

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 Alkatrész

- **Felelősség**

A mezőn található alkatrészek felvételében és használatában játszik szerepet.

- **Ősosztályok**

- **Interfészek**

Targy->Alkatresz

- **Attribútumok**

- **Metódusok**

- **bool accept(TargyVisitor v):**
- **void felvesz(Jatekos j):** A paraméterül kapott játékosnak meghívja a void alkatreszFelvesz(Alkatresz a) függvényét, paraméterül saját magát adja.
- **void hasznal(Jatekos j):**

8.1.2 Buvarruha

- **Felelősség**

A mezőn lévő buvarruha felvételéért és használatáért felel.

- **Ősosztályok**

Targy → Buvarruha

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void felvesz(Jatekos j):** A paraméterül kapott játékosnak meghívja a void BuvarruhaFelvesz(Buvarruha b) függvényét, paraméterül saját magát adja.
- **void vedelem(Jatekos j):** A paraméterül kapott játékos vedelem tagváltozóját igazra állítja.
- **void hasznal(Jatekos j):** Védelmet beállítja.
- **void accept(TargyVisitor v):**

8.1.3 Elelem

- **Felelősség**

A mezőn lévő étel felvételéért és használatáért felel.

- **Ősosztályok**

Targy → Elelem

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void felvesz(Jatekos j):** A paraméterül kapott játékosnak meghívja a void elelemFelvesz(Elelem e) függvényét, paraméterül saját magát adja.
- **void accept(TargyVisitor v):**
- **void hasznal(Jatekos j):** Megnöveli a testhőt.

8.1.4 Eszkimo

- **Felelősség**

Leszármazott osztály. Iglut tud építeni.

- **Ősosztályok**

Mozgathato-> Jatekos → Eszkimo

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void Iglutepit():** A mezőre iglut épít.

8.1.5 FulladasiAllapot

- **Felelősség**

Enumeration osztály, a játékos állapotait jeleníti meg a fuldoklás szemszögéből.

- **Ősosztályok**

- **Interfészek**

- **Literálok**

- **aktiv:** A játékos aktív, ha a jégtáblán áll

- **fuldoklik:** A játékos fuldoklik abban a körben, amikor beesett a vízbe és nincs rajta bűvárruha.
- **kimenthető:** A játékos kimenthető a vízbe esést követő körben.
- **halott:** A játékos halott, ha a vízbe esést követő körben sem mentették ki.
- **Metódusok**

8.1.6 Irany

- **Felelősség**

Enumeration osztály, amely lehetséges értékei a négy irány.

- **Ősosztályok**
- **Interfészek**
- **Literálok**
 - **Fel**
 - **Le**
 - **Jobbra**
 - **Balra**
- **Metódusok**

8.1.7 Jatekos

- **Felelősség**

Absztrakt alaposztály. Példányai kutatók, vagy eszközök lehetnek. A játékos a jégmezőn lépkedhet, illetve más tevékenységeket végezhet. Kezeli az életbenmaradásához szükséges attribútumok értékének változását (testhő, fuldoklik).

- **Ősosztályok**

Mozgathato → Jatekos

- **Interfészek**
- **Attribútumok**
 - **kontroller:** Referencia a singleton Kontroller osztályra. Láthatóság: -, típus: Kontroller
 - **tartozkodasiMezo:** A mező, amin a játékos áll. Láthatóság: -, típus: Mezo
 - **munkakSzama:** Egy körben végezhető munka aktuális értéke. Értéke minden kör kezdetén 4, de minden cselekvéssel eggyel csökken az értéke. Minimális értéke 0. Láthatóság: -, típus: int
 - **testho:** A játékos testhője. Kezdetben kutatók esetén 4, eszközök esetén 5 az értéke. Minimális értéke 0. Láthatóság: -, típus: int
 - **vedett:** Értéke igaz, ha a játékosnak van bűvárruhája, hamis, ha nincs. Láthatóság: -, típus: bool

- **allapot:** Értéke a Fulladási állapot négy értékét veheti fel a fent leírtak szerint. Láthatóság: -, Típus: FulladasiAllapot.
- **targyak:** A játékosnál lévő tárgyakat tárolja. Láthatóság: -, típus: ArrayList<Targy>
- **alkatreszek:** A játékosnál lévő alkatrészeket tárolja. Láthatóság: -, típus: ArrayList<Alkatresz>
- **Metódusok**
 - **Jatekos (Kontroller k, int testho):** konstruktor. Láthatóság: public
 - **void játszik():** 4 tevékenység végzését teszi lehetővé a játékosnak. A 4 munka elfogyhat 4 tevékenység elvégzésével, illetve, ha a játékos vízbe esik és nincs rajta bűvaruha. Láthatóság: +
 - **void lep(Irany i):** A játékos az adott irányba lép (A mező elfogad(Jatekos j) és eltávolít(Jatekos j) függvénye segítségével) és csökkenti a munkakSzama attribútum értékét 1-gyel. Láthatóság: +
 - **void meghal():** hívja a kontroller attributumon keresztül a játékVege(bool nyer) függvényt false paraméterrel. Láthatóság: +
 - **void kapar():** A befagyott tárgy felvételéért felel. A mező getTargy() függvényét meghívja, ami visszatér a tárgyat és ezt a játékos felveszi. Csökkenti a munkakSzama attribútum értékét 1-gyel. Láthatóság: +
 - **void lapatFelvesz(Lapat l):** A befagyott lapátot kapja paraméterül és felveszi a tárgyak heterogén kollekciójába a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír a Lapat felvesz(Lapat l) függvénye és az hívódik meg. Láthatóság: +
 - **void kotelFelvesz(Kotel k):** A befagyott kötelet kapja paraméterül és felveszi a tárgyak heterogén kollekciójába a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír a Kotel felvesz(Kotel k) függvénye és az hívódik meg. Láthatóság: +
 - **void elelemFelvesz(Elelem e):** A befagyott ételt kapja paraméterül és növeli a játékos testhőjét egygel a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír a Elelem felvesz(Elelem e) függvénye és az hívódik meg. Ez utóbbi egygel növeli a játékos testhőjét. Láthatóság: +
 - **void buvarruhaFelvesz(Buvarruha b):** A befagyott tárgyat kapja paraméterül és igazra állítja a védett tagváltozó értékét a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír a Buvarruha felvesz(buvarruha b) függvénye és az hívódik meg. Ez utóbbi függvény állítja true-ra a védett attribútum értékét. Láthatóság: +
 - **void alkatreszFelvesz(Alkatresz a):** A befagyott alkatrészt kapja paraméterül és felveszi az alkatrészek tárolójába a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír az Alkatresz felvesz(Alkatresz a) függvénye és az hívódik meg. Láthatóság: +
 - **void satorFelvesz(Sator s):** A paraméterül kapott sátrat beteszi a játékos tárgyak tömbjébe a Targy felvesz(Targy t) függvényén keresztül, amelyet felülír a Sator felvesz(Sator s) függvénye és az hívódik meg. Láthatóság: +
 - **satorEpit(m: Mezo):** Ellenőrzi, hogy van-e sátor a játékos tárgy tömbjében és ha van, meghívja a Sator hasznal(Jatekos: j) metódusát. Láthatóság: +
 - **void lapatol():** Végigmegy a játékos tárgyain és amennyiben van nála lapát vagy törekeny lapát (ez a lapatVisitor segítségével dől el), úgy hívja az adott tárgy hasznal(Jatekos j) függvényét. Ezáltal két egység hó kerül eltakarításra, ha pedig nincs a játékosnak lapátja, úgy egy egységgel csökken a hóréteg. Láthatóság: +

- **void kihuz(Irany i):** Végigmegy a játékos tárgyain és amennyiben van nála kötél (ez a kotelVisitor segítségével dől el), úgy hívja a Kotel osztály hasznal(Jatekos j) függvényét. Láthatóság: +
- **void lerak():** Az adott mezőre rakja a játékosnál levő alkatrészt. Láthatóság: +
- **void vizbeEsik():** Amennyiben a játékos vedett attribútuma false (tehát nincs búvárruhája) úgy beállítja a játékos állapot tagváltozójának értékét fuldoklikra, valamint lecsökkenti a elvégezhető munkák számát (munkakSzama tagváltozó) nullára, hogy a következő játékos jöjjön. Láthatóság: +
- **void osszeszerel():** Ellenőrzi, hogy az adott mezőn van-e mindhárom alkatrész, és ha igen a játékos összeszereli és elsüti a jelzőrakétát. Láthatóság: +
- **void munkaLevon(int i):** A játékos munkáját csökkenti a paraméterként kapott mennyiséggel. Láthatóság: +
- **void elsut():** Ez a függvény az összeszerelés után automatikusan hívódik, hívja a kontroller attribútumon keresztül a jatekVege(bool) függvényt true paraméterrel. Láthatóság: +
- **public ArrayList<Alkatresz> getAlkatreszek():** visszaadja a játékosnál levő alkatrészeket. Láthatóság: +
- **int getTestho():** testho attribútum gettere. Láthatóság: +
- **void setTestho(int t):** testho attribútum settere. Láthatóság: +
- **void setVedett(boolean b):** a vedett attribútum settere. Láthatóság: +
- **void setAllapot(FulladasiAllapot all):** a fulladasiAllapot settere. Láthatóság: +
- **public void epit():** üres függvény, az Eszkimo leszármazott felülírja.
- **public void kutat(Irany i):** üres függvény, a Kutato leszármazott osztály felülírja.

8.1.8 Jegesmedve

- **Felelősség**

Random irányba lép minden körben, és ha egy mezőre kerül egy játékosal megöli a játékost, azaz game over.

- **Ősosztályok**

Mozgathato → Jegesmedve

- **Interfészek**
- **Attribútumok**
- **Metódusok**

void játszik(): Random számot generál 0 és 3 között és abba az irányba meghívja a jegesmedve lép(Irany i) metódusát.

void lép(Irany i): A paraméterül kapott irányba lépteti a jegesmedvét, majd megnézi, hogy az adott jégtablán

8.1.9 Jegtábla

- **Felelősség**

Leszármazott osztály. Számon tartja a rajta lévő tárgyakat. A játékosok tevékenységének eredményeként törli a rajta található tárgyat, illetve alkatrészek esetén fogadja a ráhelyezett alkatrészt vagy a ráépített iglut.

- **Ősosztályok**

Mezo → Jegtábla

- **Interfészek**

- **Attribútumok**

- **fagyottTargy:** A mezőbe befagyott tárgy. Láthatóság: -, típus: Targy
- **fagyottAlkatresz:** A mezőbe befagyott alkatrész. Láthatóság: -, típus: Alkatresz
- **alkatreszek:** A mezőre letett alkatrészek. Láthatóság: -, típus: ArrayList<Alkatresz>
- **iglu:** Értéke defaultból hamis, igaz lesz ha építenek rá iglut. Láthatóság: -, típus: bool
- **SatorMiotaVan:** Értéke alapesetben 0, de ha a jégtáblára valaki sátrat épít értéke 1 lesz, és minden körben nő. Maximális értéke a játékosok száma, ha ezt eléri, újra 0 lesz. Láthatóság: -, típus: int.

- **Metódusok**

- **Jegtábla(int teherbiras, int hotakaro, Targy targy):** Jégtábla konstruktora. Láthatóság: +
- **void elfogad(Jatekos j):** A paraméterül kapott játékost a mező játékosai közé teszi, megnézi, hogy a mező elbírja-e a rajta álló játékosokat. Ha nem, meghívja a játékosok vizbeEsik() függvényét és törli a mezőbe befagyott tárgy értékét, illetve a mezőre lerakott alkatrészek értékét.
- **void horetegCsokkent():** 1-gyel csökkenti a hórétég vastagságát.
- **void getTargy():** Visszaadja a tárgyat ami a jégtáblán van.
- **public void setFagyottTargy(Targy t):** tesztekhez kell, beállítja a tárgyat.
- **public void setFagyottAlk(Alkatresz t):** tesztekhez kell, beállítja az alkatrészt.
- **void alkatreszNovel(int i):** Megnöveli a mezőn lévő alkatrészek számát.
- **void setIglu(bool:b):** Beállítja, hogy van-e a jégtáblán iglu.
- **bool isiglu():** Visszaadja, hogy van-e a jégtáblán iglut.
- **void satorIdoNovel():** Növeli a SatorMiotaVan attribútum értékét 1-gyel.
- **int getSatorMiotaVan():** Visszaadja, hogy mióta áll az adott jégtáblán a sátor.
- **void satratNullaz():** Nullára állítja a SatorMiotaVan attribútum értékét.
- **ArrayList<Alkatresz> getAlkatreszek():** Visszaadja a jégtáblára letett alkatrészeket.
- **Alkatresz getFagyottAlkatresz():** Visszaadja a jégtáblába fagyott alkatrészt.

8.1.10 Kontroller

- **Felelősség**

Ez egy singleton osztály. Irányítja, hogy melyik játékos tevékenykedhet a jégmezőn. Lépteti a jegesmedvét, vihart generál. Ezen felül ellenőrzi, hogy adott körben meghalt-e (kihült vagy megfulladt) valaki, hogy egy adott körben elsüllyedt-e alkatrész, illetve ellenőrzi a jégtáblákon épített sátrak állapotát (kör: adott játékos max 4 tevékenységet elvégez). Amennyiben egy körben kihült vagy megfulladt valaki, illetve ha elsüllyedt egy alkatrész véget vet a játéknak. Ha a jégtáblán a sátor már annyi ideje áll, mint amennyi játékos játszik, leveszi a sátrat a jégtábláról.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **palya:** Tárolja a jégmező összes mezőjét. Láthatóság: -, típusa: ArrayList<Mezo>
- **jatekosok:** Tárolja a játékban résztvevő játékosokat. Minimális értéke 2, maximális értéke 6. Láthatóság: -, típus: ArrayList<Jatekos>
- **jegesmedve:** Ha találkozik egy játékosal megöli. Láthatóság: -, típus: Jegesmedve

- **Metódusok**

- **void jatek():** A soron következő játékos játszik() függvényét meghívja, ami által a játékos max. 4 tevékenységet végezhet. Bizonyos esetekben vihart generál. Ellenőrzéseket végez a detektál függvény segítségével. Lépteti a jegesmedvét. Láthatósága: +
- **void detektal():** Minden körben ellenőrzi, hogy vannak-e fuldoklók, és a fuldoklásuk fázisának megfelelően beállítja a fulladási állapotuk értékét. Amennyiben a játékos állapota halott, véget vet a játéknak. Minden körben ellenőrzi, hogy semelyik alkatrész sem zuhant-e a vízbe. Ha valamelyik bezuhant, véget ér a játék. Ellenőrzi, hogy van-e kihült játékos. Ha igen, véget ér a játék. Megszünteti a sátrakat a mezőkön, ha lejárt az élettartamuk. (Kör: adott játékos maximum 4 tevékenysége) Láthatósága: +
- **void vihar():** Meghívja bizonyos mezőknek a horetegNovel() függvényét, ami által a hótakaró egy véletlen számmal megnő az érintett mezőkön. Ezen kívül azon játékosok testhőjét csökkenti, akik olyan mezőn állnak, ahol nincs iglu vagy sátor. Láthatósága: +
- **void jatekVege(bool):** Ha paramétere false game over, ha true, nyert a csapat. Láthatósága: +

8.1.11 Kotel

- **Felelősség**

A mezőn lévő kötél felvételéért és használatáért felel.

- **Ősosztályok**

Targy → Kotel

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void felvesz(Jatekos j):** A paraméterül kapott játékosnak meghívja a void KotelFelvesz(Kotel k) függvényét, paraméterül saját magát adja.
- **void hasznal(Jatekos j):** Meghívja a kihuz() függvényt.
- **bool accept(Targy Visitor v):**

8.1.12 Kutato

- **Felelősség**

Leszármazott osztály. Meg tudja vizsgálni, hogy egy adott mező hány játékost bír el.

- **Ősosztályok**

Mozgathato-> Jatekos → Kutato

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **int vizsgal(Irany i):** Visszaadja, hogy egy adott irányban található szomszédos mezőnek mekkora a teherbírása.

8.1.13 Lapat

- **Felelősség**

A mezőn lévő lapát felvételéért és használatáért felel.

- **Ősosztályok**

Targy → Lapat

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void felvesz(Jatekos j):** A paraméterül kapott játékosnak meghívja a void lapatFelvesz(Lapat l) függvényét, paraméterül saját magát adja.
- **void hasznal(Jatekos j):** Hóréteget csökkenti.

- **bool accept(Targy visitor v):**

8.1.15 Lyuk

- **Felelősség**

A nulla teherbírású mezők. Befogadja a vízbe esett játékost.

- **Össztályok**

Mezo → Lyuk

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void elfogad(Jatekos j):** A paraméterül kapott játékost a mező játékosai közé teszi és meghívja a játékos vízbeEsik() függvényét.

8.1.16 Mezo

- **Felelősség**

Absztrakt alaposztály. A mezőre lépő játékost befogadja, a róla ellépő játékost leveszi onnan. Kezeli a vízbe esést. A mező példányai Lyuk és Jegtabla típusúak lehetnek.

- **Össztályok**

- **Interfészek**

- **Attribútumok**

- **szomszedok:** Tárolja a szomszéd mezőket. Láthatóság: -, típus: Mezo
- **alloJatekos:** Azok a játékosok, akik a mezőn állnak. Láthatóság: -, típus: ArrayList<Jatekos>
- **alloJegesmedve:** A mezőn tartózkodó jegesmedve. Láthatóság: -, típus: Jegesmedve
- **teherbiras:** A mező teherbíró képessége. Minimális értéke 0. Láthatóság: -, típusa: int
- **hotakaro:** A mezőn lévő hóréteg vastagságát reprezentálja. Láthatóság: -, típus: int

- **Metódusok**

- **Mezo(int teherbiras, int hotakaro):** Mezo konstruktora. Láthatóság: public
- **void horetegNovel():** A mező hórétegének vastagságát eggyel növeli. Láthatóság: public
- **abstract void elfogad(Jatekos j):** A leszármazott osztályok implementálják. Láthatóság: public
- **void eltavolit(Jatekos j):** A paraméterül kapott játékost kiveszi az alloJatekosok tárolójából. Láthatóság: public
- **void elfogad(Jegesmedve j):** Befogadja a paraméterül kapott jegesmedvét. Láthatóság: public
- **void eltavolit(Jegesmedve j):** Eltávolítja az adott mezőről a paraméterül kapott jegesmedvét. Láthatóság: public

- **void testhotCsokkent():** Ellenőrzi, hogy van-e a mezőn iglu, és ha nincs, a mezőn álló játékosok testhőjét csökkenti 1-gyel. Láthatóság: public
- **Mezo getSzomszed(Irany i):** Visszaadja a mező paraméterül kapott irányban lévő szomszédját. Láthatóság: public
- **int getTeherbiras():** Visszaadja a mező teherbíróképességét. Láthatóság: public
- **void satorIdoNovel():** Üres metódus, a jégtábla felülírja. Láthatóság: public
- **int getSatorMiotaVan():** Üres metódus, a jégtábla felülírja. Láthatóság: public
- **void satratNullaz():** Üres metódus, a jégtábla felülírja. Láthatóság: public
- **void utkozik(Jegesmedve j):** Hívja az adott mezőn álló játékos meghal() függvényét. Láthatóság: public

8.1.17 Mozgatható

Absztrakt osztály. Leszármazottai a Jegesmedve és a Játékos.

- **Attribútumok**

tartozkodasiMezo: A mező amin a Mozgatható példány áll. Láthatósága: -, típus: Mezo

- **Metódusok**

abstract void játszik(): A Mozgatható példányok implementálják. Láthatóság: public

abstract void lep(Irany i): A mozgatható objektum lép. Láthatóság: public

void setMezo(Mezo m): A tartozkodasiMezo settere. Láthatósága: +

public Mezo getTartozkodasiMezo(): A tartozkodasiMezo gettere. Láthatósága: +

8.1.18 Sator

- **Metódusok**

void felvesz(Jatekos j): Meghívja a paraméterül kapott játékos satorFelvesz(Sator s) függvényét this paraméterrel. Láthatóság: public

void hasznal(Jatekos j): Az adott mező SatorMiotaVan értékét 1-re állítja. Láthatóság: public

8.1.19 SatorVisitor

A TárgyVisitor interfacet implementálja.

- **Metódusok**

boolean visit(Sator s): truet ad vissza.

boolean visit(Elelem s): falset ad vissza.

boolean visit(Buvarruha s): falset ad vissza.

boolean visit(Kotel s): falset ad vissza.

boolean visit(Lapat): falset ad vissza.

boolean visit(Alkatresz s): falset ad vissza.

8.1.20 Targy

- **Felelősség**

Absztrakt alaposztály. A mezőn található tárgyak játékosnak való átadásáért, illetve a játékosnál levő tárgyak használatáért felel.

- **Ősosztályok**

- **Interfész**

- **Attribútumok**

- **Metódusok**

- **void felvesz(Jatekos j):** A Targy osztály minden leszármazottja implementálja.
- **void hasznal(Jatekos j):** A Targy osztály leszármazottai implementálják.

8.1.21 TorekenyLapat

- **Attribútumok**

int hasznalatSzama: 0 és 3 közötti értéket vehet fel..

- **Metódusok**

void hasznal(Jatekos j): Ha a hasznalatSzama attribútum értéke kisebb 3-nál az adott mezőnek kétszer meghívja a horetegCsokkent() függvényét, ha a hasznalatSzama attribútum értéke kisebb 3-nál. Majd növeli a hasznalatSzama attribútum értékét 1-gyel.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 lepTest

- **Leírás**

A teszt létrehoz egy játékost és elhelyezi egy mezőn. A mezőnek 4 (6?) irányba vannak szomszédai. A teszt kipróbálja a paraméterként kapott irányba, hogy a játékos eltűnik-e a tartózkodási mezőről és megjelenik-e a szomszédos mezőn.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A lépések megfelelő működését ellenőrzi a teszt. Ha olyan irányba lép a játékos amerre nincs szomszédos mező, a játékos a saját mezőjén kéne maradjon, itt jelentkezhet hiba.

- **Bemenet**

*JatekosTest jt create
JajtekosTest jt lepTest String Fel
JatekosTest jt lepTest String Le
JatekosTest jt lepTest String Jobb
JatekosTest jt lepTest String Bal*

- **Elvárt kimenet**

*(4x)
Siker*

mezo: <A játékos eredeti mezőjének megfelelő irányban lévő szomszédja>
testho: <nagyobb 0>
allapot: aktiv

8.2.2 lyukraLepTest

- **Leírás**

A teszt létrehoz egy játékost és elhelyezi egy jégtáblán. A jégtáblának egy szomszédja van és az egy lyuk. A teszt kipróbálja, hogy ha szomszédos mezőn a játékos beleesik a lyukba először bűvárruhával majd anélkül, akkor a fuldoklási állapota megfelelő lesz -e.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy ha a játékos bűvárruhával esik a vízbe, akkor továbbra sem fuldoklik, viszont ha bűvárruha nélkül akkor kezdjen fuldokolni. Hiba ott lehet amikor a mező elfogadja a játékost. Ilyenkor ha rosszul állítja be az állapot a bűvárruha függvényében akkor nem elvárt eredményt kapunk.

- **Bemenet**

JatekosTest jt create
JajtekosTest jt lyukraLepTest

- **Elvárt kimenet**

Siker
mezo: <mező neve>
testho: <nagyobb 0>
allapot: aktiv

Siker
mezo: <mező neve>
testho: <nagyobb 0>
allapot: fuldoklik

8.2.3 megfulladTest

- **Leírás**

A teszt létrehoz egy játékost és elhelyezi egy lyukon. A játékosnak nincs bűvárruhája. A teszt kipróbálja, hogy ha a játékosnak nincs bűvárruhája és 2x meghívódik a detektál függvény (a játékban ez a saját körének a végére és a következő körre vonatkozik) akkor a játékos tényleg megfullad -e;

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy a játékos fuldoklási állapota a két detektálás után megegyezik e az elvárt állapottal.

Hiba akkor jelentkezik, ha az állapotok nem egyeznek meg, ez vagy a detektál függvény nem megfelelő működésére utal, vagy a játékosnak nem jól lettek beállítva az attribútumai.

- **Bemenet**

JatekosTest jt create
JajtekosTest jt medgulladTest

- **Elvárt kimenet**
Siker
mezo: <mező neve>
testho: <nagyobb 0>
allapot: halott

8.2.4 lapatolTest

- **Leírás**
A teszt létrehoz egy kontrollert(ezt csak azért mert kell a Kutato konstruktorához), egy eszkimót, egy kutatót, két jégtaáblát és egy lapátot. Az egyik tábla az eszkimó, a másik tábla a kutató tartalmazási mezője lesz. A lapátot az eszkimó kapja meg. A tesztet leteszteli, hogy helyes mértékű hóréteg marad-e a táblákon.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A funkcionalitás a lapáttal és anélkül való hóeltakarítás, lapáttal 2-t lehet, anélkül csak egyet. Hibalehetőség: hóréteg 1 és a lapát nem 1 réteget takarít el.
- **Bemenet**
JatekosTest jt create
JajtekosTest jt lapatolTest
- **Elvárt kimenet**
Siker, amikor van lapát
Siker, amikor nincs lapát
Siker, amikor van lapát és 1 a hóréteg

8.2.5 kihuzTest

- **Leírás**
A teszt létrehoz egy jégtaáblát és egy lyukat, a megmentő a jégtaáblán áll, a fuldokló a lyukban szenved. Létrejön egy kötél ami értelemszerűen a megmentő lesz. A szomszédságok beállítása után meghívódik a kihúz függvény,
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A funkcionalitás amit ellenőrizni hivatott a teszt az, hogy ki tudja e helyesen húzni valamelyik játékost. Hibalehetőség: nem működik a kihúz függvény és a teszt végeztével nem ugyan azon a mezőn lesz a két játékos.
- **Bemenet**
JatekosTest jt create
JajtekosTest jt kihuzTest
- **Elvárt kimenet**
Siker, ugyan azon a mezőn vannak

8.2.6 lep (JegesmedveTest)

- **Leírás**
A teszt létrehoz 3 mezőt, az egyikre elhelyez egy jegesmedvét, a középsőre egy eszkimót iglu nélkül akit meg kell ölnie a macinak, a harmadikra pedig egy eszkimót igluval. A jegesmedve 2-t fog lépni és automatikusan öl vagy megvédi a játékost az iglu
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A funkcionalitás amit ellenőriz az a Jegesmedve tud-e ölni ha van iglu vagy nem.

- **Bemenet**

- JegesmedveTest jt create*

- JegesmedveTest jt lep*

- **Elvárt kimenet**

- Siker, a medve megölte az iglutanal mezőn csapongó eszkimót*

- Siker, a medve nem ölte meg az iglus mezőn álló eszkimót*

8.2.7 viharTest

- **Leírás**

- A teszt létrehoz egy kontrollert, 4 eszkimót, 9 mezőt. A szomszédságok helyes beállítása után a játékosokat elhelyezi és hozzáadja ezeket a kontrollerhez, majd az meghívja a vihart.*

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A funkcionalitás a vihar havat hord a mezőkre, a játékosok kihűlnek. Hiba lehetőségek: nem működik, rossz az implementáció*

- **Bemenet**

- KontrollerTest kt create*

- KontrollerTest kt viharTest*

- **Elvárt kimenet**

- Siker, minden mezőn több a hó*

- Siker, minden jatekos meghult*

8.2.8 kikaparTest (JatekosTest)

- **Leírás**
A teszt létrehoz egy mezőt amin van egy kötél, egy eszkimót, az eszkimót ráállítja a mezőre, majd meghívja a kapar függvényét.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Funkcionalitás a tárgyak felvétele. Hibalehetőség: nem tűnik el a tárgy a mezőről.
- **Bemenet**
*JatekosTest jt create
JatekosTest jt kikaparTest*
- **Elvárt kimenet**
Siker, a játékosnak van tárgya, a mezőről pedig eltűnt.

8.2.9 osszeszerelTest (JatekosTest)

- **Leírás**
A teszt létrehoz egy játékost és egy mezőt amin áll. A mezőre tesz három alkatrészt, majd a játékos összeszerel függvényét meghívja.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Funkcionalitás: csak akkor lehet összeszerelni ha megvan a három alkatrész egy mezőn. Hibalehetőség: nem hívódik meg az elsut függvény és/vagy abból a jatekVege fv.
- **Bemenet**
*JatekosTest jt create
JatekosTest jt osszeszerelTest*
- **Elvárt kimenet**
Siker, a pszitolyt elsüti a játékos és vége a játéknak.

8.2.10 alkatreszElsullyedTest

- **Leírás**
A teszt létrehoz egy mezőt aminek 1 a teherbírása, és amin van egy fagyott alkatrész. A mezőre tesz két játékost, ennek hatására a jégtábla beszakad és az alkatrész elsüllyed. A

detektál függvény amikor ellenőrzi az alkatrészek számát nem lesz meg a három, így vége a játéknak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
Az alkatrész elsüllyed ha a jégtábla felborul. Hibalehetőség: nem tűnik el az alkatrész így a kontroller nem talál hibát és nem vet véget a játéknak.
- **Bemenet**
KontrollerTest kt create
KontrollerTest kt alkatreszElsullyedTest
- **Elvárt kimenet**
A játéknak vége.

8.2.11 alkatreszFelvesziTest

- **Leírás**

A teszt létrehoz egy jégtáblát nulla hóréteggel, egy eszkimót, aki a jégtáblán áll és egy kontrollert (mert az kell az eszkimó konstruktorához). A jégtáblán be van fagyva egy alkatrész. A játékosnak meghívjuk a kapar() függvényét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy a kapar függvény meghívása után a jégtáblán levő fagyott tárgy értéke null lesz-e, a játékosnál az alkatrészek lista mérete egy lesz-e. Hibalehetőség: kapar() függvény után nem lesz null a jégtáblán a fagyott tárgy értéke, vagy a játékos alkatrészek lista mérete nulla marad.

- **Bemenet**
JatekosTest jt create
JatekosTest jt alkatreszfelvesz()
- **Elvárt kimenet**
Siker, az alkatrészek list mérete 1.
Siker, a jégtáblán a fagyottTargy attribútum értéke null.

8.2.12 vizsgalTest

- **Leírás**

A teszt létrehoz egy kontrollert(ezt csak azért mert kell a Kutato konstruktorához), egy kutatót, és két jégtáblát egymás mellett. Az egyik táblán a kutató áll, a másik tábla teherbírása 2. Hívjuk a játékos vizsgal(Irany i) függvényét a megfelelő irányba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt ellenőrzi, hogy a megadott irányban levő mező teherbírása valóban kettő e. Hibalehetőség: Nem egyezik az elvárt és a kapott szám.

- **Bemenet**
JatekosTest jt create
JatekosTest jt vizsgal String jobb
- **Elvárt kimenet**
Siker, az elvárt és az aktuális kimenet megegyezik.

8.3 A tesztelést támogató programok tervei

A tesztelést JUnit-tal végezzük, ezért a kiértékelés már a teszt esetekben megtörténik. A rendszer állapotát a tesztek kiírják, de emellett megjelenítenek egy olyan üzenetet is ami jelzi, hogy sikerült e a teszt, és ha nem akkor a hibaüzenetből lehet következtetni arra, hogy mi lehet a gond. Ha nem parancssorból futtatjuk a teszteket, hanem a fejlesztőkörnyezetben egy JUnit konfigurációval, akkor akár egyszerre kipróbálhatjuk az összes tesztet, és azok eredményeit egyszerre is megtekinthetjük.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2020.04.09.	1.5 óra	csapat	
2020.04.10 9:00	1.5 óra	Csényi	8.1 rész
2020.04.10 20:00	2.5 óra	Somogyi	8.2.3 - 8.2.5
2020.04.11 10:30	2.5 óra	Erdei	8.2.6-8.2.8
2020.04.11 11:00	2 óra	Erős	8.2.9-8.2.10 és 8.1 egy része