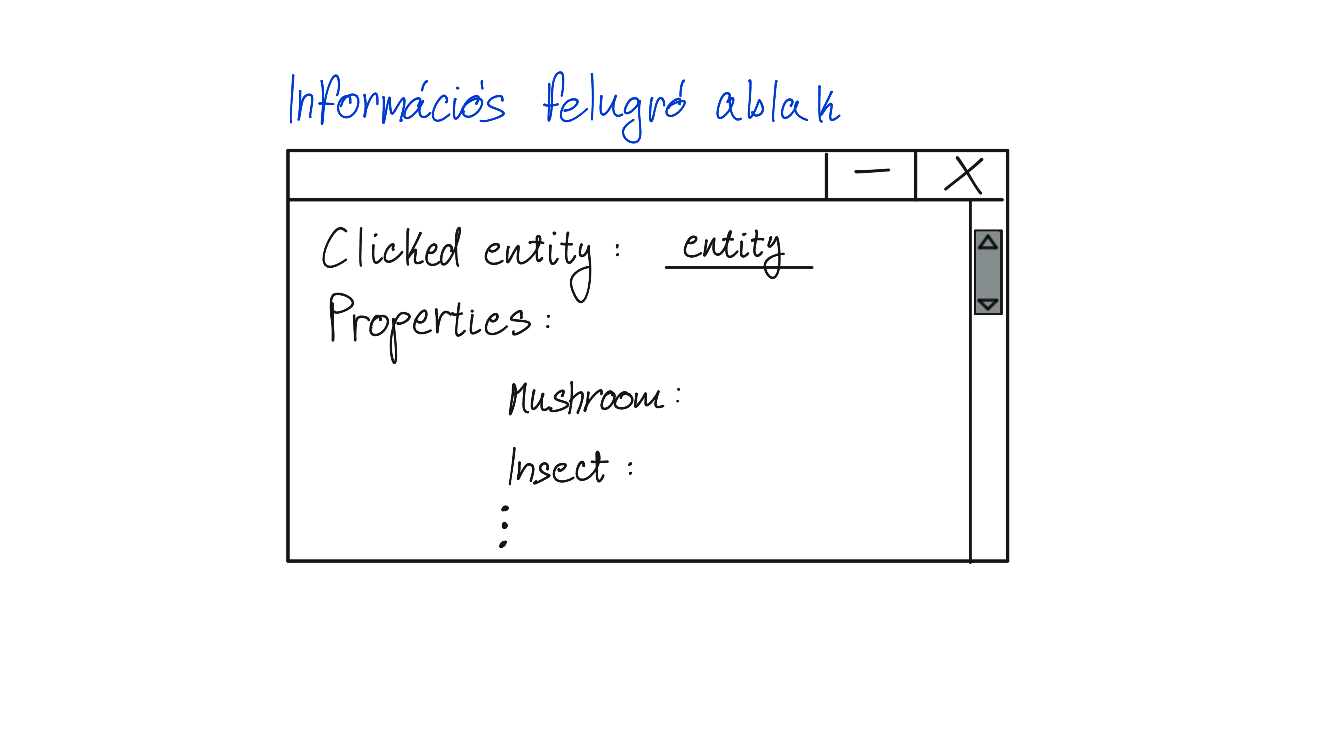
# Változtatások

* Mushroom-nak, Spore-nak, Insect-nek G-osztály, View Interface. A GTecton draw fg-e kirajzolja magát és meghívja a GMushroom, GSpore, GInsect draw fg-eit.
* GTecton-ban nem tároljuk el a szomszédos GTecton-okat.
* GamePanel tartalmaz egy DrawingPanel és ezen lesznek rajta a GTectonok, ő végzi a rajzolásukat

# A grafikus interfész

**

**

**

A grafikus felhasználói felület 5 fő különböző megjelenést támogat.

1. **A játékosok számának megadására szolgáló nézet**

Ez az első megjelenő ablak tartalma a program futtatását követően. Tartalmazza a játék nevét, mint címet. Ezen kívül címkéket, mellettük pedig beviteli mezőket, amelyek a megfelelő játékos típusok számának (maximum 4-4), illetve a játék körei számának felső határának megadására adnak lehetőséget. Bal alsó sarokban pedig található egy gomb, amivel tovább lehet lépni a következő nézetre.

1. **A játékosok neveinek megadására szolgáló nézet**

Ez a nézet az első nézet ‘Next’ gombját követően mutatkozik meg, ha a megadott paraméterek helyesek. Címként látható egy utasítás, ami elárulja, hogy itt a játékosok neveinek megadása fog következni. A cím alatt egy 2 oszlopos és legfeljebb 5 soros rács mintájában lesz látható a két játékos kategória (Fungus-players, Insect-players) és alattuk pedig az előző nézetben megadott számoknak megfelelő mennyiségű beviteli mező, ahova a játékosok neveit várja a program. Továbbá még található egy ‘Back’ és egy ‘Start’ gomb is az alsó sarkokban, amikkel az előző nézetre vissza lehet lépni, vagy pedig elindítani a játékot.

1. **A játékfelület**

Az előző nézet ‘Start’ gombjára kattintva kerül megjelenítésre maga a játékfelület, ha minden mező helyesen lett kitöltve. A játékfelület bal felső sarkában lesz látható az aktuális játékosnak a neve, tehát annak, aki éppen sorra került, jobb felül pedig az aktuális kör sorszáma. Középen lesz látható a térkép, ahol a játék zajlik. Itt lesznek a tektonok (sokszögek), rajta a gombák, rovarok, spórák (ikonok) táblázatos formában és köztük a gombafonalak. Legalul pedig egy sor gomb foglal helyet, melyeken az aktuális játékos típusának megfelelő parancsok szerepelnek majd.

1. **A végeredmény megtekintésére szolgáló nézet**

A játék befejeztével kerülünk erre a nézetre, ahol a cím is megmutatja, hogy a végeredményt fogjuk tudni megtekinteni. Egy 4 oszlopos táblázat fogad minket, az első sora megmutatja melyik oszlopban milyen típusú játékosok nevei és eredményei láthatók. Az első oszlop a Fungus-player játékosok neveit tartalmazza, a második azoknak az elért pontjait, a harmadik pedig az Insect-player játékosok neveit, a negyedik pedig szintén azok pontjait. A játékosok adatai pontszám szerint csökkenően jelennek meg, illetve az egyes kategóriában legtöbb pontszámot elért játékosok adatainak a háttere sárga színnel lesz kiemelve. Illetve a táblázat alatt található egy gomb a ‘Back to main menu’ felirattal, ami visszavisz minket az első nézetre, hogy esetleg egy újabb játéknak foghassunk neki.

1. **Egy információs felugró ablak**

Ez egy olyan felugró ablak, ami a ‘Show info’ gombra kattintva jelenik meg, ha azt megelőzve bal egérgomb lenyomásával kijelölésre került egy entitás (tekton, gomba, rovar, spóra, fonal). Az első sorában láthatjuk a kijelölt entitás típusát, alatta pedig annak jellemzőit. (például tekton esetében a rajta levő gombát, rovart, stb.)

# A grafikus rendszer architektúrája

## A felület működési elve

A felület működési elve kevert. A lépések után, ahol a felület változik, a kontroller értesíti a nézetet a változásról, ezáltal újrarajzolódik a felület. A nézet újrarajzolásakor a GTectonok a modellben levő tektonokról lekérdezik a rajtuk levő objektumokat, és ennek függvényében rajzolják ki a tektonokat.

## A felület osztály-struktúrája

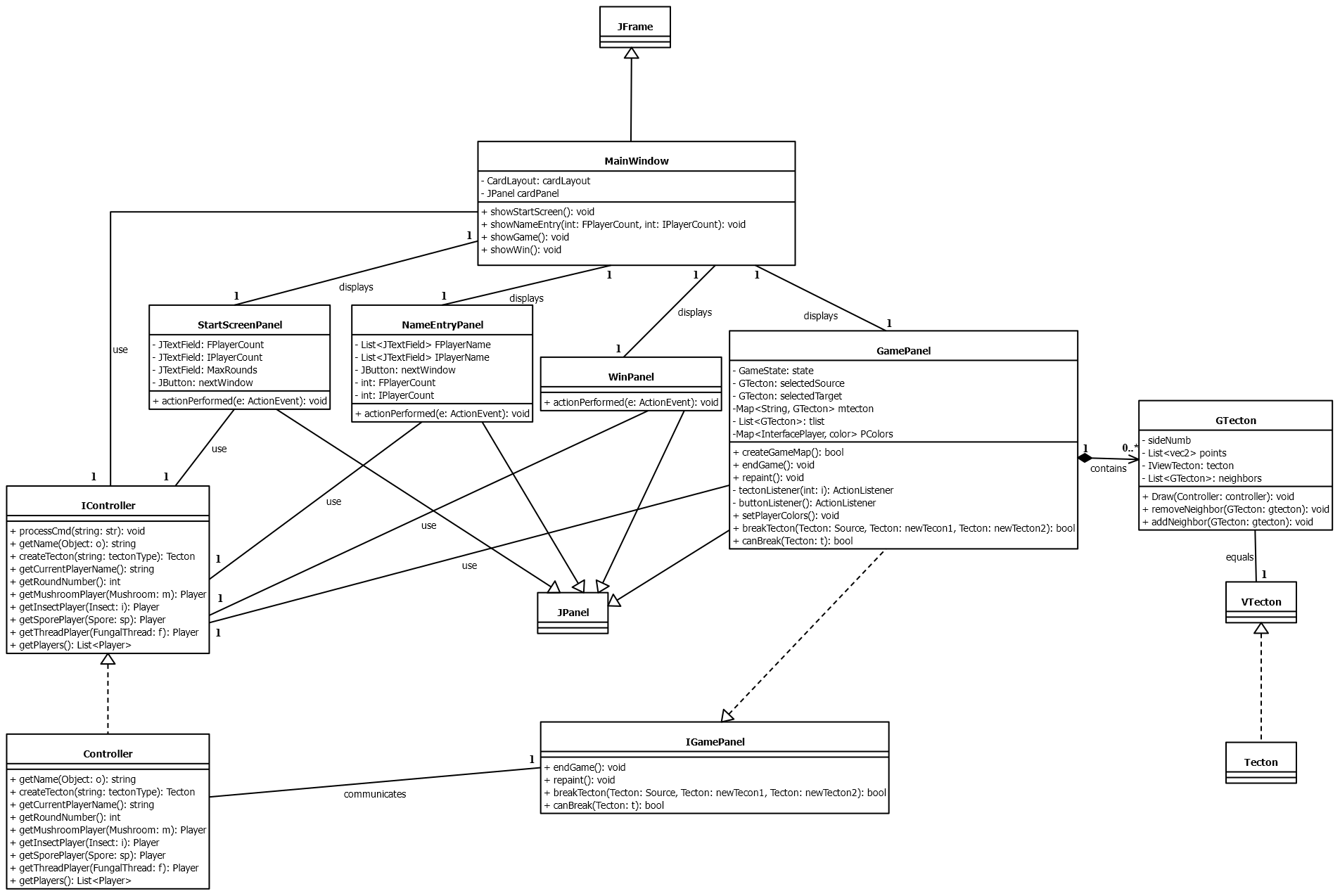
# A grafikus objektumok felsorolása

## IController

* + - * **Felelősség**

A Controller által megvalósított és a View osztályai által használt interfész.

## IGamePanel

A GamePanel által megvalósított és a Controller által használt interfész.

## Controller (változtatások)

Megvalósítja az IController interfészt. Tartalmazza az IGamePanelt.

* + - * **Új interfészek**

IController

* + - * **Új attribútumok**
* **- IGamePanel gamePanel**
  + - * **Új megvalósított metódusok**
* **+ String getName(Object o):** Kikeresi a controller objects map-jéből a paraméterül kapott o-hoz a nevét és visszaadja azt.
* **+ Tecton createTecton(String tectonType):** Létrehozz egy tektont a paraméterül kapott tektontípus alapján és visszaadja azt.
* **+ String getCurrentPlayerName():** Visszaadja a currentPlayer nevét (kikeresi az objects mapből).
* **+ int getRoundNumber():** Visszaadja hogy hányadik körben jár a játék.
* **+ Player getMushroomPlayer(Mushroom: m)**: Visszaadja a paraméterül kapott gombához tartozó játékost.
* **+ Player getInsectPlayer(Insect i):** Visszaadja a paraméterül kapott rovarhoz tartozó játékost.
* **+ Player getThreadPlayer(FungalThread f):** Visszaadja a paraméterül kapott fonálhoz tartozó játékost.
* **+ Player getSporePlayer(Spore s):** Visszaadja a paraméterül kapott spórához tartozó játékost.
* **+ List<Player> getPlayers():** Visszaadja az összes játékost (gombászokat, rovarászokat).

## JFrame

Java Swing-beli osztály.

## JPanel

Java Swing-beli osztály.

## MainWindow

* + - * **Felelősség**

A MainWindow osztály felelős a játék megjelenítéséért, CardLayout-ot használ a játék további képernyői közötti váltásra. A JFrame osztályból származik le. Ezzel a megoldással csak egyetlen JFrame kell, nem kell minden képernyőváltáskor újat létrehozni.

* + - * **Ősosztályok**

JFrame -> MainWindow

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- CardLayout cardLayout**: Java layout típus, segítségével könnyen váltogathatunk a képernyők között.
* **- JPanel cardPanel:** A megfelelő játék képernyők közötti váltogatásra szolgál.
* - **StartScreenPanel startScreenPanel:** A kezdő képernyő megjelenítéséért.
* **- NameEntryPanel nameEntryPanel:** A nevek megadásáért.
* **- WinPanel winPanel:** A győztesek megjelenítéséért.
* **- Controller controller:** A Controllerrel való kommunikálásért.
  + - * **Metódusok**
* **+ void showStartScreen()**: A kezdőképernyő megjelenítéséért felelős metódus
* **+ void showNameEntry( int FPlayerCount, int IPlayerCount ):** Annak a képernyőnek megjelenítéséért felelős, amelyiknél meg kell adni a játékosok nevét, annyi írható szövegdoboz fog megjelenni, amennyit paraméterként bekap rovarászra illetve gombászra.
* **+ void showGame():** A játékképernyő megjelenítése, itt fog történni a játél érdemi része, itt láthatóak majd a tektonok, stb.
* **+ void showWin()**: A játék végén megjelenő képernyő megjelenítése, ranglista látható rajta a játékosokról.

## StartScreenPanel

* + - * **Felelősség**

Ez az osztály felelős a kezdőképernyő megjelenítéséért, a rajta található 3 írható szövegdobozban megadott paraméterek alapján beállítódnak a játék megfelelő paraméterei: a gombászok száma, a rovarászok száma és a maximális körök száma.

* + - * **Ősosztályok**

JPanel -> StartScreenPanel

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- JTextField FPlayerCount**: írható szövegdoboz, itt kell megadni hogy mennyi gombász lesz a játékban.
* **- JTextField IPlayerCount**: írható szövegdoboz, itt kell megadni, hogy mennyi rovarász lesz a játékban.
* **- JTextField MaxRounds:** írható szövegdoboz, itt kell megadni, hogy maximum mennyi kör legyen a játékban.
* **- JButton: nextWindow:** Gomb, a következő ablakra váltáshoz.
* **- MainWindow: mainWindow:** Lehessen továbblépni a NameEntryPanel-re.
  + - * **Metódusok**
* **+ void actionPerformed(ActionEvent e)**: A panel eseménykezelője.

## NameEntryPanel

* + - * **Felelősség**

Ez az osztály felelős a játékos nevek megadásához szükséges ablakért, rajta található annyi írható szövegdoboz, amennyit a StartScreenPanel-en megadtunk. Az itt megadott nevek lesznek a játékosok nevei a játék során.

* + - * **Ősosztályok**

JPanel -> NameEntryPanel

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- List<JTextField> FPlayerName:** A Gombászok neveinek megadásához szükséges írható szövegdoboz lista. Annyi szövegdoboz található benne, ahány gombász van.
* **- List<JTextField> FPlayerName:** A Rovarászok neveinek megadásához szükséges írható szövegdoboz lista. Annyi szövegdoboz található benne, ahány rovarász van.
* **- JButton nextWindow:** A következő ablakra váltás gombja.
* **- MainWindow: mainWindow:** Lehessen továbblépni a GamePanel-re.
  + - * **Metódusok**
* **+ void actionPerformed(ActionEvent e):** A panel eseménykezelője.

## WinPanel

* + - * **Felelősség**

Ez az osztály felelős a játék végén kiírandó eredmény tábla grafikus megjelenítéséért. A játékosok nevét és a hozzájuk tartozó pontot írja ki.

* + - * **Ősosztályok**

JPanel -> WinPanel

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- MainWindow: mainWindow:**  Hogy át lehessen menni a StartScreenPanel-re, kezdődik előlröl a folyamat.
  + - * **Metódusok**
* **+ void actionPerformed(ActionEvent e):** A panel eseménykezelője.

## GamePanel

* + - * **Felelősség**

Ez az osztály felelős a játékablakért, ahol a játékpálya és a játékos lépésfajtájának kiválasztásához tartozó gombok találhatóak. A játékpályán helyezkednek el a tektonok és a rajtuk lévő modellbeli objektumok. A játékosokhoz tartozó lépésfajták (amikhez a gombok tartoznak) a következők:

* Első kör esetén:
  + Gombász: Első gombatest lehelyezése
  + Rovarász: Első rovar lehelyezése
* Egyébként:
  + Gombász: Spóraszórás, fonál elágaztatás
  + Rovarász: Rovar léptetése, fonál vágása

Mindegyik játékosnak van továbbá egy olyan gombja, amellyel a körét zárja és ezzel jelzi, hogy jöhet a következő játékos. Ezek a gombok csak akkor jelennek meg ha a játékos a szabályok szerint azt a lépésfajtát a körében elvégezheti. Miután kiválasztja a játékos a lépésfajtáját, azután a lépés során kiválasztandó objektumot a megfelelő tektonra kattintva a játékpályán választhatja ki.

* + - * **Ősosztályok**

Jpanel -> GamePanel

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- GameState state:** A játék állapotát tárolja: milyen lépésfajta van kiválasztva az adott körben az adott játékos által.
* - **GTecton selectedSource**: Ha olyan lépésfajta van érvényben, ahol ki kell választani egy forrás és egy cél tektont, abban az esetben ez a forrás tekton. (Pl.: Ha az adott gombász játékos kiválasztott lépésfajtája spóraszórás, akkor ez lesz a forrás tekton, amin a kilövést végrehajtó gombatest rajta van.) Egyébként null.
* - **GTecton selectedTarget**: Ha olyan lépésfajta van érvényben, ahol ki kell választani egy forrás és egy cél tektont, abban az esetben ez a cél tekton. (Pl.: Ha az adott gombász játékos kiválasztott lépésfajtája spóraszórás, akkor ez lesz a cél tekton, amire a kilövést végrehajtó gombatest a spórát lövi.) Egyébként null.ű
* **- Map<String, GTecton> mtecton:** Olyan map, ami minden GTecton objektumhoz tárolja a hozzá tartozó egyedi nevet.
* **- List<GTecton>: tlist:** A játékpályát felépítő GTecton objektumokat tárolja.
* **- Map<InterfacePlayer, color> PColors:** A játékban résztvevő játékosokhoz tárolja a színűket. A játékosnak a tektonon lévő objektumai ezzel a színnel fognak megjelenni.
  + - * **Metódusok**
* **+ boolean createGameMap():** Létrehozza a modellben a játékpályát felépítő tektonokat és a hozzájuk tartozó View-beli GTectonokat és ezeket eltárolja a tlist-ben és az mtecton-ban. Az mtecton-ban a kulcs értékeket a controllertől kéri el. Tehát a controller-beli tektonnév lesz az mtectonban a grafikus GTectonhoz hozzárendelve. Visszatérési értéke igaz ha sikerült létrehoznia mindent megfelelően, egyébként hamis.
* + **void endGame():** Befejezi a játékot és hatására megjelenik az az ablak, ami kiírja a végeredményt (játékosok és pontjaik).
* **+ void repaint():** Újra rajzolja a játékpályát a tlist-ben tárolt GTecton-ok alapján.
* **- ActionListener tectonListener(int i):** A tektonra való kattintás esetén hívódik meg és a state attribútumnak megfelelően végrehajtódik a kívánt esemény a state-nek megfelelően.
* - **ActionListener buttonListener(int i):** A lépésfajt ákat ábrázoló gomb kattintása esetén hívódik meg, beállítja a gombnak megfelelően a state attribútumot.
* **+ void setPlayerColors():** Beállítja a játékosok színét és eltárolja a PColors map-be. Két játékosnak nem lehet ugyanaz a színe.
* + **boolean breakTecton(Tecton: Source, Tecton: newTecon1, Tecton: newTecton2):** Miután a modellben megvalósult egy tekton ketté törése azután hívódik meg ez a függvény. Paraméterül a ketté tört tektont (Source) és a két újonnan keletkezett tektont (newTecton1, newTecton2) kapja. A törést a Viewban is elvégzi. Létrehoz a két új tektonnak egy-egy GTectont, aminek a paramétereit a Source tekton GTectonja utáni törésnek megfelelően állítja be (points, sideNum, neighbors, tecton). Működését az *Inicializálás\_lefutása\_4\_el\_osztható\_kör\_esetén\_ekkor\_egy\_tekton\_kettétörik* szekvencia jól bemutatja.
* **+ boolean canBreak(Tecton t):** Megmondja a paraméterül kapott tektonról, hogy a viewbeli GTecton-ja törhető-e, tehát több mint 3 oldala van e (sideNumb > 3).

## GTecton

* + - * **Felelősség**

A GTecton a játék mezőjeinek grafikus megjelenítéséért és kezeléséért felelős. Tárolja a mező poligon‐alakját (oldalak száma és csúcspontok), a mezőhöz tartozó nézet‐interfészt, valamint a szomszédos mezők referenciáit. Feladata továbbá, hogy a játékpanel (GamePanel) hívására a saját alakzatát és a rajta levő objektumokat kirajzolja.

* + - * **Ősosztályok**

Nincs őse.

* + - * **Interfészek**

Nem valósít meg interfészt.

* + - * **Attribútumok**
* **- sideNumb : int -** hány oldala van**.**
* **- points : List<Vector<float>> -** tárolja a pontjainak x, y koordinátáit.
* **- tecton : IViewTecton** - tárolja, hogy a Controllerben milyen tecton felel meg neki, ennek a tectonnak meg kell valósítania a IViewTecton interfészt.
* **- neighbors : List<GTecton>** - tárolja a szomszédait.
  + - * **Metódusok**
* **- draw() : void** - felelős a mező és azon lévő entitások (gombatest, rovar, spórák, fonál) grafikus megjelenítéséért a játéktérben.

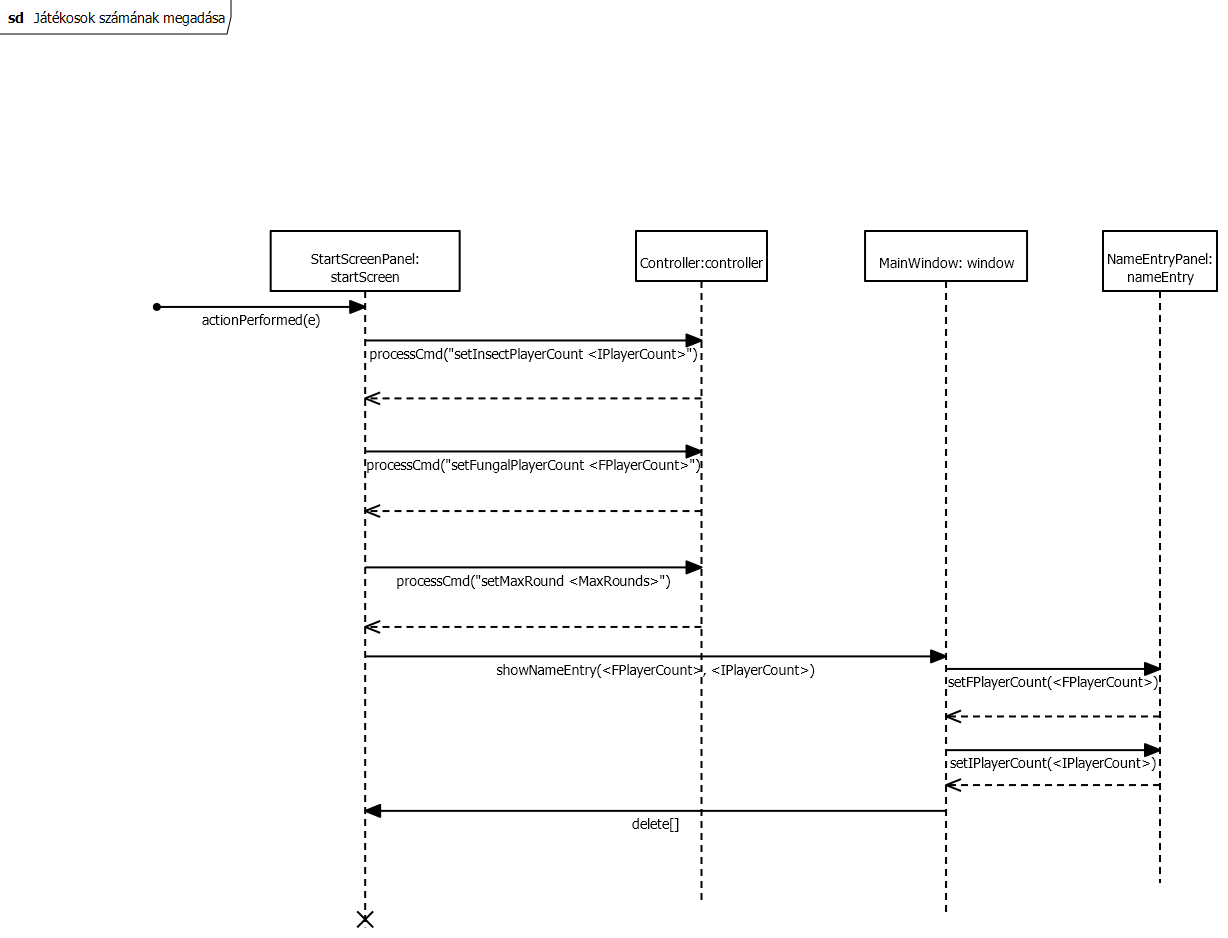
Részletes feladatok sorrendben

1. Poligon megrajzolása
   * A GTecton saját koordinátáiból felépíti a poligon alakú mezőt.
   * Kitöltéssel és körvonallal megjeleníti a mezőt.
2. Gombatest kezelése
   * Lekérdezi a mezőhöz tartozó Mushroom objektumot (tecton.getMushroom()).
   * Ha létezik gombatest:
     1. Lekéri a gombatest aktuális állapotát a modellből.
     2. A vezérlőtől meghívja a mezőhöz rendelt játékost (controller.getMushroomPlayer(m)), hogy megkapja a kirajzoláshoz szükséges színt.
     3. A gombatestet a tecton közepén levő táblázat bal felső sarkába kirajzolja egy olyan ikonként, ami függ a játékos színétől, és a lekért típústól.
3. Rovar kezelése
   * Lekérdezi, hogy van-e az adott mezőn Insect.
   * Ha igen, ugyanúgy lekéri a hozzá tartozó játékost (controller.getInsectPlayer(i)) és egy kis „rovar-ikon” kirajzolásával jeleníti meg. (függ a játékos színétól) (jobb felső sarokba kerül)
4. Spórák kezelése
   * A mező getSpores() metódusával megszerzi a spórákat, ezeken végigmegy.
   * Minden spóra esetén lekéri a spóra játékosát (controller.getSporePlayer(s)) és kis körökként vagy ikonként jeleníti meg az alsó sorban. Ha egy játékosnak van több spórája a tectonon, az akkor is csak egy körként jelenik meg.
5. Gombafonalak (Threads) kirajzolása
   * Lekérdezi a mezőn lévő gombafonálokat (getThreads()).
   * Minden fonál esetén lekéri a megfelelő játékost (controller.getThreadPlayer(t)).
   * Megvizsgálja az összes, e mezőhöz szomszédos GTecton-t (neighbors): ha van közös, ugyanolyan gombafajhoz tartozó fonál, akkor a két mező közé, a játékos színével, egy vékony vonallal ábrázolja az elágazást.

* **- removeNeighbor(g : GTecton) : void**
* **- addNeighbor(g : GTecton) : void**

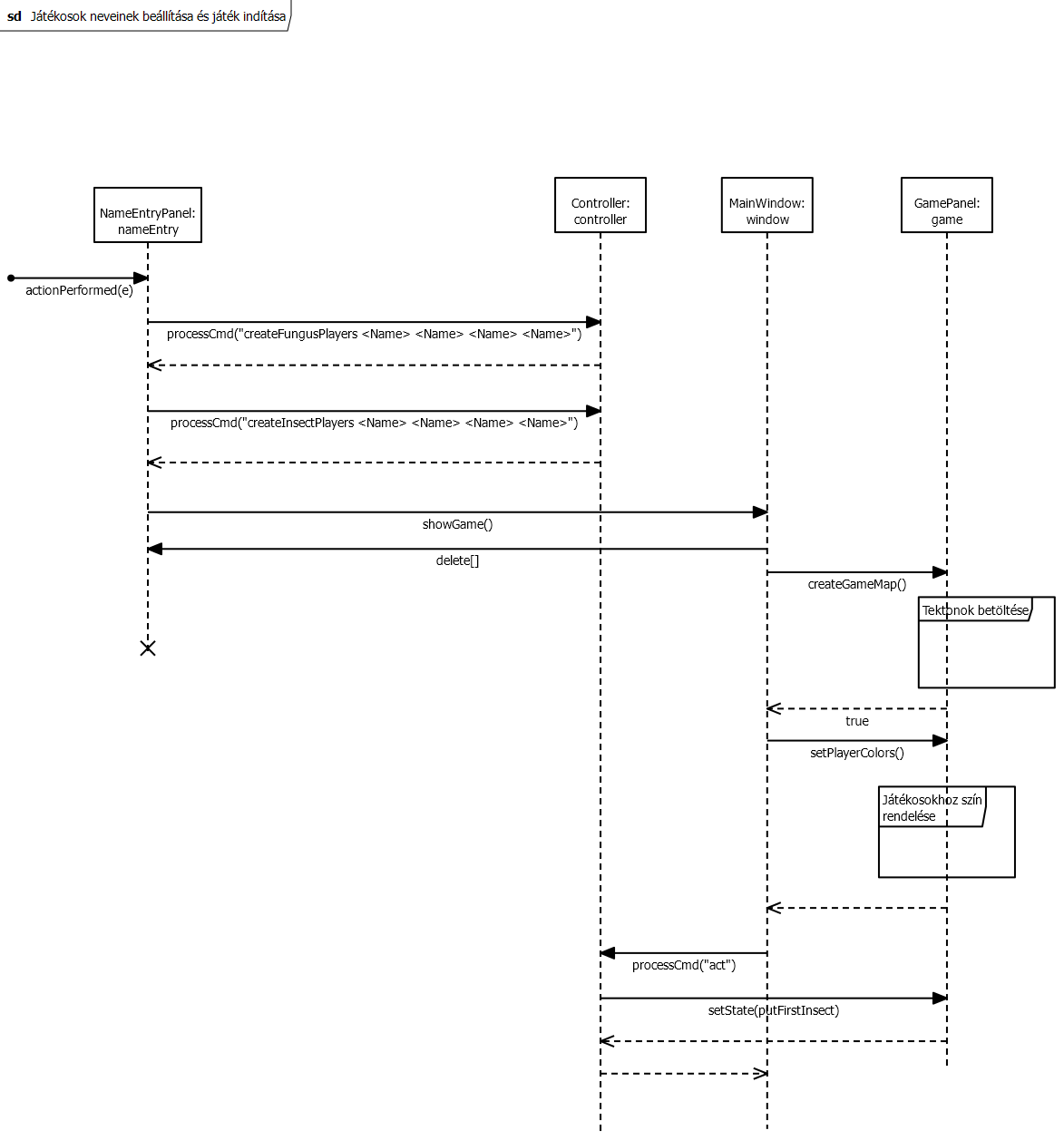
# Kapcsolat az alkalmazói rendszerrel

## Játékosok\_számának\_megadása

Hívás: rákattintunk a képernyőn levő nextWindow gombra

A szekvencia diagram azt ábrázolja, mikor az a képernyő van megjelenítve, amellyel beállíthatjuk a gombászok és a rovarászok számát, valamint a maximális körök számát. Miután a JTextFieldekbe beírtuk az adatokat rákattinthatunk a next gombra, ennek hatására meghívódik az actionPerformed függvény, ami beállítja a kontrollerben a gombászok és rovarászok számát, valamint a maximális körszámot, ezután meghívja a window showNameEntry metódusát, egy NameEntryPanel típusú JPanelt fog megjeleníteni az ablakban.

## Játékosok\_neveinek\_beállítása\_és\_játék\_indítása

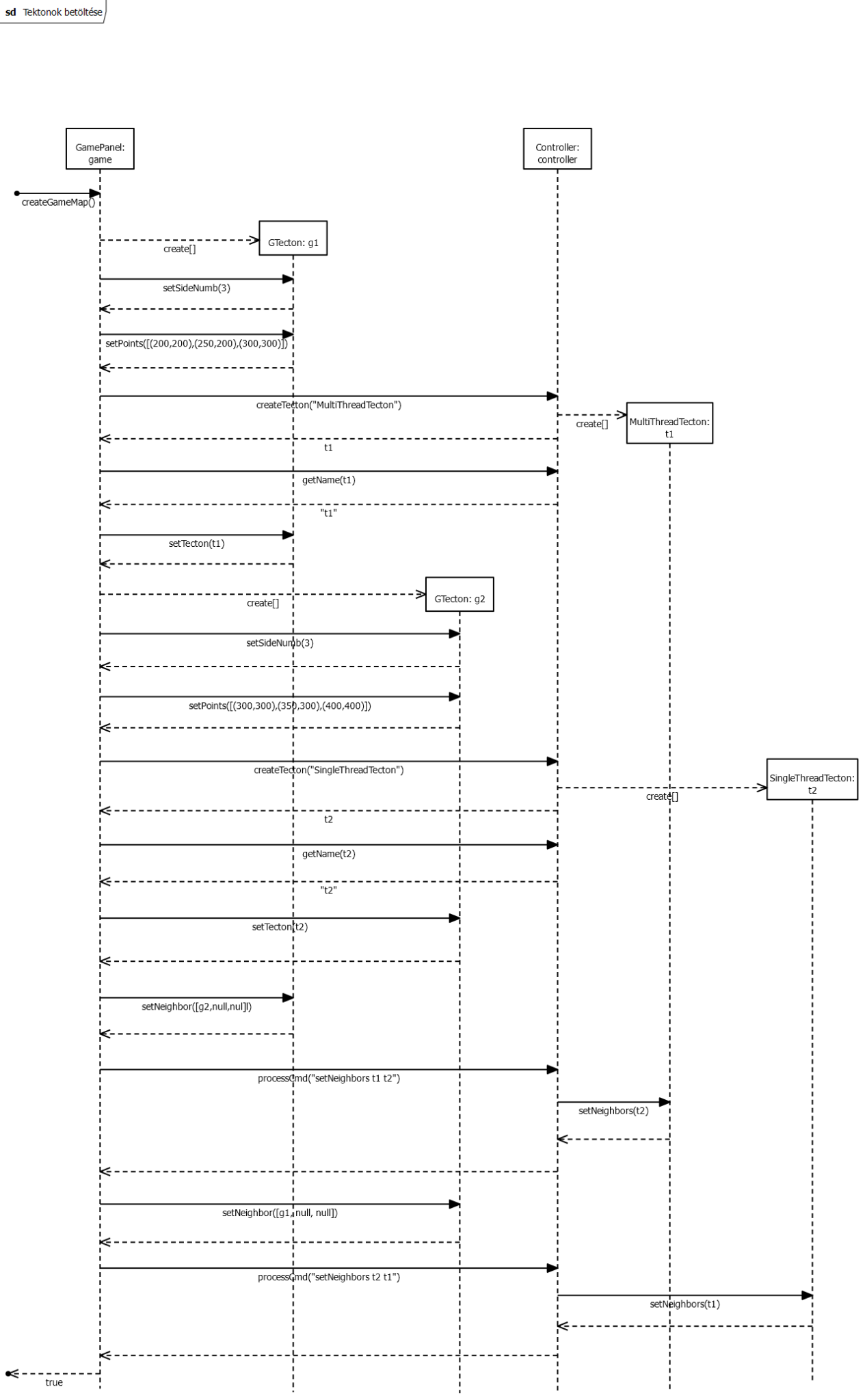
**

Hívás: rákattintunk a képernyőn levő nextWindow gombra

A szekvencia diagram azt ábrázolja, mikor a játékosok neveit adja meg. Ekkor a JTextFieldekbe beírhatjuk a neveket, miután ezt megtettük, rákattinthatunk a start gombra, ennek hatására meghívódik az actionPerformed metódus, ami beállítja a játékosok neveit a kontrollerben, és meghívja a MainWindow showGame metódusát, ezáltal betöltődnek a txt fájlból a tektonok, beállítódnak a játékosok színei, és elkezdődik a játék.

A Tektonokat a txt fájlokban a következőképp tároljuk: eltároljuk, hogy hány oldaluk van, eltároljuk minden pontjának a koordinátáját, majd miután az összes tektont felsoroltuk, eltároljuk a szomszédságokat a kontrollernél megismert módszer segítségével, azzal a különbséggel, hogy minden oldalra eltároljuk a szomszédot, és amely oldal nem szomszédos egyetlen tektonnal sem, ott a szomszéd értéke null.

## **Tektonok\_betöltése**

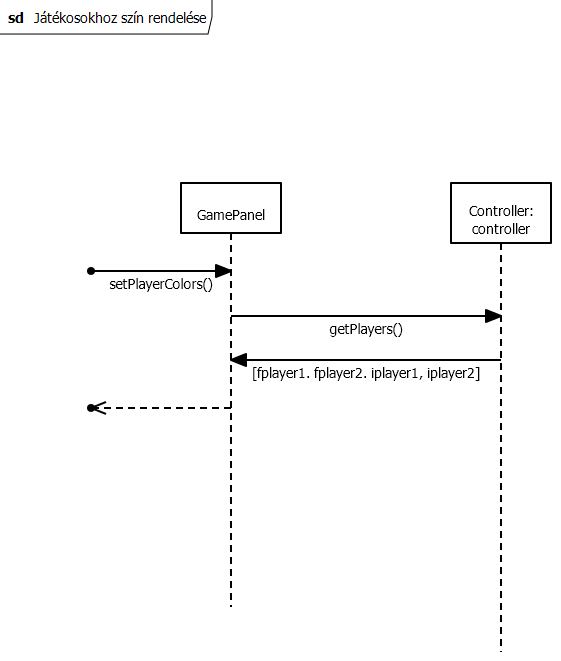


A MainWindow meghívja a GamePanel createGameMap() metódusát.

Ennek során sorra olvassuk be a tektonokat a txt fájlból, és a beolvasott adatoknak megfelelően létrehozunk GTectonokat, melyeknek a paramétereit a beolvasott adatoknak megfelelően beállítjuk. Ezután a kontroller createTecton függvéynét meghívva, létrehozunk a grafikus tektonnak megfelelő tektont, amit visszaadunk a GamePanelnek, ezáltal a GTektonnak beállíthatjuk az attribútumát. Lekérjük az adott tekton nevét is string-ként, és eltároljuk egy Map-ben a név GTekton összerendelést, erre majd szükségünk lesz a szomszédságok beállításánál.

Miután minden tektont beolvastunk, a szomszédságokat is be kell állítanunk, a text fájlban a kontrollernél megísmert módon vannak a szomszédságok eltárolva, azzal a különbséggel, hogy minden oldalra eltároljuk a szomszédot, és amely oldal nem szomszédos egyetlen tektonnal sem, ott a szomszéd értéke null. A Map összerendelés segítségével beállítjuk a GTektonoknál a szomszédságokat, majd a megfelelő parancsokat kiadva a modellben is beállítjuk a szomszédságokat.

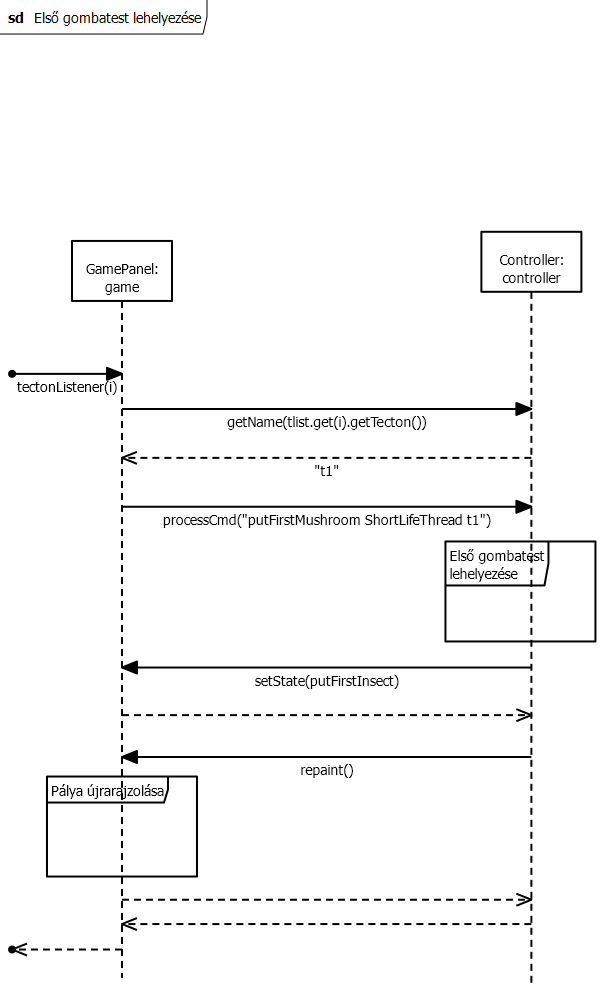
## Játékosokhoz\_szín\_rendelése



A MainWindow meghívja a GamePanel setPlayerColors metódusát

A metódus során a GamePanel lekérdi a kontrollertől az összes játékost, majd belehelyezi egy Map-be, egy színt rendelve hozzájuk úgy, hogy két különböző játékos ugyanazt a színt ne kapja. Ez majd fontos lesz, mivel egy játékos minden objektumát a tektonokon ilyen színnel színezünk ki.

## Első\_gombatest\_lehelyezése

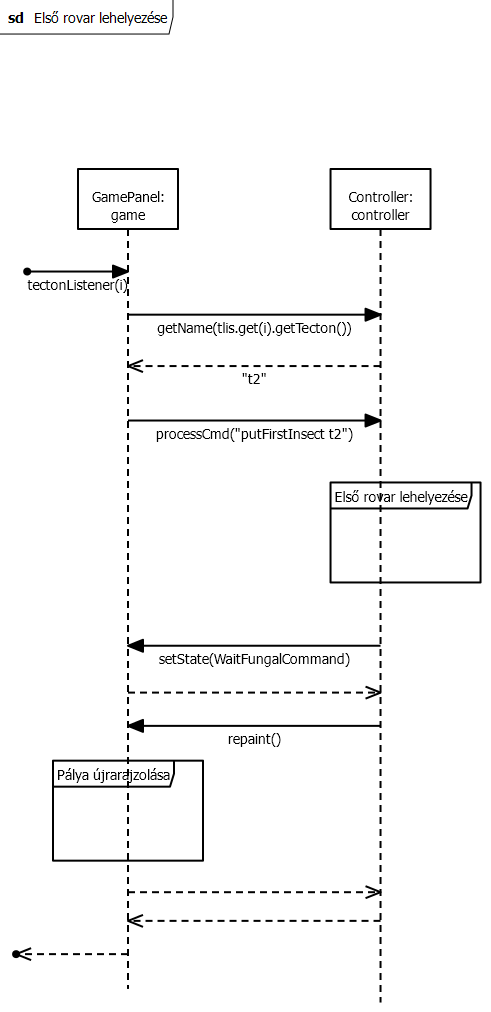


Hívás: Rákattintottunk egy tektonra.

A GamePanel putFirstMushroom állapotban van, azaz vár, hogy az adott gombász helyezze le az első gombatestjét.

Miután a gombész kiválasztotta a tektont és rákattintott, meghívódik a tectonListener metódus, amely paraméterként megkapja, hogy a kattintott tekton hányadik a tlist listában. Lekérjük a kontrollertől a tekton nevét, majd létrehozzuk a parancsot, amellyel lehelyezhetjük az első gombatestet, és odaadjuk ezt a kontrollernek. Ennek során megtörténik az első gombatest lehelyezése, a program tovább lép a következő játékosra, amely egy rovarász, így a GamePanel állapota putFirstInsect-be vált, azaz várja, hogy a rovarász az első rovarát lehelyezze, valamint a repaint() metódus meghívása során újrarajzolódik az ablak.

## Első\_rovar\_lehelyezése

**

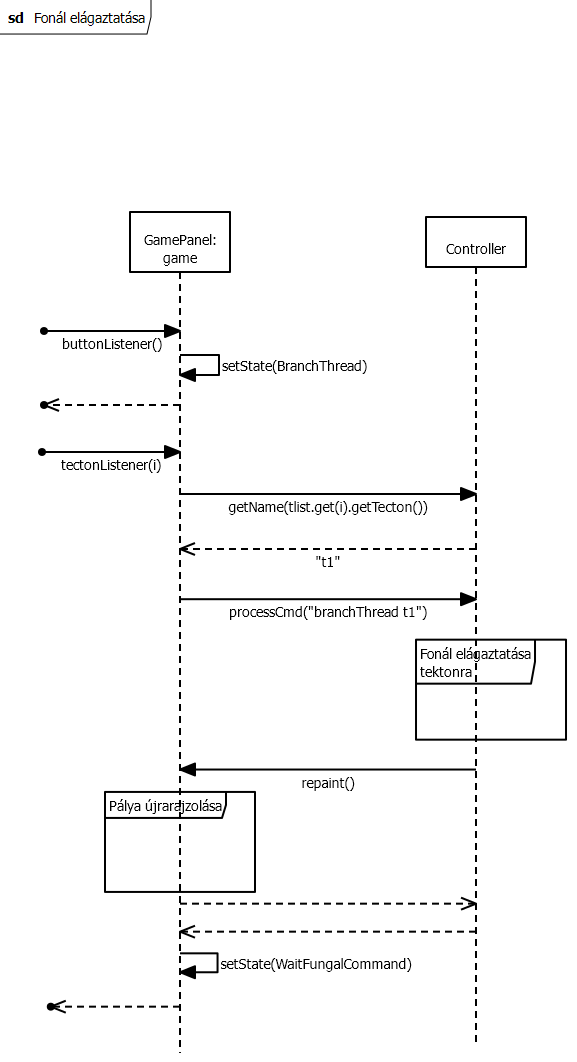
Hívás: Rákattintottunk egy tektonra.

A GamePanel putFirstInsect állapotban van, azaz vár, hogy az adott rovarász helyezze le az első rovarát..

A tektonra kattintás után meghívódik a tectonListener, amely paraméterként megkapja a tekton helyét a tlist tömbben, majd ennek alapján előállítja azt a parancsot, amellyel a rovarász lehelyezheti az első rovarát a modellben.

Miután megtörtént a rovar lehelyezése, mivel több rovarász nincs, ezért új kör következik, lefut az új körnek az inicializálása. Megkezdjük tehát az első kört, a gombászok kezdenek, így a GamePanel a WaitFungalCommand állapotba kerül. Újrarajzoljuk a pályát is.

## Fonál\_elágaztatása



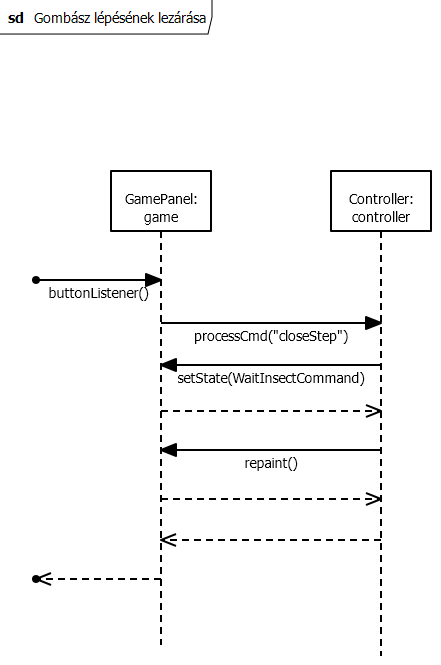
Hívás: rákattintottunk a fonál ágaztatása gombra

A GamePanel a WaitFungalCommand állapotban van, azaz várja, hogy a gombász válassza ki, hogy milyen lépést szeretne végrehajtani.

A gombász a fonál elágaztatása gombra kattintott, ennek hatására a buttonListener fog meghívódni, amely beállítja a GamePanel állepotát BranchThread-re, azaz várni fog hogy kattintsunk egy tektonra, amelyre a gombafonalat elágaztathatjuk.

Miután rákattintottunk a tektonra, meghívódik a tectonListener, átadva paraméterként, hogy a tlist listában hányadik az a tekton, amelyre kattintottunk. Létrehozzuk ez alapján a parancsot, majd kiadjuk a kotnrollernek. Megtörténik a modellben a gombafonál ekágaztatása, majd újrarajzoljuk az ablakot. A GamePanel újra a WaitFunglaCommand állapotba kerül, mivel a gombász még nem zárta le a lépését, így várja, hogy a következő lépések közül válasszon.

## Gombász\_lépésének\_lezárása

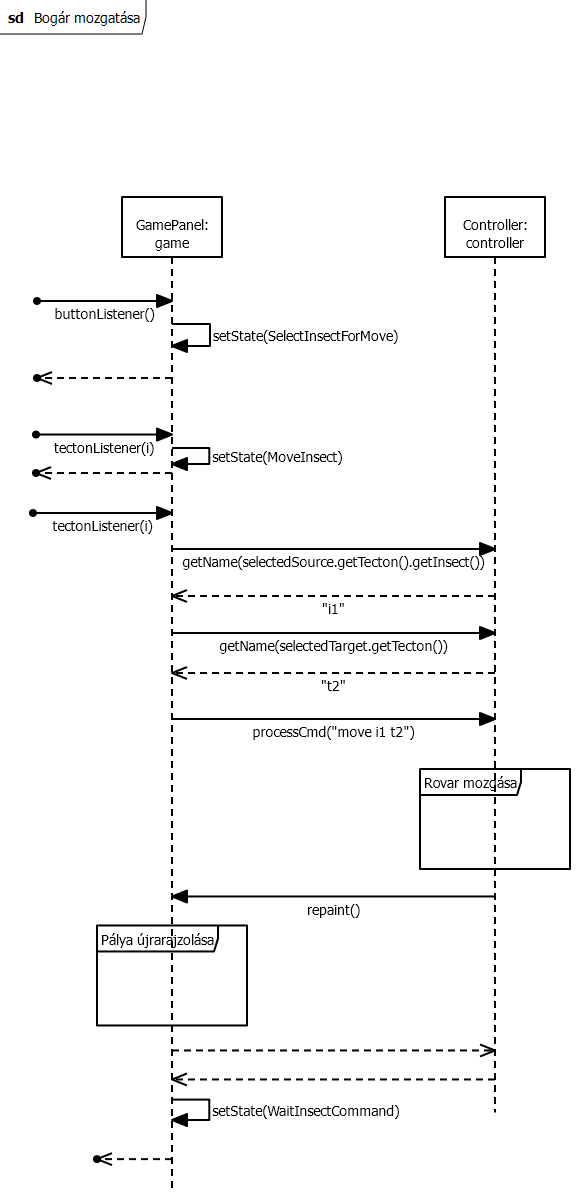
**

Hívás: rákattintunk a lépés befejezése gombra

A GamePanel a WaitFungalCommand állapotban van, azaz várja, hogy a gombász válassza ki, hogy milyen lépést szeretne végrehajtani.

A gombász a lépés lezárása gombra kattintott, ezáltal kiadjuk a kontrollernek a closeStep parancsot. Mivel egy rovarász következik, így nem történik körváltás, beállítjuk a GamePanel állapotát WaitInsectCommand-ra, majd úrjarajzoljuk az ablakot.

## **Bogár\_mozgatása**

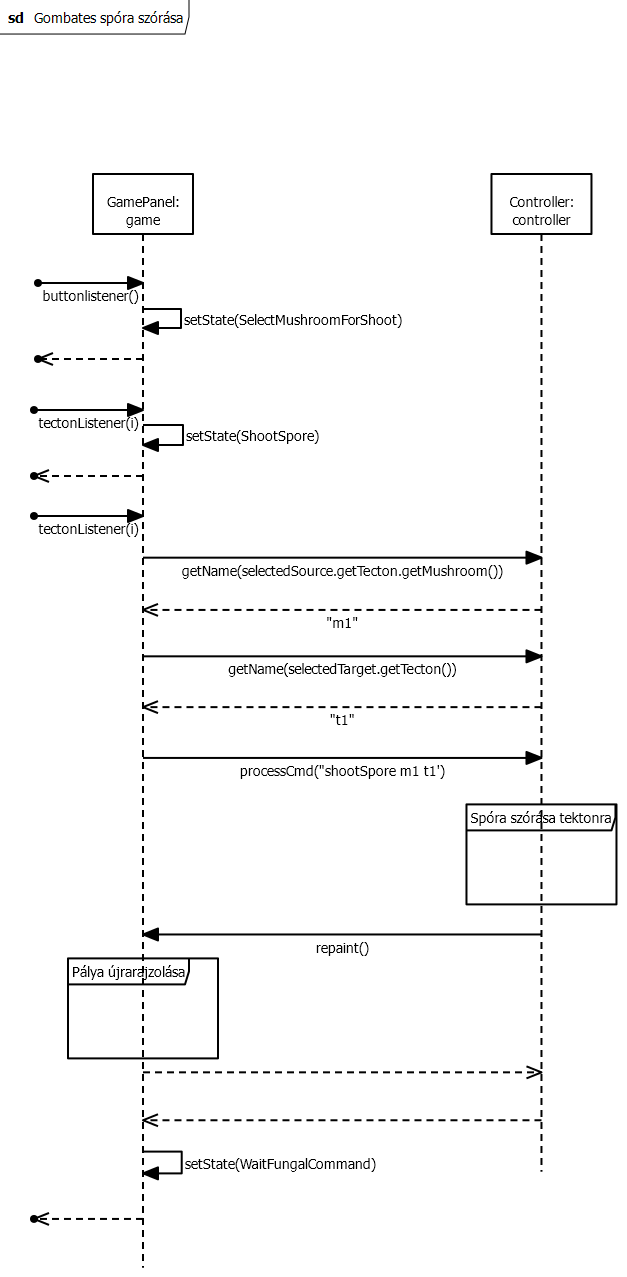


Hívás: A rovarász rákattint a bogár mozgatása gomra

A GamePanel kezdeti állapota WaitInsectCommand, azaz vár, hogy a rovarász válasszon ki egyet a lépései közül.

A rovarász a rovar mozgatása gombra kattint, ennek hatására az állapot átvált SelectInsectFormove-ra, azaz vájuk, hogy a rovarász jelölje ki azt a rovart, amellyel szeretne lépni. Ezután a rovarász egy tektonra kattint. Ekkor A GamePanel átkerül a MoveInsect állapotba, tehát azt a tektont várja, amelyre a rovarat mozgatni akarjuk, illetve a selectedSource változónak átadjuk a kijelölt tektont. A rovarász újra kijelöl egy tektont, a selectedTarget változőnak átadjuk a kijelölt tektont, majd ennek és a selectedSourcenak megfelelően létrehozzuk a paracsot, amit kiadunk a kontrollernek. Megtörténik a rovar mozgása, ezután újrarajzoljuk az ablakot. Mivel a rovarász még nem zárta le a lépését, így a GamePanel a WaitInsectCommand állapotba kerül, azaz vár, hogy a rovaráász kiválasszon egy lépést.

## **Gombatest\_spóra szórása**

**

Hívás: A gombász rákkatint a spóra szórása gombra.

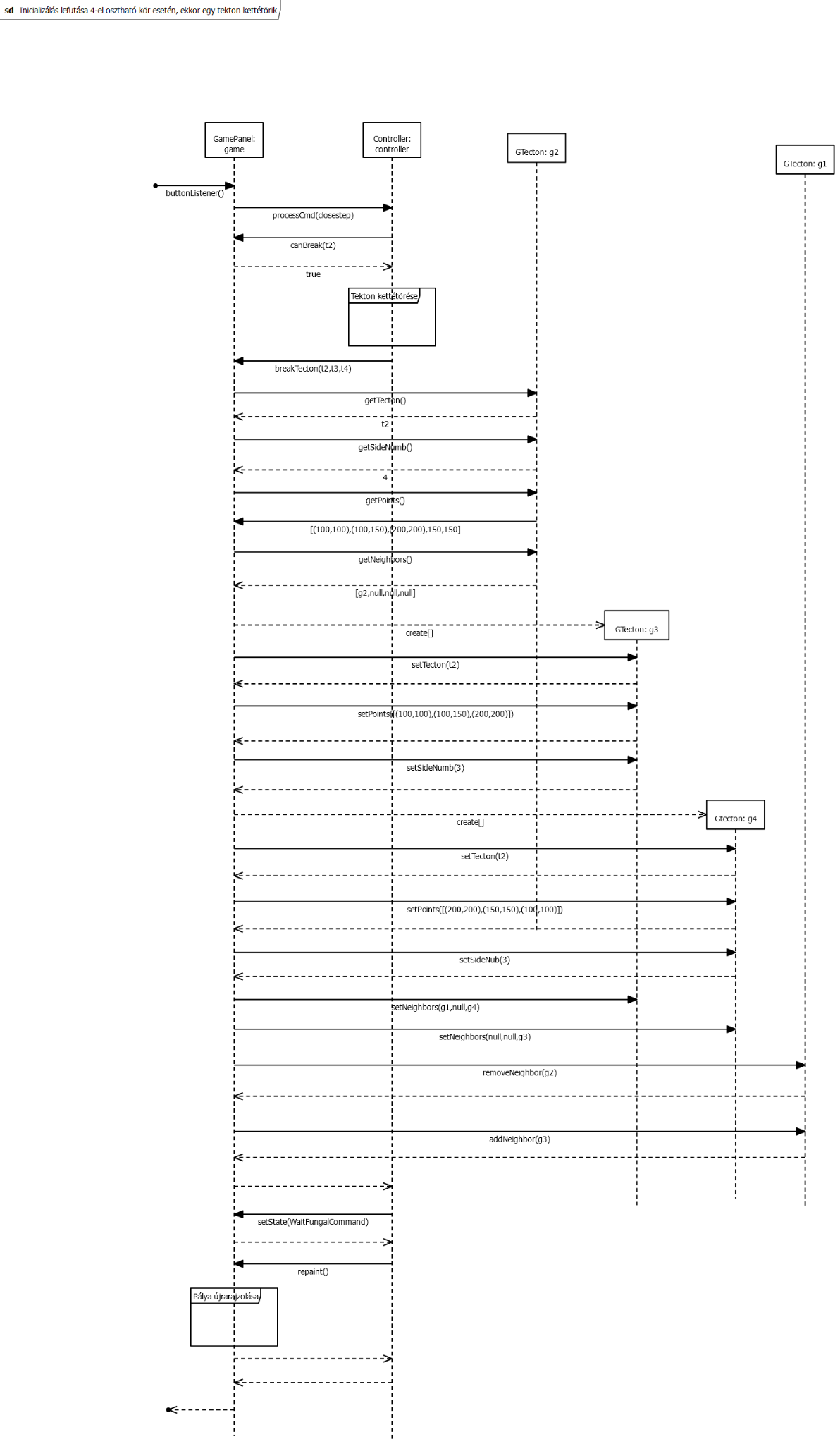
A Gamepanel kezdeti állapota a WaitFungalCommand, azaz vár, hogy a gombász válassza ki milyen lépést szeretne végrehajtani.

A gombász kijelöli a spóra szórása gombot, ennek során a GamePanel a SelectedMushroomForShoot állapotba kerül, azaz várja, hogy jelöljük ki azt a gombatestet, amellyel spórát szeretnénk kilőni.

Ezután kijelölünk egy tektont, amely során a ShootSpore állapotba kerülünk, azaz a program arra vár, hogy kijelöljük, hogy melyik tektonra akarjuk a spórákat kilőni.

Ezután újra kijelölünk egy tektont, létrehozzuk a parancsot, amit átadunk a kontrollernek, megtörténik a spóra kilövése, újrarajzoljuk az ablakot, és újra a WaitFungalCommand állapotba kerülünk.

## **Inicializálás\_lefutása\_4\_el\_osztható\_kör\_esetén\_ekkor\_egy\_tekton\_kettétörik**



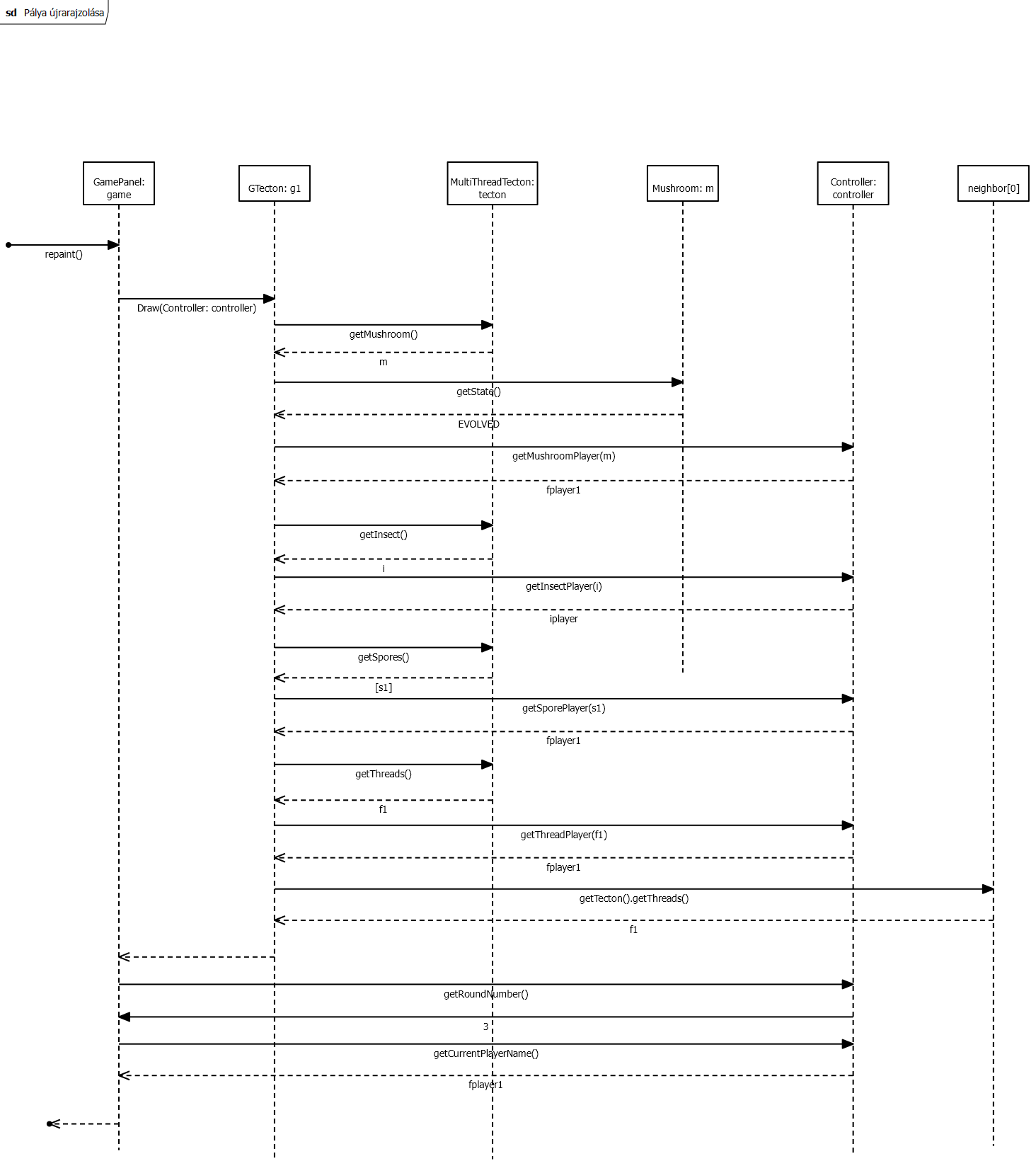
Hívás: A rovarász a lépés lezárása gombra kattint.

A GamePanel a WaitInsectCommand állapotban van, azaz várja, hogy a rovarász kiválassza, hogy milyen lépést szeretne végrehajtani.

A rovarász a lépés lezárása gombra kattint, ezáltal a kontrollernek kiadjuk a closeStep parancsot. Mivel több rovarász játékos nincs, ezáltal egy új kör következik, amely a 4. kör lesz, azaz egy tektonnak ketté kell törnie. Először a kontroller megkérdi, hogy a törésre kijelölt tekton valóban kettétörhető-e, azaz legalább 4 csúcsa van. ezután végrehajtódik a modellben a törés. A GamePanelnek meghívódik a breakTecton metódusa, átadva azt a tektont, amit ketté kell törni, illetve a létrejött két tektont is. A program megkeresi melyik tekton törött ketté végigmenve a tlist listán, amíg meg nem találja. Ezután létrehoz két új tektont, ezeknek a koordinátáját, és oldalszámát az eredeti tekton szerint beállítja, majd beállítja a két új tekton szomszédságait is, valamint azoknak a grafikus tektonoknak a szomszédságait is, amellyel szomszédos volt. Végül a kettétört tektont töröljük a tlist listából.

Miután megtörtént a kör inicializálása, a GamePanel a WaitFungalCommand állapotba kerül, mivel az első gombász jön, és újrarajzolódik az ablak.

## **Pálya\_újrarajzolása**

**

Hívás: A kontroller meghívja a GamePanel repaint metódusát.

Ennek során sorban végigmegyünk a tlist-ben eltárolt összes GTectonon, és meghívjuk a Draw metódusukat, paraméterként átadva a kontrollert. Ennek során a GTecton által eltárolt tektontól lekérdezzük előszőr a gombatestet, ha van gombatest, akkor lekérdezzük a gombatest állapotát. illetve lekérjük a kontrollertől a gombatest játékosát, ez kell, hogy tudjuk, hogy milyen színnel rajzoljuk fel a tektonra. Ezután megnézzük, ha van a tektonon rovar, és ha van akkor ennek is lekérjük a játékosát. Ezután megnézzük ha van spóra, és sorban lekérjük ezek játékosát. Végül megnézzük, ha van gombafonál rajta, és ezeknek is lekérjük a játékosát. Ezután kirajzoljuk a tektont, majd feltüntetjük rajta a gombateste, rovart, spórákat, ha vannak. Megnézzük, hogy az adott tektonnal szomszédos tektonokon van-e ugyanolyan gombafajból származó fonal, és ha igen a játékos színével húzunk egy vonalat a két tekton közé, jelezve, hogy elágazik rajtuk a gombafonál.

Miután a tektonokat kirajzoltuk lekérjük a kontrollertől az adott kör számát, és a játékos nevét, aki most van soron, ezeket is megjelenítjük a képernyő felső részén.

## Játék\_befejezése

**

Hívás: A rovarász megnyomja a lépés lezárása gombot.

A GamePanel kezdeti állapota WaitInsectCommand, azaz a program várja, hogy a rovarász kiválassza milyen lépést fog végrehajtani.

A rovarász a lépés lezárása gombra kattint, ennek során a kontrollernek átadjuk a closeStep parancsot. Mivel nincs több rovarász, ezért a következő kör jön azonban a következő kör megegyezik a max körök számával, így vége lesz a játéknak, meghívódik a GamePanel endGame() függvénye, amely meghívja a MainWindow showWin függvényét, amely egy WinPanel segjtségével megjeleníti, hogy ki nyert és a játékosok pontszámait.

# 

## Napló

| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| --- | --- | --- | --- |
| 2025.05.03 17:00 | 3 óra | Fejes  Szolár  Lekli  Toronyi  Zelch | Grafikus felület tervezése |
| 2025.05.04 10:00 | 9 óra | Zelch | Grafikus felület tervezése, osztálydiagram elkészítése, szekvenciadiagramok elkészítése |
| 2025.05.04 10:00 | 1 óra | Fejes | Osztályok leírásának pontosítása, befejezése. |
| 2025.05.04 13:00 | 4 óra | Szolár | GTecton leírása, formázás. |
| 2025.05.04 15:00 | 3 óra | Lekli | Osztályok leírása |
| 2025.05.04 17:00 | 4 óra | Fejes | Kiosztott View-beli osztályok leírása. |
| 2025.05.05 19:00 | 3 óra | Toronyi | Grafikus felület megjelenítésének finomítása, vázlatrajzok elkészítése és részletes dokumentálása |