Snake Game

dokumentáció

Készítette: Biczó Eszter, Buti Tibor, Szalai Ákos

Tartalom

[Jegyzőkönyv: 3](#_Toc122273046)

[Specifikáció 5](#_Toc122273047)

[State diagram 6](#_Toc122273048)

[Use case diagram 7](#_Toc122273049)

[Osztály diagramok 8](#_Toc122273050)

[Tesztelési terv 12](#_Toc122273051)

[Felhasználói kézikönyv 13](#_Toc122273052)

# Jegyzőkönyv:

* Jó napot kívánok! Kérem foglaljon helyett!
* Jó napot kívánok! Köszönöm!
* Buti Tibor vagyok kollégáim Biczó Eszter és Szalai Ákos.
* Én pedig Gipsz Jakab Pista vagyok.
* Milyen programot szeretne velünk fejleszteni?
* Egy régi típusú játékot, amivel még a nagyapám is játszott.
* A Tetris-re gondol?
* Nem. Esetleg hallott a snake nevű játékról?
* Igen emlékszem rá. Én is sokat játszottam vele gyerekkoromban. De pontosan melyik verzióra gondol?
* Egy egyszerű verzióra gondoltam mivel magam is szeretném megtanulni ennek a játéknak a működését programozói szemmel. Szeretném, hogy ha a program kódot is megkaphatnám, hogy tanulás céljából felhasználjam.
* Természetesen. De ez további költségeket jelenthet.
* Rendben, nem gond.
* Milyen programozási nyelven szeretné, hogy elkészítsük a programot?
* A JAVA-ra gondoltam mert arról nagyon sok jót hallottam.
* Milyen fejlesztő környezetben szeretné elemezni a szoftvert?
* NetBeans-ben mivel úgy értesültem, hogy grafikusan is lehet programozni benne.
* Milyen operációs rendszerben szeretné futtatni az alkalmazást?
* Windows 10.
* Mik az elképzelései a játékkal kapcsolatban?
* Arra gondoltam olyan legyen, mint az eredeti, hogy van egy kígyó és megjelenik egy alma, amit megkell enni, hogy pontot kapjál és emiatt nő a kígyó mérete. Viszont jó lenne, ha inkább kettő kígyó lenne egy helyet, hogy érdekesebb legyen a játék.
* Hogyan legyen a pontozás?
* Legyen kétfajta alma a sima piros, ami 100 pontot ér és lenne egy aranyszínű alma, ami a három szorosát éri. Illetve legyen benne egy rohadt alma, ami levon 500 pontot.
* Negatívba lemehet az eredmény?
* Igen, hogy számon lehessen tartani az igazi összeget.
* Szeretné, hogy a játék közben a pontok növekedése látszódjon?
* Nem, nem lenne jó, ha közben számolná a pontokat mert zavarná a koncentrációmat a játék közben. Ha a játék befejeződött írja ki a nyertes pontszámát.
* A kígyó gyorsuljon miután megette az almát?
* Szerintem ne mert két kígyóval már bőven nehéz a játék.
* Két felhasználós legyen vagy a két kígyó ellentétesen mozogjon egy felhasználó alatt?
* A testvéremmel szeretnék majd játszani tehát legyen inkább két felhasználós, ahol külön lehet irányítani a kígyókat. De legyen lehetőség egyedül is játszani.
* A játék mikor érjen véget?
* Amikor a kígyó saját magába ütközik vagy esetleg falnak ütközik, illetve, ha a másik kígyónak megy neki.
* Milyen platformon gondolkodik használni a szoftvert?
* Leginkább a laptopomat használom így azon szeretném használni.
* Milyen funkciókat szeretne még a programban?
* Szeretném, ha a lenne olyan menü, ahol ki lehetne választani a kígyók színét külön-külön a két felhasználósnál. A színek választásást magukra bízom, hogy ne legyenek ebből a játékban problémák. Mire tudnák elkészíteni a programot és mennyiért?
* Sajnos most nagyon elfoglaltak vagyunk, de január elejére készek leszünk, ha minden jól megy. Mi egy ilyen 100 $/órában gondolkodtunk.
* Rendben. Nekem az megfelel.
* Van még valami, amiről nem beszéltünk?
* Nem szerintem mindent megbeszéltünk. Köszönöm a szakszerű profi hozzá állást. A jövőben is önöket választom majd.
* Köszönjük a bizalmat. További szép napot!

# Specifikáció

Az ügyfél igényei: Mivel a felhasználót érdekli a java nyelv ezért megkért minket, hogy egy egyszerű játék programot(Snake game) készítsünk neki, amiből megtanulhatja a JAVA nyelv és a játék működését. Így együttesen tanul és élvezettel próbálhatja ki gyerekkorát felidéző játékot.

Ügyfél külön kérése: NetBeans-ben szeretné futtatni majd a programot

Játék lényege: Van egy karakterünk, ami jelen esetben egy kígyó, amit irányíthatunk. Célja, hogy a véletlenszerűen megjelenő almákkal tovább növelhetjük a kígyónk méretét, amivel pontot szerzünk. Ahogy nő az elfogyasztott almák száma nő kígyónk mérete, amit egyre nehezebb emiatt irányítani. A játéknak akkor van vége, ha neki meggyünk a saját magunknak vagy a falnak (+két játékosnál ha neki megyünk a másik játékosnak). Ilyenkor kiírja az elért pontszámunkat.

Humán erőforrások: három tervező/fejlesztő

Hardver erőforrások: egy fejlesztői, egy tesztelői számítógép (közepes hardverigény)

Szoftver erőforrások: fejlesztőkörnyezet (NetBeans), verziókövető (Git), projektvezető alkalmazás (Projekt 2013)

Program nyelv: Java

Üzemeltetés: üzemeltetést nem kell biztosítani

Karbantartás: az esetleges hibajavításon felül nem kell biztosítani

Hatékonyság: a program válaszideje gyors, erőforrásokra való támaszkodás mértéke alacsony

Megbízhatóság: megfelelően működő program

Hordozhatóság: a program csak Windows 7, vagy újabb operációs rendszeren fut, személyi számítógépen

Felhasználás paraméterei: két felhasználós, informatikai szaktudás nem szükség a program intuitív módon használható nem kell hozzá kézikönyv

Biztonság: két ugyanolyan szint ne tudjon kiválasztani

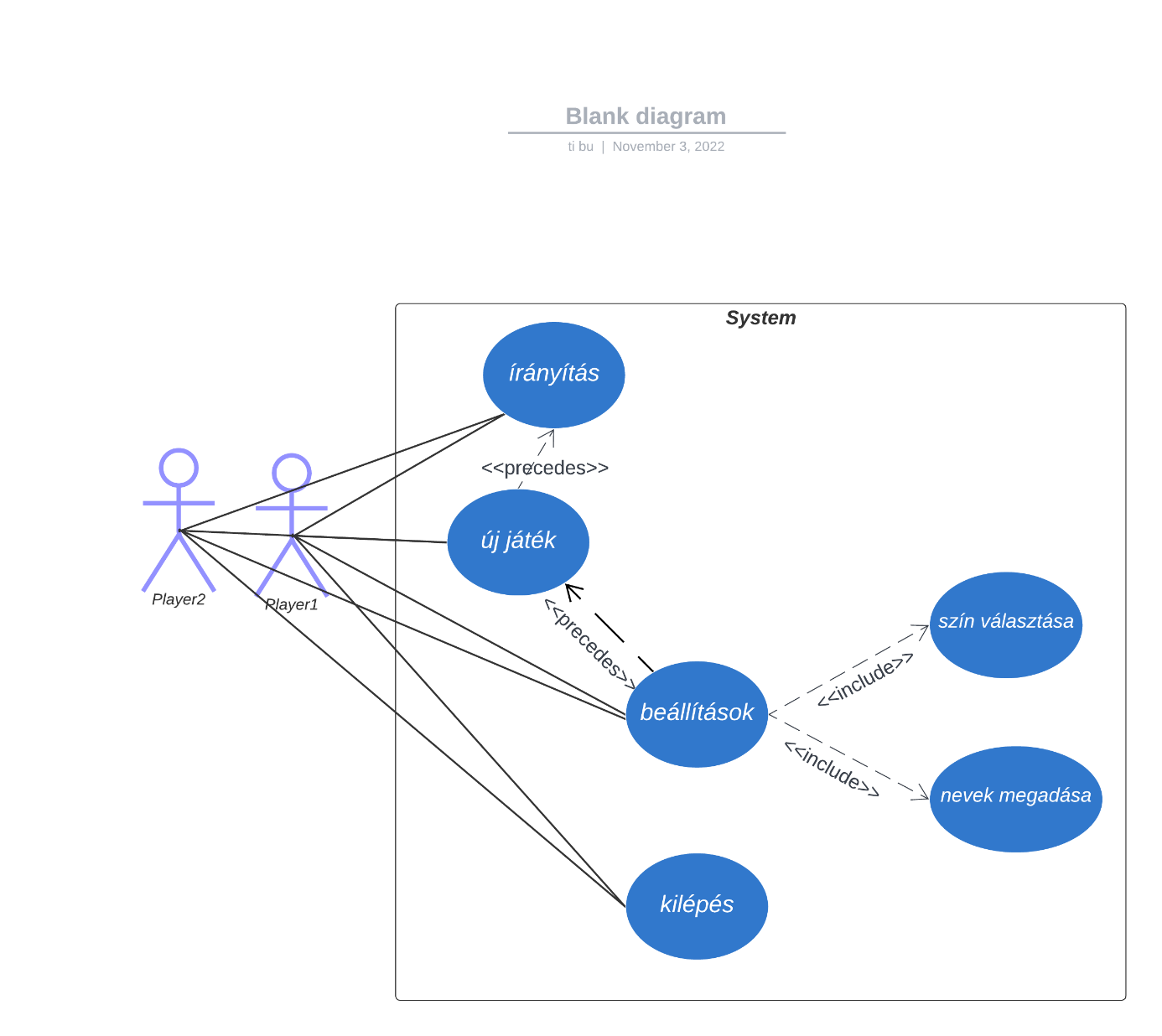
Költség: 100 $/óra

Időigény: január 3

Dátum: 2022.09.26

# State diagram

# Use case diagram



# Osztály diagramok

A képen asztal látható

Automatikusan generált leírás

**Apple**: TimerTask osztályból származik, hogy megfelelő idő eltelte után jelenjen meg a panelen az alma. Az almának van x, y koordinátája és két konstruktora. Az elsőt (Apple (snake: Snakes)) a kezdésnél használjuk. A másodikat (Apple ()) akkor használjuk, amikor a kígyó megeszi az almát. Ilyenkor beállított idő után random generálódik egy új alma a run metódusban.

A képen asztal látható

Automatikusan generált leírás

**RottenApple**: TimerTask osztályból származik, hogy megfelelő idő eltelte után jelenjen meg a panelen az alma. Az almának van x, y koordinátája és két konstruktora. Az elsőt (RottenApple (snake: Snakes)) a kezdésnél használjuk. A másodikat (RottenApple ()) akkor használjuk, amikor a kígyó megeszi az almát. Ilyenkor beállított idő után random generálódik egy új alma a run metódusban.

A képen asztal látható

Automatikusan generált leírás

**GoldApple**: TimerTask osztályból származik, hogy megfelelő idő eltelte után jelenjen meg a panelen az alma. Az almának van x, y koordinátája és két konstruktora. Az elsőt (GoldApple (snake: Snakes)) a kezdésnél használjuk. A másodikat (GoldApple ()) akkor használjuk, amikor a kígyó megeszi az almát. Ilyenkor beállított idő után random generálódik egy új alma a run metódusban.

A képen asztal látható

Automatikusan generált leírás

**Main**: A Main (a: string) meghívja a Settings Panel-t, ami a kezdő oldala a programnak. A Main (a: int) meghívja a Color Chooser-t és miután kiválasztottuk a két színt meghívódik a Main (first:Color, second: Color). Itt a két játékos színei beállítódnak és elindul a két játékos mód. A Main ()-ben az egy játékos mód indul el.

A képen asztal látható

Automatikusan generált leírás

**SettingsPanel**: A kezdő oldalt jeleníti meg, ahol ki lehet választani, hogy egy vagy két játékos játszon.

A képen asztal látható

Automatikusan generált leírás

**ColorChooser**: Ez a két játékos esetén jelenik meg, ahol kiválaszthatják milyen színű legyen a kígyójuk, illetve vissza lehet lépni a kezdő oldalra, ha meggondolnák magunkat.

A képen asztal látható

Automatikusan generált leírás

**Rectangle**: Kígyó test részeinek a pozícióját lehet lekérni, illetve az ütközést figyeli két testrészre. (önmaga vagy egy másik test)

A képen szöveg látható

Automatikusan generált leírás

**Snakes**:

Snakes(window: snakegame.Main, winwidth: int, winheight: int): Egy játékos

Snakes(window: snakegame.Main,first: Color, second: Color, winwidth: int, winheight: int): Két játékos

getBody1(): ArrayList<Rectangle>: Tesztelésnél használjuk az első kígyó teste

getBody2(): ArrayList<Rectangle>: Tesztelésnél használjuk az második kígyó teste

setDirectiona(direction: String): void: Beállítja az irány értékét

getDirectiona(): String: : Lekéri az irány értékét

setDirectionb(direction: String): void: Beállítja az irány értékét

getDirectionb(): String: : Lekéri az irány értékét

addParta(): void: Ha az első kígyó ütközik egy almával, akkor egy kockával hosszabb lett

addPartb(): void: Ha az második kígyó ütközik egy almával, akkor egy kockával hosszabb lett

checkColissiona(): void: Ha a kígyó belemegy a saját farkába vagy ütközik a másik kígyóval a játék véget ér illetve ha a kígyó almát eszik hosszabb lesz

checkColission(): void: Ha a kígyó belemegy a saját farkába vagy ütközik a másik kígyóval a játék véget ér illetve ha a kígyó almát eszik hosszabb lesz

moveSnakea(): void: Ha a kígyó eléri a pálya szélét a játék véget ér, és kiírja a pontszámot viszont ha nem akkor ellenőrzi, hogy történt-e ütközés

moveSnakeb(): void: Ha a kígyó eléri a pálya szélét a játék véget ér, és kiírja a pontszámot viszont ha nem akkor ellenőrzi, hogy történt-e ütközés

drawSnakea(g Graphics): void: Megrajzolja az első kígyót az almákkal amennyiben nincsenek

drawSnakeb(g Graphics): void: Megrajzolja a második kígyót az almákkal amennyiben nincsenek

setApple (apple: Apple): void: Panelre almát állít

getApple (): Apple: Panelen lévő almát lekéri

setGoldApple (goldapple: GoldApple): void: Panelre arany almát állít

getGoldApple (): GoldApple: Panelen lévő arany almát lekéri

setRottenApple (rottenapple: RottenApple): void: Panelre rohadt almát állít

getRottenApple (): RottenApple: Panelen lévő rohadt almát lekéri

paintComponent(g Graphics): void: Háttért fest almákkal illetve egy vagy két kígyót rajzol

# Tesztelési terv

A tesztkörnyezet a SnakesTest.java állományban található

Megvalósított műveletek tesztelése:

1. Első kígyó létrejön
2. A két kígyó nem egyenlő
3. Az irány beállítása megfelelő
4. A kígyó megeszi az almát és emiatt növekszik
5. A kígyó nem ütközik össze az almával
6. A kígyó eléri a pálya jobb szélét
7. A kígyó elmozdul jobbra és le
8. Az újonnan létrehozott alma megegyezik a panelen lévővel

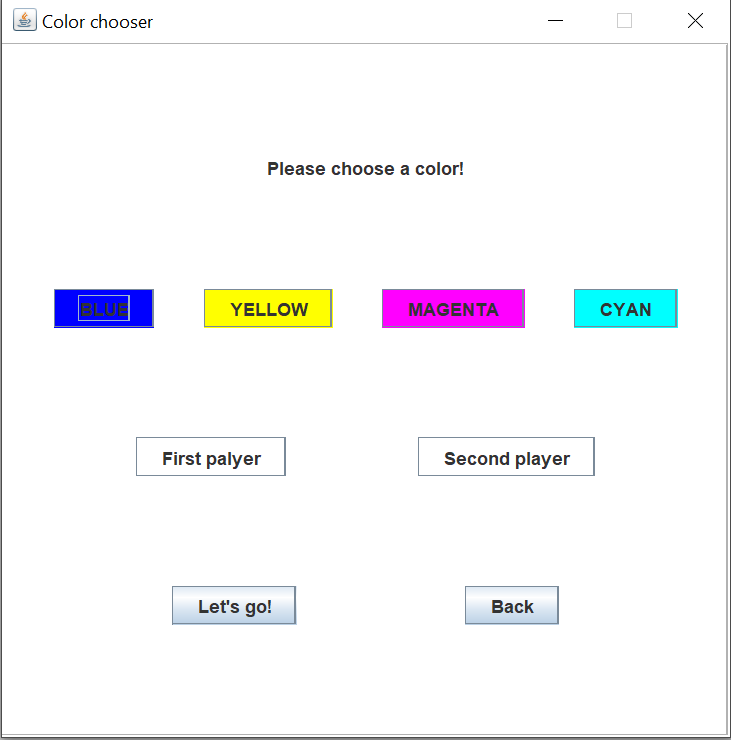
# Felhasználói kézikönyv

# A program elindítása után a fő menü jön elő, ahol ki lehet választani hány játékos akar játszani.

A képen szöveg, hüllő, képernyőkép látható

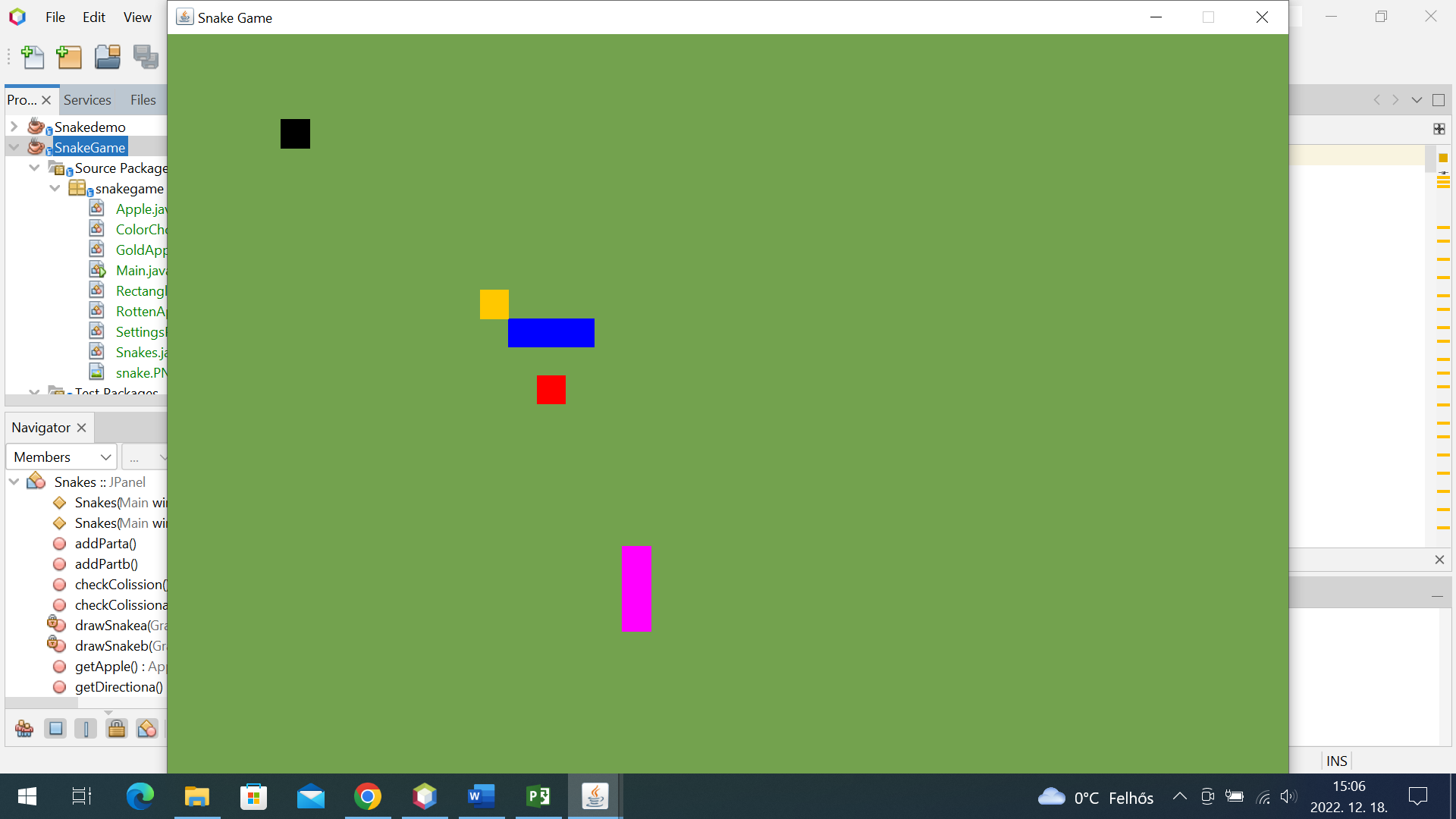
Automatikusan generált leírás

A játék fő menüjében választható ki az egy, illetve két játékos mód. Ezt követően a két játékos módban a szín választó ablak segítségével tudják 4 szín közül kiválasztani a játékosok a számukra megfelelőt (kék, sárga, magenta, cián) ha ez megtörtént akkor a 'let' s go' megnyitásával már indulhat is a játék. Ha meggondolnánk magunkat a 'back' segítségével vissza juthatunk a főmenübe.



A játék indításakor a pályán van:

* kígyó vagy kígyók, amit a játékos irányít (az ábrán a kék-first player(nyilak), illetve magenta színű-second player(WASD))
* alma, plusz 100 pont (piros)
* aranyalma, plusz 300 pont (sárga)
* romlott alma, mínusz 500 pont (fekete)



A kígyó irányításához az egy játékos módban a nyilakkal történik.

Két játékos módban az első játékos a nyilakkal, a második WASD-vel lehet irányitani a kígyókat.

A játék célja minél több pont összegyűjtése falnak ütközés, vagy a kígyó saját magába való ütközése nélkül.

A játéknak vége van, ha a kígyó neki ütközik a falnak vagy saját magának illetve két játékos esetén a másik kígyóba ütközik.

