Pac-Man játék

A programozás alapjai 3 házi feladat

Iváncsics Barnabás Pál

CKQKMC

1 A FELADAT LEÍRÁSA

A Pac-Man játékot választottam az idei nagy házi feladatomnak. A játék szabályai hasonlóak az eredetihez képest, de nem ugyanazok. A játékot egyszerre egy ember tudja játszani. A felhasználó van a pacmannel és ellene van 3 szellem.
A három szellem különböző színű és különböző metódussal mozog. Ezen kívül másban nem különböznek.
A pálya téglalap alapú, aminek a széleit falak veszik körbe, valamint a pálya belső részei is falakkal van felbontva. Ezek a falak egységnyi négyzetekből állnak.
Ha egy fallal találkozik (neki megy) a pac-man, akkor megáll egyhelyben és ott marad amíg új irányt nem ad megn neki a felhasználó. Tehát a fallal való érintkezés során a játékos nem veszít az életéből, illetve a pontjaiból.
A szellemek is csak a falak között tudnak mozogni, falakon nem mehetnek át. Ha egy szellem átmegy egy ponton akkor azt nem veszi fel, otthagyja a helyén. A szellemek egymáson át tudnak menni.
A falak között kis pontok vannak, amiket felszedve (átmegy rajta) tudja növelni a pontszámát a játékos. Egy játék során 3 élete van a pacmannek.Ha a pacman felszedte az összes pontot a pályán, akkor pálya újra feltöltődik pontokkal, de a pacman élete és pontjai nem változnak
Ha egy szellemmel találkozik (nekünk jön) a játékos, akkor egy életét elveszíti. Utána a pacman és a szellemek az eredeti helyükről folytatják a játékot.
Ezen kívül a pályán vannak piros pontok is amiket ha felvesz a játékos, akkor körülbelül 15 másodpercig megehet egy szellemet. Ez úgy lehetséges, hogy átmegy rajta. Ha megesz egy szellemet, akkor plusz 10 pontot kap. Viszont egyszerre csak egy szellemet tud megenni tehát az első szellem után visszaáll az eredetire és újra menekülnie kell a játékosnak.
A megevett szellem körülbelül 15 másodperc múlva újra éled és visszatér a játékba. Tehát a szellemeknek az elfogyasztása csak plusz pontok szerzését teszi lehetővé.
A játék célja, hogy minél több pontot szerezzen a játékos mielőtt az utolsó életét is elveszítené.

2 USER MANUAL

A felhasználó miután elindította a játékot a menübe érkezik. A játékos mind a menüben, mind a játékban a billentyűzet gombok segítségével navigálhat.

2.1 MENÜ HASZNÁLATA

A menü két részre osztódik fel a megjelenítés szempontjából. Az egyik a főmenü, a másik pedig a nehézségi szint beállításához szolgáló almenü.

2.1.1 A főmenüben irányítása:

Az alábbi műveletek hajthatóak létre a főmenüben.

Felhasználói művelet	**Művelet eredménye**
Jobb felül található „Bezárás” gombra kattintás.	Kilépés a játékból. (A játék bezáródik mentés nélkül, a program teljesen lefut.)
„1”-es billentyűzet gomb lenyomása.	Játék nehézségének beállítása. (Belép a szint beállításának almenüjébe.)
„2”-es billentyűzet gomb lenyomása	Elmentett játék betöltése majd indítása.
„3”-as billentyűzet gomb lenyomása	Új játék kezdése. (Betöltés és indítás.)

2.1.2 Nehézségi szint almenü

A főmenüben az 1-es gomb lenyomásával juthatunk el ide. A szint beállításával a pacman és a szellemek sebességét állíthatjuk be 2 szintben. Az alábbi műveletek hajthatóak létre ebben az almenüben.

Felhasználói művelet	**Művelet eredménye**
Jobb felül található „Bezárás” gombra kattintás.	Kilépés a játékból. (A játék bezáródik mentés nélkül, a program teljesen lefut.)
„1”-es billentyűzet gomb lenyomása.	Szint beállítása normálra. Visszalép a főmenübe.
„2”-es billentyűzet gomb lenyomása	Szint beállítása nehézre. Visszalép a főmenübe.

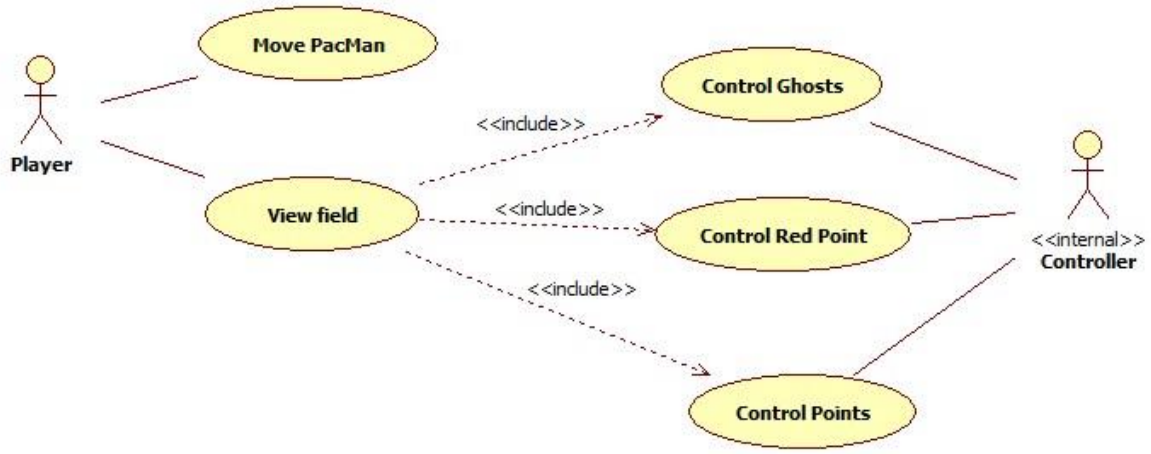
2.2 A JÁTÉK IRÁNYÍTÁSA

A fűmenüben a 2-es vagy a 3-as gomb lenyomása után indul a játék. A játékban az alábbi műveletek hajthatóak létre.

Felhasználói művelet	**Művelet eredménye**
Jobb felül található „Bezárás” gombra kattintás.	Kilépés a játékból. (A játék bezáródik mentés nélkül, a program teljesen lefut.)
„fel” billentyűzet gomb lenyomása.	A pacman mozgási irányának beállítása felfelé.
„le” billentyűzet gomb lenyomása	A pacman mozgási irányának beállítása lefelé.
„jobbra” billentyűzet gomb lenyomása	A pacman mozgási irányának beállítása jobbra.
„balra” billentyűzet gomb lenyomása	A pacman mozgási irányának beállítása balra.
„esc” billentyűzet gomb lenyomása	Kilépés a játékból mentéssel. A játék bezáródik, de a játék állapota elmentődik. Ezt a főmenüben indíthatjuk újra. Az automatikus mentés után a menübe jut vissza a felhasználó.

3 USE-CASE-EK

3.1 USE-CASE DIAGRAM



3.2 USE-CASE LEÍRÁSOK

Cím	**Move PacMan**
Leírás	A játékos a pac-man-t irányítja a pályán
Aktorok	Player
Főforgatókönyv	**1.** A játékos a pacman mozgási irányát átllítja. Lehetséges irányok: fel, le, jobbra, balra.
Alternatív forgatókönyv	**1.A.1.** Ha olyan irányba állítja a járékos amerre fal van, akkor tovább megy a már aktuális irányba.
Alternatív forgatókönyv	**1.A.1.A.1** A játékos új irányt állíthat be.
Alternatív forgatókönyv	**1.B.1.** Ha a játékos nem változtat az irányon, akkor addíg megy abba az irányba a pac-man amíg egy fallal nem találkozik és ott megáll.
Alternatív forgatókönyv	**1.B.1.A.1** A játékos új irányt állíthat be.
Alternatív forgatókönyv	**1.C.1.** Ha a pac-man átmegy egy sárga ponton, akkor azt felveszi és kap egy pontot
Alternatív forgatókönyv	**1.C.1.A.1** Ha az összes pontot felvette, akkor új kör indul. (Az élete és pontszáma megmarad.)
Alternatív forgatókönyv	**1.C.1.A.1.A.1** A pálya újra feltöltődik pontokkal.
Alternatív forgatókönyv	**1.D.1.** Ha a pac-man átmegy egy piros ponton, akkor azt felveszi és utána körülbelül 15 másodpercig támadni képes.
Alternatív forgatókönyv	**1.D.1.A.1.** Ha találkozik egy szellemmel akkor azt megöli és kap érte 10 pontot
Alternatív forgatókönyv	**1D.1.A.1.A.1.** A szellem körülbelül 15 másodpercen belül feltámad.

Cím	**View map**
Leírás	A játékos megnézi a térképet.
Aktorok	Player
Főforgatókönyv	**1.** A program kirajzolja a képernyőre a pálya aktuális állapotát. **2.** A játékos megtekinti a pálya aktuális állapotát.

Cím	**Control Ghosts**
Leírás	A szellemek mozognak a pályán (a falak között) és sebzik a pac-man-t.
Aktorok	Controller
Főforgatókönyv	**1.** A szellemek mozogni tudnak a pályán a saját algoritmusuk alapján a következő irányokba: fel, le, jobbra, balra.
Alternatív forgatókönyv	**1.A.1.** Ha fallal talákozik, akkor nem áll meg, hanem elindul egy új lehetséges irányba.
Alternatív forgatókönyv	**1.B.1.** Ha találkozik a pac-man-nel, akkor annak egy életét leviszi.
Alternatív forgatókönyv	**1.B.1.A.1.** Ha a pac-man-nek nem marad több élete, akkor vége a játéknak.

Cím	**Control Red Point**
Leírás	Piros pontok a helyükön maradnak, amíg fel nem veszik őket.
Aktorok	Controller
Főforgatókönyv	**1.** Egy piros pontot felvett a pacman.
Alternatív forgatókönyv	**1.A.1.** Elindul egy visszaszámláló.
Alternatív forgatókönyv	**1.A.1.A.1.** Ha lejár az idő, akkor random lerak a controller egy új piros pontot a pályára.
Főforgatókönyv	**1.B.1.** A pac-man támadni képe.
Alternatív forgatókönyv	**1.B.1.A.1.** Ha a pac-man találkozik egy szelemmel akkor azt megöli.
Alternatív forgatókönyv	**1.B.1.A.2** Az időzítő leáll és a controller egy új piros pontot lerak a pályára.
Alternatív forgatókönyv	**1.C.1.** A piros pontot törli a controller.

Cím	**Control Points**
Leírás	A pontok a helyükön maradnak, amíg fel nem veszik őket.
Aktorok	Controller
Főforgatókönyv	**1.** Egy pontot felvett a pacman.
Alternatív forgatókönyv	**1.A.1.** A pontot törli a controller.
Alternatív forgatókönyv	**1.B.1.** A pac-man-nek egyel több pontja lesz.

4 STRUKTURÁLIS LEÍRÁS

4.1 AZ OSZTÁLYOK LEÍRÁSA

Az olvashatóság kedvéért az egyes osztályok működéséhez szükséges getter és setter metódusokat nem írtam le az osztályok metódusai közé.

4.1.1 Main

Felelősségek

A program fő osztálya. Itt van a main metódus, aminek a meghívásával indul a program.

Attribútumok

-

Metódusok

+main(args: Sring[]): static void	Létrehoz egy JFrame-et ami a játék megjelenítéséért és a bemenetek figyeléséért felel. Utána egy Timer-ben futtatja a program működéséhez a megfelelő metódusokat, amíg a programot le nem állítják.

4.1.2 Thing

Felelősségek

A játékban lévő dolgok ősosztálya. Ebből származik le minden olyan osztály ami a pálya részét képezi. Implementálja a Serializable interfészt.

Attribútumok

#x: double	A dolog x koordinátája.
#y: double	A dolog y koordinátája.
#unit_s: double	A dolog egységmérete a pályán.

Metódusok

+Thing(x: double, y: double, unit_s: double)	Az osztály konstruktora. Beállítja a fent leírt adattagok értékeit.

4.1.3 PacMan

Felelősségek

A pac-man-t reprezentálja a játékban. Mozgásának, életének és pontjainak állításáért felel. A Thing ősosztályából származik le. További attribútumai és metódusai:

Attribútumok

- original_x: double	A pac-man x koordinátáját tárolja le és őrzi meg. (Mozgatás miatt szükséges.
- original_y: double	A pac-man y koordinátáját tárolja le és őrzi meg. (Mozgatás miatt szükséges.
-points: int	A pac-man pontszáma.
-life: int	A pac-man élete.
direction: Direction	A pac-man aktuális iránya.
next_direction: Direction	A pac-man következő iránya.
-t: int	A pacman támadásának időzítéséhez szükséges változó.
-attack: Boolean	Igaz, ha pac-man támadni képes.

Metódusok

+PacMan(x: double, y: double, unit_s: double)	PacMan konstruktora. A paraméterként kapott dolgokat átadja az ősosztály konstruktorának, majd beállítja az original_x, original_y attribútumait.
+move()	A pac-man-t mozgatja. Meghívja az elején azokat a metódusokat, amik ellenőrzik, hogy pontot szed-e fel, vagy ütközik-e szellemmel. Ezután átállítja az aktuális irányt a következőre, ha az nem a falnak irányítja
+Collision(bricks: Brick[0..*], plus_x: double, plus_y: double)	Ezt a metódust hívja meg a move metódus, ahhoz, hogy a átállítsa a pac-man irányát. Ellenőrzi, hogy falnak menne-e az új irány beállításáva és ha nem akkor beállítja azt.
+eat_points()	Meghívja annak a pontnak a CollisonPacman metódusát amin éppen átment.
+coll_Ghost()	Ha találkozik egy szellemmel és tud támadni akkor azt megöli és kap 10 pontot. Ha nem tud támadni, akkor meghal és átállítja a newgame-t igazra, hogy a game folytatni tudja a játékot a megfeleő paraméterekkel.

4.1.4 Brirck

Felelősségek

A játékban lévő téglákat reprezántalja. A Thing le. Más attribútumokat és függvényeket nem valósít meg. (Az objektum orientáltság miatt van külön osztályban.)

Attribútumok

-

Metódusok

+Brick(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat.

4.1.5 Points

Felelősségek

A játékban lévő pontokat reprezántalja. A Thing ősosztályból származik le. További attribútumai és metódusai:

Attribútumok

-

Metódusok

+Points(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat.
+CollisonPacman(pac: PacMan)	Absztrakt metódus. A leszármazottjai itt valósítják meg azt, hogy mi történjen akkor amikor megeszi őt a pac-man.
+isRed()	Absztrakt metódus. Visszatérési értékét a leszármazottjai állítják be a különböző szempontok alapján.

4.1.6 Point

Felelősségek

A játékban lévő sima pontokat reprezántalja. A Points ősosztályból származik le. Absztrakt metódusok megvalósítása.

Attribútumok

-

Metódusok

+Point(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat.
+CollisonPacman(pac: PacMan)	Növeli a pac-man pontját egyel, majd törli magát a pontok listájából.
+isRed()	Hamissal tér vissza.

4.1.7 RedPoint

Felelősségek

A játékban lévő sima pontokat reprezántalja. A Points ősosztályból származik le. Absztrakt metódusok megvalósítása.

Attribútumok

-

Metódusok

+Point(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat.
+CollisonPacman(pac: PacMan)	Ha pacman nem támad, akkor törli magát a pontok listájából. Majd beállítja a pac-man-t támadásra.

+isRed()	Igazzal tér vissza.

4.1.8 Ghosts

Felelősségek

A játékban lévő szellemeket reprezántalja. A Thing ősosztályból származik le. További attribútumai és metódusai:

Attribútumok

#color: GhostColor	A szellem színét tárolja.
- original_x: double	A szellem x koordinátáját tárolja le és őrzi meg. (Mozgatás miatt szükséges.
- original_y: double	A szellem y koordinátáját tárolja le és őrzi meg. (Mozgatás miatt szükséges.
#direction: Direction	A szellem aktuális iránya.
#next_direction: Direction	A szellem következő iránya.
#t: int	A szellem feltámadásának időzítéséhez szükséges változó.
#time: int	A szellem random mozgásának időzítéséhez szükséges változó.

Metódusok

+Ghost(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat. Ezuttán beállítja az original_x és original_y attribútumokat.
+step_one()	Az iránynak megfelelően mozgatja a szellemet.
+stepWhenCollison(): boolean	Az következő iránynak megfelelően meghívja a Collison metódust a megfelelő paraméterekkel. Boolean- al tér vissza, hogy tudja az a metódus, hogy falnak menne-e amelyik meghívta.
+Collison(): boolean	Ellenőrzi, hogy falnak menne-e az új irány beállításáva és ha nem akkor beállítja az új irányt, majd hamissal tér vissza. Ha ütközik, akkor igazzal tér vissza.
+countdown(): boolean	A szellem feltámadásának a késleltetéséhez szükséges számláló.
+new_direcrion()	A szellem mozgási irányának megváltoztatásához szüksége metódus. Random general egy új következő irányt. Ezt addig csinálja amíg az nem egyezik meg az aktuális iránnyal.

+new_direcrion2()	A szellem mozgási irányának megváltoztatásához szüksége metódus. Lekéri a pac-man koordinátáit és azok alapján változtatja meg a következő irányt. Ha az a falnak vinné, akkor akkor meghívja a new_direction metódust.
+new_direcrionUporDown()	A szellem mozgási irányának megváltoztatásához szüksége metódus. Random beállítja a következő irányt “fent”-re vagy “lent”-re. Az előző metódus egyszerűsítéséhez szükséges.
+random_move()	A szellem mozgási irányának megváltoztatásához szükséges metódus. Random számok segítségével vagy visszatér igazzal vagy beállítja a következő irányt az aktuális irányra. Ez ahhoz szükséges, hogy ne cask a falnak ütközéskor változtasson irányt a szellem.

4.1.9 BlueGhost

Felelősségek

A játékban lévő kék szellemet reprezántalja. A Ghost ősosztályból származik le. Az ősosztály step függvényét írja felül.

Attribútumok

-

Metódusok

+BlueGhost(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat. Utána beállítja a szellem színét.
+step()	A szellemet mozgatja az ősosztály new_direction2 metódusával.

4.1.10 RedGhost

Felelősségek

A játékban lévő piros szellemet reprezántalja. A Ghost ősosztályból származik le. Az ősosztály step függvényét írja felül.

Attribútumok

-

Metódusok

+RedGhost(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat. Utána beállítja a szellem színét.
+step()	A szellemet mozgatja az ősosztály new_direction és random_move() metódusaival.

4.1.11 OrangeGhost

Felelősségek

A játékban lévő kék szellemet reprezántalja. A Ghost ősosztályból származik le. Az ősosztály step függvényét írja felül.

Attribútumok

-

Metódusok

+OrangeGhost(x: double, y: double, unit_s: double)	Az osztály konstruktora. Átadja az ősosztálynak a paraméterként kapott attribútumokat. Utána beállítja a szellem színét.
+step()	A szellemet mozgatja az ősosztály new_direction2 és random_move() metódusaival.

4.1.12 Menu

Felelősségek

A játékban lévő menüt reprezántalja. A menü állapotaihoz megfelelően állítja be a a játék és a menü állapotát. Implementálja a Serializable interfészt.

Attribútumok

-state: MenuState	A menu állapotát tárolja.

Metódusok

+menu_step()	A menü adott állapotához képest állítja be a játék állapotát, tölt be mentett játékot. Illetve utána mindig visszaállítja a menü állapotát MAIN-re

4.1.13 BrickVis

Felelősségek

A játékban lévő téglák kinézetéért felel. Implementálja a Drawable interfészt.

Attribútumok

-brick: Brick	A megjelenítei kivánt tégla.
-size: double	A pálya egy egységének mérete.

Metódusok

+BrickVis()	Beállítja a paraméterként kapott brick-et és a méretet.
+draw()	Kirajzolja a paraméterben kapott Graphics2D-re kasztolt framre a brick-et a beállított paraméterek segítségével.

4.1.14 Game

Felelősségek

Tárolja a játékban lévő pac-man-t, szellemeket, pontokat, piros pontokat és a játék menüjét. A játék logikai irányításáért felel. Implementálja a Stepable és Serializable interfészeket.

Attribútumok

-pacman: PacMan	A játékban lévő pac-man.
-menu: final Menu	A játékban lévő menu.
-bricks_list: Brick[0..*]	A játékban lévő téglák listája.
-points_list: Points[0..*]	A játékban lévő pontok listája.
-ghosts_list: Ghost[0..*]	A játékban lévő élő szellemek listája.
-death_ghosts_list: Ghost[0..*]	A játékban lévő halott szellemek listája.
-state: GameState	A játék állapota.
-wait: int	Az új kör késleltetéséhez szükséges változó.
-line: int	A pálya sorainak száma.
-column: int	A pálya oszlopainak száma.
-speed: int	A játék sebessége.
-unit_size: double	A pálya egy egységének (cellájának) a mérete.
-newgame: boolean	Az új kör beállításhoz szükséges Boolean.
game: static Game	Statikus singleton osztály. (Az aktuális játékhoz.)

Metódusok

+Game(unit_size: double, speed: int)	Beállítja a paraméterként kapott egységet és sebességet, majd meghívja a reading metódust.
+step()	A játékot a játék állapotához képest megfelelően lépteti. Meghívja a többi léptethető osztály lépés függvényét és a pacman mozgásához szükséges függvényt. Ezt a függvényt egy timer hívja meg, a beállított időközönként. (Így frissíti a játék állapotát.)
+reading()	Beolvassa a megadott .txt kiterjesztésű fájlból a pályát és annak megfelelően feltölti a listákat és beállítja a pac-man-t is.
+rebornGhost()	Átteszi a halott szellemek lista első elemét az élő szellemek listájába.

+remove_points(point: Point)	Törli a paraméterként kapott pontot a pontok listájából.
+remove_ghost(index: int)	Átteszi a paraméterként kapott szellemet az élő szellemek listájából a halott szellemek listájába.
+addRandomRedPoint()	Random helyre lerak egy piros pontot a pályán olyan helyre, ahhol nincsen még sárga pont vagy fal.
+delayNewGame(): boolean	Késlelteti az új kör indulását (hogy ne egyből induljon újra az új kör). Hamissal tér vissza ha lejárt a számláló.
+ghostStep()	Meghívja az összes élő szellem step metódusát.
+rebornPacman()	Visszarakja a pac-man-t és a szellemeket az eredeti helyükre. (Ez akkor hívódik meg amikor egy szellem megöli pac-man-t.)
+EndRound()	Ha nincs már több pont a pályán, akkor meghívja a NewRound metódust.
+NewRound()	Feltöltö újra a játékot pontokkal és szelemekkel. A pac-man-t is visszateszi a kiindulási pontra, de pontjait és életét nem változtatja. A reading metódust használja segítségként.
+resetGame()	Reseteli a játékot, mindent visszaállít az eredetire. (Töröl mindent, majd újra beolvassa a pályát a fájlból.)
+loadField()	Betölti a szerializáltan mentett Game osztályt majd átadja azt a singleton game-nek. (Ezzel tud mentett játékot betölteni.)
+saveField()	Szerializáltan menti a game-et. (Így menti a játék állapotát ahhoz hogy később a fenti loadField-el be lehessen olvasni.

4.1.15 PointVis

Felelősségek

A játékban lévő pontok kinézetéért felel. Implementálja a Drawable interfészt.

Attribútumok

-point: Points	A megjelenítei kivánt pont.
-size: double	A pálya egy egységének mérete.

Metódusok

+PointVis()	Beállítja a paraméterként kapott point-ot és a méretet.
+draw()	Kirajzolja a paraméterben kapott Graphics2D-re kasztolt framre a point-ot a beállított paraméterek segítségével.

4.1.16 GhostVis

Felelősségek

A játékban lévő téglák kinézetéért felel. Implementálja a Drawable interfészt.

Attribútumok

-ghost: Ghost	A megjelenítei kivánt szellem.
-filename: Stirng	A szellem ábázolásához szükséges kép elérési útja és neve.

Metódusok

+GhostVis()	Beállítja a paraméterként kapott ghost-ot.
+draw()	Kirajzolja a paraméterben kapott Graphics2D-re kasztolt framre a ghost-ot a filename segítségével.

4.1.17 PacManVis

Felelősségek

A játékban lévő pac-man kinézetéért felel. Implementálja a Drawable interfészt.

Attribútumok

-pacman: PacMan	A megjelenítei kivánt pac-man.
-size: double	A pálya egy egységének mérete.

Metódusok

+PacManVis()	Beállítja a paraméterként kapott brick-et és a méretet.
+draw()	Kirajzolja a paraméterben kapott Graphics2D-re kasztolt framre a brick-et a beállított paraméterek segítségével.

4.1.18 MenuVis

Felelősségek

A játékban lévő meu kinézetéért felel. Implementálja a Drawable interfészt.

Attribútumok

-menu: Menu	A megjelenítei kivánt pac-man.
-unit: double	A pálya egy egységének mérete.
-line: double	A pálya szélessége.

Metódusok

+PacManVis()	Beállítja a paraméterként kapott menu-t.
+draw()	Kirajzolja a paraméterben kapott Graphics2D-re kasztolt framre a brick-et a beállított paraméterek segítségével.
+level_draw()	A szint beállítási almenü kinézetének beállítása.
+maind_draw()	A főmenü kinézetének beállítása.

4.1.19 PacManPanel

Felelősségek

A játék megjelenítéséért felelős panel. Ez hívja meg az összes pályán lévő dolog draw fügvényét és rajzoltatja ki azt. Jpanel-ből származik le.

Attribútumok

-unit: int	A pálya egy egységének mérete.
- line: int	A pálya szélessége.

Metódusok

+paintComponent(g: Graphic)	Felülírja a JComponent metódusát úgy, hogy az összes pályán lővő dolog megjelenjen.
+setTransparety(i: int)	Beállítja az átlátszóságot a paraméterként kapott változónak megfelelően.

4.1.20 PacManFrame

Felelősségek

A játék megjelenítéséért felelős Frame. Megjeleníti a hozzáadott panelen lévő dolgokat. Ezen kívül a felhasználói billentyűzet műveleteire figyel. JFrame osztályból származik le.

Attribútumok

-

Metódusok

+PacManFrame()	Létrehoz egy JFrame-et, majd azt megfelelően beállítja. Aztán hozzáad egy PacManPanel-t. Ezen kívül a játék és a menü állapotainak megfeleőn kezeli le a billentyűzeten lenyomott gombokat és hívja meg a hozzájuk tartozó különböző metódusokat.
+playSound(filename: String)	Lejátsza a paraméterben kapott file névhez tartozó hang file-t.

4.1.21 GameTest

Felelősségek

A Game osztály step, remove_points és addRandomPoint metódusait teszteli le.

Attribútumok

-game: Game	Tesztelni kívánt osztály objektuma.

Metódusok

+step()	Leteszteli azt, hogy ha már nics több élete a pac-man-nek, akkor megváltoztatja-e a a játék állapotát a megfelelő állapotra.
+remove_points()	Leteszteli, hogy törölt-e a megadott pontot a pontok listából.
+addRandomRedPoint()	Leteszteli, hogy hozzáadott-e egy piros pontot a pontok listájához

4.1.22 GhostTest

Felelősségek

A Ghost osztály step_one, stepWhenCollison, collison, countdown és new_direction metódusait teszteli le.

Attribútumok

-ghost: RedGhost	Tesztelni kívánt osztály objektuma.

Metódusok

+step_one ()	Leteszteli azt, hogy elmozdult-e a szellem.
+stepWhenCollison()	Leteszteli azt, hogy érzékeli-e ha fal van arra amerre a következő irány be van állítva.
+collison()	Leteszteli azt, hogy ha falnak megy akkor változtat-e az irányon.
+countdown()	Leteszteli azt, hogy az időzítő a megfelelő érték felett nulázza magát.
+new_direction()	Leteszteli azt, hogy az irány tényleg megváltozott-e.

4.1.23 PacManTest

Felelősségek

A PacMan osztály move és collison metódusait teszteli le.

Attribútumok

-pac: PacMan	Tesztelni kívánt osztály objektuma.

Metódusok

+move ()	Leteszteli azt, hogy elmozdult-e a pac-man a megfelelő irányban.
+collison()	Leteszteli azt, hogy mozog-e a szellem ha nincs előtte fal, illetve azt, hogy ha megváltoztja a játékos a fal felé az irányt, akkor mozog-e tovább és nem megy neki a falnak.

4.1.24 PointTest

Felelősségek

A Point osztály collisionPacman teszteli le.

Attribútumok

-pac: PacMan	Teszteléshez szükséges PacMan objektum.

Metódusok

+collisionPacman()	Leteszteli azt, hogy ha a pac-man átmegy-e pontot, akkor annak növeli-e pontját.

4.1.25 Direction

Felelősségek

Ez az enumeráció a mozgatható dolgok lehetséges irányait reprezentálja: DOWN, LEFT, RIGHT, UP.

4.1.26 GameState

Felelősségek

Ez az enumeráció a játék lehetséges állapotait reprezentálja: MENU, GAME.

4.1.27 GhostColor

Felelősségek

Ez az enumeráció a szellemek lehetséges színeit reprezentálja: RED, BLUE, ORANGE.

4.1.28 MenuState

Felelősségek

Ez az enumeráció a menü lehetséges állapotait reprezentálja: MAIN, LOAD_GAME, SET_LEVEL, START_GAME.

4.1.29 Stepable

Felelősségek

Inerfész, ami az összes olyan dolgot reprezentálja amit körönként léptetni kell.

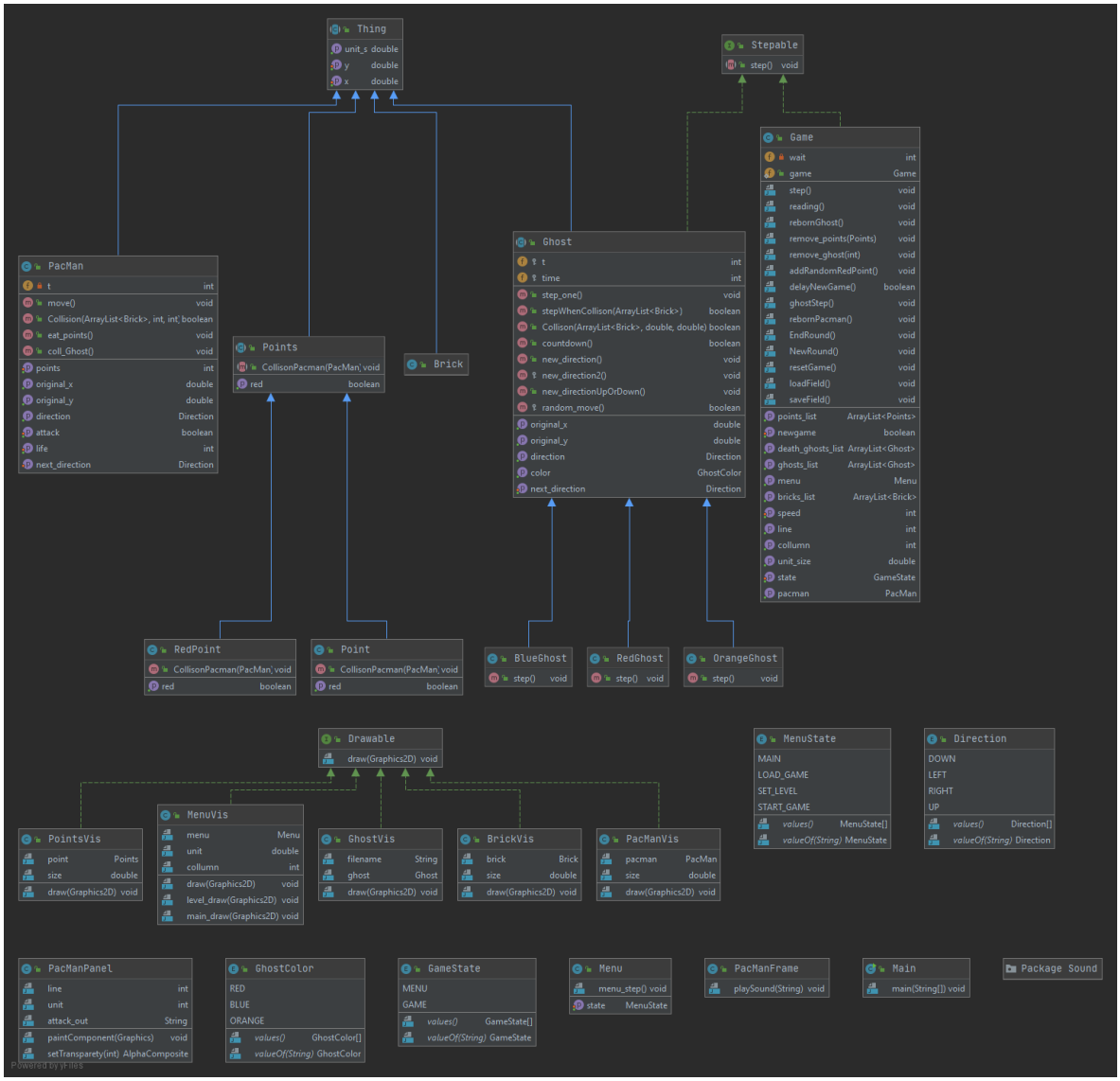
4.1.30 Drawable

Felelősségek

Inerfész, ami az összes olyan dolgot reprezentálja amit ki meg kell jeleníteni a játék folyamán.

4.2 Osztálydiagram

További osztálydiagramok a mappában vannak képként csatolva. A csatolt osztálydiagramon nem jelenik meg az összes kapcsolat.



5 BE ÉS KIMENETEK

A pálya létrehozását egy .txt kiterjesztésű fájl segítségével oldottam meg. A reading függvény a fájl sorait egyesével bolvassa és végigmegy rajta karakterenként. A fájl egy sora a pálya egy sora és egy oszlopa a pálya egy oszlopa. Tehát minden sorba ugyanannyi karakternek kell lennie. Az „F” karakter egy fal, „p” egy pont, „P” a pac-man (ebből értelemszerűen csak egy lehet), „x” egy piros pont, a „R”, „B”, „O” a három szellem. A játék állapotát szerializáltan mentem és olvasom be a loadField- és saveField metódusok segítségével. A game osztályt mentem le és olvasom vissza.
Ezen kívül a programban találhatóak hangeffektek is. Ezt a playSound a megadott nevű .wav kiterjesztésű fájlok lejátszával végzi el.
A szellemek megjelenítéséhez .png kiterjesztésű fájlokat használok amit GhostVos osztály draw metódusa végzi el.

6 TERVEZÉSI MEGFONTOLÁS

A program elemeinek megalkotásánál a cél a könnyű elérhetőség volt, ezért ezeket: a falakat, a pontokat, a szellemeket, és a halott szellemeket mind külön, a saját listájukban tárolódnak. Így könnyen lehet bennük keresni, vagy hozzáadni és kitörölni egy elemet. A szellem halálánál például a megevett szellemet csak átteszi a halott szellemek listájába, majd miután lejárt az időzítő, visszateszi az élők közé.

A szellemek algoritmusa különbözőek. A legbutább szellem csak véletlenszerűen közlekedik a pályán. A két okosabb minden irányváltásnál lekéri a pac-man helyzetét és ahhoz közelít. Az okos szellemeket is két típusra lehet bontani: az egyik akkor változtat irányt amikor falnak ütközik, a másik pedig random időközönként is.

A legtöbb objektumomhoz két osztály tartozik: egyik a működését valósítja meg a másik pedig a megjelenítését. A megjelenítéshez, Swing alapú GUI-t használok. Minden egy framen jelenik meg, amit a repaint metódus segítségével tud frissíteni. Az egyes elemek kirajzolásához a Graphics2D metódusait használom fel.