

A programozás alapjai 3 Házi Feladat

Dokumentáció

Program leírása:

A program egy amőbajátékot valósít meg, ahol a felhasználó amőbát tud játszani a gép ellen. A játék során az egyik játékos a felhasználó, a másik játékos helyében a program játszik. A felhasználó lépése közben eltelt időt a program nyomon követi, és limitálja; amennyiben a rendelkezésére álló idő alatt a játékos nem lép, a program nyer. Emellett a játékos el tudja menteni a játékállást, és később a legutóbbi játékállást be tudja tölteni, így a korábbi játékot tudja folytatni. A játékot a felhasználó egy grafikus felületen keresztül éri el, amivel egérmegnyitásokkal tud kommunikálni.

Tervezési megfontolások, felhasznált adatszerkezetek:

A játéktábla mezőinek jelölésére egy Enum-ot vezettem be Marker néven. Ennek lehetséges értékei az üres mező, a kör, az iksz, és egy úgynevezett határmező (Border jelölő). Határmező az a mező, amellyel van szomszédos, egy játékos által lerakott jelölő. Ez a határmező típus a felhasználó számára nem látható, csupán belső reprezentáció.

A gép a lépése meghatározásra előre gondolkodik egy lépést, és azon mezőkre, ahol jelenleg Border jelölő van, elhelyezi saját jelölőjét. Ezután az előálló játékállásokat lepontozza, és a számára legkedvezőbbet választja ki.

A pontozás azon alapul, hogy az amőbában léteznek kedvező bábuelrendezések. A legmagasabb rangú az öt egymás melletti azonos bábu, ekkor nyert valaki, ez egyféle elrendezést jelent (két jelölővel kettőt). Ennél alacsonyabb rangúból van 5 db olyan elrendezés, amikor már csak egy bábut kell elhelyezni, hogy nyerjünk. Ezután van 10 db elrendezés, ahol két lépés kell a győzelemhez. Valamint van 10 db elrendezés, ahol három lépés kell még a győzelemhez.

A pontozás lényege, hogy egy mezőre megpróbálom ráilleszteni a fenti 25 elrendezést. A ráillesztés négy irányban történhet, vízszintesen, függőlegesen, jobb átlósan és bal átlósan. A pontozás során egy mező ötösben (amely csak egy féle bábut tartalmaz) megszámlolom hány bábu van és az alapján adok neki pontot.

Pl.: vizsgált mező pirossal | . . . x . . x . ; |: a játéktábla széle

vízszintesen öt illesztés lehetséges (illesztések kiemelve):

| . . . x . . x . , ez egy jó minta két bábuval, három lépés és nyerhet az iksz

| . . . x . . x . , ez egy jó minta két bábuval

| . . . x . . x .

| . . . x . . x .

| . . . x . . x . , ez a vizsgált mező ötös nem létezik

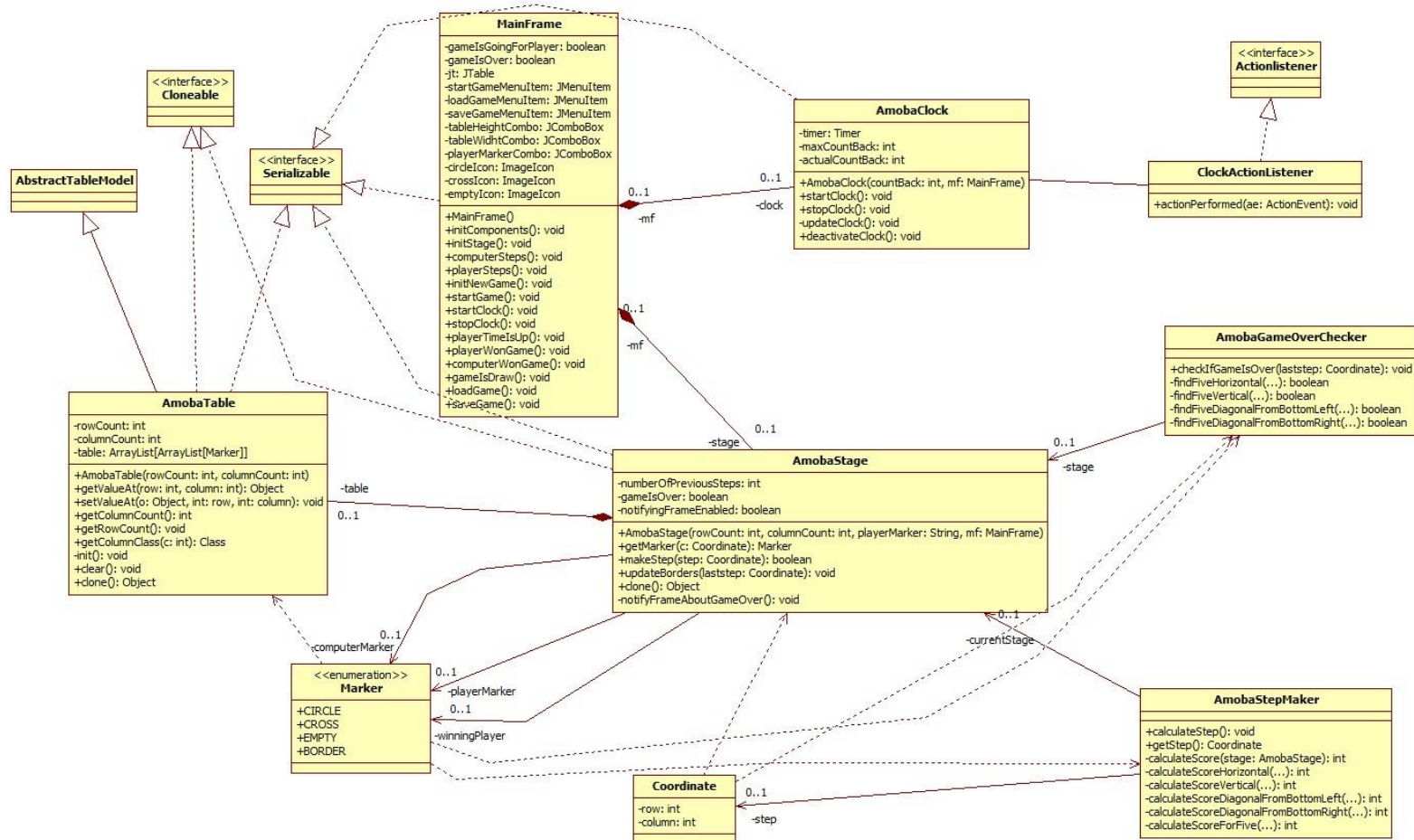
Természetesen így pl. az első jó mintát még egyszer megtalálja a program, amikor a másik ikszet vizsgálja.

Annak felderítése, hogy véget ért-e a játék ugyanilyen elvek mentén zajlik.

A játéktábla eltárolására egy 2D-s tömböt használ a program, amit ArrayList-ekkel valósít meg. Ezáltal a játéktábla egy mezője elérhető a sor, és oszlop értékének megadásával.

Osztályok dokumentációja:

Az osztály diagramon a getter/setter metódusokat nem tüntettem fel, hogy átláthatóbb legyen. Emellett a MainFrame osztály kevésbé fontos inicializációs metódusait sem tüntettem fel, hogy átlátható maradjon a diagram. A használt modellező eszköz(WhiteStarUML) nem támogatja az attribútumok között a < és > jelet, ezért nem teljesen pontosan jelenik meg az AmobaTable osztály table attribútuma, és getColumnClass metódusa. Ezenkívül a tagosztályokat sem tüntettem fel az átláthatóság kedvéért, kivéve az AmobaClock osztályban lévő ClockActionListenert, amelyet egy asszociációval jelöltem, a megfelelő szimbólumok hiányában. Az osztálydiagram külön képként is csatolva van.



A getter/setter metódusok leírását nem adtam meg, mivel azok egyértelműek, és átláthatatlanná tennék a dokumentációt.

Enum Marker

Leírás: Egy mezőjét reprezentálja a játéktáblának.

Elemek:

EMPTY	Üres mező.
CROSS	Iksz-et reprezentál.
CIRCLE	Kör-t reprezentál.
BORDER	Egy határmezőt reprezentál. Azaz olyan mezőket, amelyek szomszédosok valamelyik játékos jelölőjével. A felhasználó számára nem jelenik meg, csupán belső reprezentáció.

Coordinate

Leírás: Egy lépést reprezentál, azaz tulajdonképpen a játéktábla egy mezőjét.

Attribútumok:

-row: int	A játéktáblán a sor száma
-column: int	A játéktáblán az oszlop száma

Metódusok: (getter/setter-ek vannak csak még)

Coordinate(row: int, column: int)	Létrehozza az objektumot.

MainFrame

Leírás: A felhasználó számára látható ablak. Feladata a felhasználó inputjának fogadása, a bemenetek függvényében történő akciók elindítása, a játéktábla megjelenítése.

Attribútumok:

gameIsGoingForPlayer: boolean	A játékos számára aktív-e a játék(ő van-e soron)
gameIsOver: boolean	Vége van-e a játéknak.
stage: AmobaStage	A játékállás
clock: AmobaClock	Az idő megjelenítéséért felel.
jt: JTable	A játéktábla megj.-éséhez kell.

startGameMenuItem: JMenuItem	A játék elindításáért felelős menüItem.
loadGameMenuItem: JMenuItem	A játék betöltéséért felelős menüItem.
saveGameMenuItem: JMenuItem	A játék elmentéséért felelős menüItem.
tableHeightCombo: JComboBox	A játéktábla magasságát lehet rajta keresztül beállítani.
tableWidthCombo: JComboBox	A játéktábla szélességét lehet rajta keresztül beállítani.
playerMarkerCombo: JComboBox	A felhasználó jelölőjét lehet rajta keresztül beállítani.
circleIcon: ImageIcon	A kör jelölő megjelenítéséhez kell.
crossIcon: ImageIcon	Az iksz jelölő megjelenítéséhez kell.
emptyIcon: ImageIcon	Az üres mező jelölő megjelenítéséhez kell.

Metódusok:

MainFrame()	Létrehozza az objektumot.
initComponents(): void	Inicializálja a megjelenítésért felelős komponenseket.
initStage(): void	Beállítja a jt-nek a stage table-ját Model-nek.
computerSteps(): void	A számítógép lép.
playerSteps(): void	A játékos lép.
initNewGame(): void	Egy új játék indítása előtt inicializálja az attribútumokat.
startGame(): void	Új játékot indít.
startClock(): void	Elindítja az órát.
stopClock(): void	Leállítja az órát.
playerTimeIsUp(): void	A játékos ideje lejárt.
playerWonGame(): void	A játékos nyert.
computerWonGame(): void	A számítógép nyert.
gameIsDraw(): void	A játék döntetlen lett.
loadGame(): void	Betölti az utoljára elmentett játékot.
saveGame(): void	Elmenti az aktuális játékot.
initTableHeightComboBox(): void	Inicializálja a TableHeightComboBox-t.
initTableWidthComboBox(): void	Inicializálja a TableWidthComboBox-t.
initPlayerMarkerComboBox(): void	Inicializálja a PlayerMarkerComboBox-t.
setRowHeight(): void	Inicializálja a játéktábla magasságának megjelenését.
setRowHeight(): void	Inicializálja a játéktábla szélességének megjelenését.
initMarkerPictures(): void	Inicializálja a megjelenítéshez szükséges ImageIconokat.

AmobaStage

Leírás: Egy játékállást tárol. Számon tartja a játéktáblát, hány lépés volt eddig, ki melyik jelölővel van, vége van-e a játéknak, illetve, hogy ki nyert. Amennyiben engedélyezve van neki, értesíti a MainFrame-et ha nyert valaki.

Attribútumok:

numberOfPreviousSteps: int	Eddigi lépések száma.
gameIsOver: boolean	Véget ért-e ez a játékállás.
notifyingFrameEnabled: boolean	Értesítheti-e mf-et ha véget ért a játékállás. Erre azért van szükség, hogy szabályozni lehessen, mert az „előre gondolkozás” során a gép jelölőket helyez el a játéktáblán, és így is kialakulhat egy nyerő ötös, ekkor pedig semmilyen MainFrame-nek sem szabad szólni, hiszen ezek a lépések még nem történtek meg valójában.
table: AmobaTable	A játéktábla.
computerMarker: Marker	A gép jelölője.
playerMarker: Marker	A játékos jelölője.
winningPlayer: Marker	A nyertes jelölője. EMPTY ha nincs még nyertes, és BORDER ha döntetlen az állás.
mf :MainFrame	Az adott játékállás MainFrame-je.

Metódusok:

AmobaStage(rowCount: row,columnCount: int, playerMarker: String, mf: MainFrame)	Létrehozza az objektumot. Eközben létrehozza az adott attribútumú AmobaTable-t. A playerMarker-nek „Circle”-nek vagy „Cross”-nak kell lennie.
getMarker(c: Coordinate): Marker	Visszaad egy adott mezőn álló jelölőt.
makeStep(step: Coordinate): boolean	Végrehajt egy adott mezőre egy lépést. Visszaadja, hogy sikeres volt-e a lépés. Ha sikeres volt frissíti a határmezőket és ellenőrzi, hogy véget ért-e a játék, adott esetben szól mf-nek.
updateBorders(laststep: Coordinate): void	Frissíti a játéktáblán a border mezőket, meg kell neki adni az utoljára megtett lépést.
clone(): Object	Mély másoláshoz, viszont a mf-et sekélyen másolja.
-notifyFrameAboutGameOver(): void	Ha meghívjuk, akkor meghívja a mf megfelelő függvényeit, ha vége a játéknak.

AmobaTable

Leírás: A játéktáblát reprezentálja. Egy 2D-s tömbben tárolja a játéktáblát. Az AbstractTableModel-ből származik, ezáltal egy JTable-nek a „Model”-je lehet.

Attribútumok:

rowCount: int	Összes sor száma.
---------------	-------------------

columnCount: int	Összes oszlop száma.
table: ArrayList<ArrayList<Marker>>	A játéktábla.

Metódusok:

AmobaTable(int : rowCount, int columnCount)	Létrehozza az objektumot, az adott magasságú és szélességű játéktáblát.
-init(): void	Inicializálja a játéktáblát a sorok és oszlopok száma alapján.
clear(): void	Az egész játéktáblát EMPTY-vel tölti fel.
clone(): Object	Mély másoláshoz.

(+ AbstractAmobaTable felüldefiniált metódusai, amik a felüldefiniált működést valósítják meg)

AmobaStepMaker

Leírás: Egy kapott játékalállásból kiszámítja a legkedvezőbb következő lépést.

Attribútumok:

currentStage: AmobaStage	Az eredeti játékalállás.
step: Coordinate	A kiszámolt legjobb lépés. Kezdetben (-1,-1). (nem érvényes indexelésű).

Metódusok:

AmobaStepMaker(stage : AmobaStage)	Létrehozza az objektumot. step-et (-1,-1)-nek állítja.
calculateStep(): void	Kiszámolja a legjobb lépést, és beállítja erre step-et.
-calculateScore(stage: AmobaStage): int	Kiszámolja egy adott játékalállás pontszámát, és azt visszaadja.
-calculateScoreForFive(int, int, Marker, Marker)	Egy adott mezőből számolt ötösnek kiszámolja a pontszámát.
-calculateScoreHorizontal(AmobaTable, Marker, Marker, int, int, int, int)	A játéktáblán egy adott mezőhöz kiszámolja a pontszámot vízszintesen
-calculateScoreVertical(AmobaTable, Marker, Marker, int, int, int, int)	A játéktáblán egy adott mezőhöz kiszámolja a pontszámot függőlegesen.
-calculateScoreDiagonalFromBottomRight(AmobaTable, Marker, Marker, int, int, int, int)	A játéktáblán egy adott mezőhöz kiszámolja a pontszámot a jobb alsó sarokból jobbfelfele átlósan(dok. elején kifejtve)

-calculateScoreDiagonalFromBottomLeft(AmobaTable, Marker, Marker, int, int, int, int)	A játéktáblán egy adott mezőhöz kiszámolja a pontszámot a bal alsó sarokból balfelfele átlósan.
---	---

AmobaGameOverChecker

Leírás: Egy kapott játékállás esetén eldönti, hogy véget ért-e már a játék. Ha véget ért, a megfelelő információkat beállítja a játékállás attribútumain.

Attribútumok:

stage: AmobaStage	A vizsgált játékállás.

Metódusok:

AmobaStepMaker(stage : AmobaStage)	Létrehozza az objektumot.
checkIfGameIsOver(laststep: Coordinate): void	Megnézi, hogy véget ért-e játék. Ha véget ért, a megfelelő információkat beállítja a játékállás attribútumain.
-findFiveHorizontal(AmobaTable, int, int, int, int)	Megpróbál a játéktáblán egy adott mezőnél öt egyező jelölőt vízszintesen találni.
- findFiveVertical(AmobaTable, int, int, int, int)	Megpróbál a játéktáblán egy adott mezőnél öt egyező jelölőt függőlegesen találni.
- findFiveDiagonalFromBottomRight(AmobaTable, int, int, int, int)	Megpróbál a játéktáblán egy adott mezőnél öt egyező jelölőt a jobb alsó sarokból jobbfelfele átlósan indulva találni. (dok. elején kifejtve)
- findFiveDiagonalFromBottomLeft(AmobaTable, int, int, int, int)	Megpróbál a játéktáblán egy adott mezőnél öt egyező jelölőt a bal alsó sarokból balfelfele átlósan indulva találni.

AmobaClock

Leírás: Az idő mérését és megjelenítését végzi. Képes szólni a MainFrame-nek ha lejárt a felhasználó ideje. Az időmérés megvalósításához a Swing keretrendszer Timer osztályát használja.

Attribútumok:

timer: Timer	Swing keretrendszerben lévő Timer osztály.
maxCountBack: int	A játékos összes ideje egy lépés során sec-ban.
actualCountBack: int	A játékos aktuális ideje egy lépés során sec-ban.
mf : MainFrame	Az objektumot tartalmazó MainFrame.

Metódusok:

startClock(): void	Elindítja az órát.
stopClock(): void	Leállítja az órát.
-updateClock(): void	Frissíti az óra képét.
deactivateClock(): void	Deaktiválja az óra képét.

Ki-és Bemenetek, Fájlok szerkezete:

A program öt fájlt használ a működése során, amelyek a program projektmapájában vannak.

Ebből 4 db képfájl, amely a különböző jelölők, illetve játéktábla mezők megjelenítésére szolgál. Ezek a fájlok rendre a hatar.png, iksz.png, kor.png, ures.png . Mindegyik képfájl 59x57 pixeles. Ezen fájlok módosítása esetén a program a módosított képet jeleníti meg az eredeti helyett.

A program által használt ötödik fájl egy mentésfájl, aminek a neve amobasavefile.dat . Ez a fájl tartalmazza a szükséges adatokat egy korábbi játék visszatöltéséhez. A fájl a Java által felkínált sorosítás segítségével készül, és egy AmobaStage osztálypéldányt tartalmaz. Azaz ha az AmobaStage, vagy bármely általa tárolt osztály szerkezete megváltozik, akkor a változtatások előtti mentésfájl már nem lesz kompatibilis a programmal.

Felhasználói kézikönyv:

A programmal egérkattintásokkal lehet kommunikálni. A bal felső sarokban található egy Actions menü, aminek a menüpontjaira való kattintásokkal különböző akciókat lehet végrehajtani.

Ezek az alábbiak:

- **Start new game!**
A megadott beállításokkal egy új játékot kezd.
- **Load previous game!**
Ha előzőleg elmentettünk egy játékállást, akkor erre kattintva azt betölthetjük.
- **Save current game!**
A jelenlegi játékállásunkat tudjuk elmenteni a gombra kattintva. Mindig csak egy játékállásunk lehet elmentve egyszerre. Ezenkívül véget ért játékot már nem lehet elmenteni.

Középen felül különböző beállítási lehetőségek vannak, valamint egy óra. Az óra mutatja a hátralévő időt. Ha az idő lejár, akkor a program nyer. Ha már csak öt másodpercünk van, az óra számjegyei piros színűek lesznek.

A különböző beállítási lehetőségek:

- Table height:

- A játéktábla magasságát lehet beállítani vele.
- Table width:
A játéktábla szélességét lehet beállítani vele.
- Your marker:
A felhasználó által használt jelölőt lehet beállítani vele.

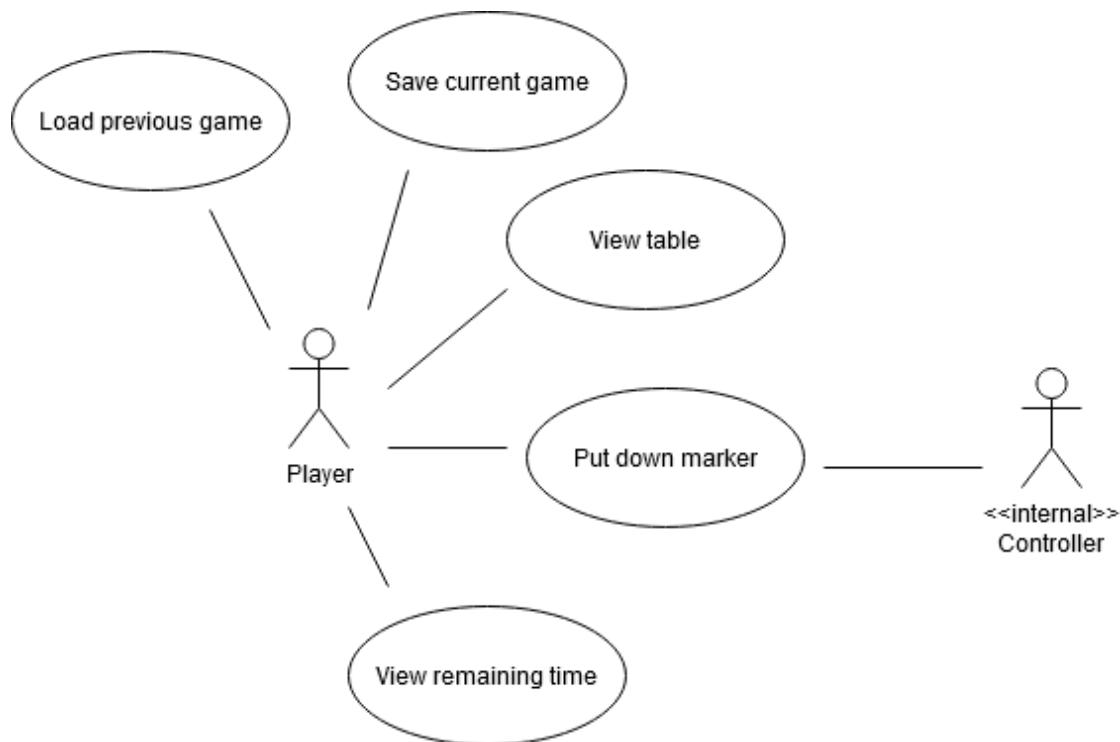
Ha be van töltve egy játék, akkor a játéktábla a beállítási lehetőségek alatt található.

Egy játék elindulása esetén mindig a felhasználó kezd. Ahhoz, hogy egy jelölőt lerakjunk a táblára, az adott mezőbe kell egyszer kattintani. Egy érvényes lépés után a program is lép.

Ha a játéknak vége, akkor az eredményről egy felugró ablak tájékoztatja a felhasználót. Ezen az ablakon az Ok-ra kattintva ismét a programba jutunk, ahol új játékot kezdhethetünk, vagy éppenséggel betölthetjük a legutoljára mentettet.

A programból kilépni az ablak sarkában található X-re kattintva lehet.

Use Case-ek:



Use-case diagram

Use case leírások

Cím	View table
-----	------------

Leírás	A játékos megtekinti a játéktáblát.
Aktorok	Player
Főforgatókönyv	1. A rendszer kirajzolja a játéktáblát. 2. A játékos megtekinti a játéktáblát.

Cím	Put down marker
Leírás	A játékos köre során lép.
Aktorok	Player
Főforgatókönyv	1. A játékos egérgattintással kijelöli, hova szeretné a szimbólumát letenni. 2. A játéktáblán megjelenik a szimbóluma, a játékos köre véget ér.
Alternatív forgatókönyv	1.A.1. A játékos nem lép, ideje a lépésre lejár.
Alternatív forgatókönyv	1.A.2. A program lép a játékos helyett.
Alternatív forgatókönyv	2.A.1 A játékos érvénytelen mezőre szeretné letenni a szimbólumát, a játékos újból próbálkozhat a lépéssel.
Alternatív forgatókönyv	2.B.1 A játékos lépésével előáll a nyerés feltétele, a játékos nyer, a játék véget ér.

Cím	Put down marker
Leírás	A gép köre során lép.
Aktorok	Controller
Főforgatókönyv	1. A gép elhelyezi szimbólumát egy mezőn.
Alternatív forgatókönyv	1.A.1. A gép lépésével előáll a nyerés feltétele, a gép nyer, a játék véget ér.

Cím	View remaining time
Leírás	A játékos megtekinti a lépése során hátralévő idejét.
Aktorok	Player
Főforgatókönyv	1. A rendszer kirajzolja a játékos lépése során hátralévő idejét. 2. A játékos megtekinti a lépése során hátralévő idejét.

Cím	Save current game
Leírás	A játékos elmenti a jelenlegi játékállást.
Aktorok	Player
Főforgatókönyv	1. A játékos rákattint a jelenlegi játékállás elmentésére. 2. A rendszer elmenti a jelenlegi játékállást.

Cím	Load previous game
Leírás	A játékos betölti az utoljára elmentett játékállást.
Aktorok	Player
Főforgatókönyv	1. A játékos rákattint az utoljára elmentett játékállás betöltésére. 2. A rendszer betölti az utoljára elmentett játékállást.
Alternatív forgatókönyv	2.1.A Nem található utoljára mentett játékállás, ezért a betöltés megghiúsul.

Kapus Ádám