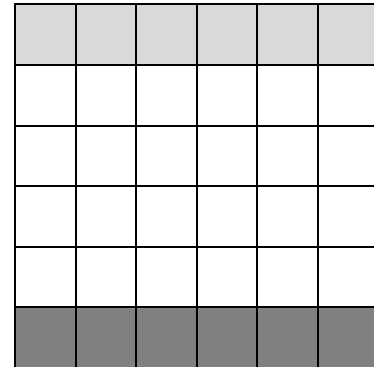


A	Név:	Neptun kód:
Gép sorszám:	Értékelő oktató:	Jegy:

Feladat: Roham

Készítsünk egy *WPF* grafikus felületű, *MVVM* architektúrájú alkalmazást a következő két személyes játékra. A játékot egy $n \times n$ -es táblán játsszák, kezdetben n -n figurával, amelyek a tábla két végén helyezkednek el 1-1 sorban, az ábrán látható módon. A figurákkal 4 irányban, **átlósan** lehet lépni (egy vagy 2 mezőt), illetve balra, jobbra, felfelé vagy lefelé lehet ütni (egy mezőt). A játékosok egyenként lépegetnek, bármelyik saját figurájukat kiválasztva. A játék célja, hogy az egyik játékos a másik játékos összes figuráját leüsse.

**Részfeladatok:**

- (2 pont)** A program jelenítse meg a játéktáblát, amelynek méretét a felhasználó adja meg három lehetőség közül választva (4×4 , 6×6 , vagy 8×8). A játékosoknak legyen lehetősége egymást váltva lépegetni, illetve ütni valamelyik táblán lévő bábuval. A program csak szabályos lépéseket tegyen lehetővé, illetve folyamatosan jelezze a képernyőn, melyik játékos következik, és hány bábuja van még a táblán.
- (1 pont)** A program ismerje fel, ha vége a játéknak, ekkor jelentse meg, melyik játékos győzött, majd automatikusan kezdődjön új játék. Lehesse továbbá bármikor új játékot kezdeni, akár a táblaméret megváltoztatásával.
- (1 pont)** Legyen lehetőség passzolni. Ha mindkét játékos egymást követően passzolt, automatikusan legyen vége a játéknak döntetlen eredménnyel.
- (1 pont)** Az alkalmazásban legyen lehetőség az aktuális játékállapot mentésére és a legutolsó mentés visszatöltésére külön perzisztencia réteg segítségével.

A megoldást az elfogadást követően **ZIP** formátumban fel kell tölteni a <https://assignment.elte.hu/> beadandókezelő rendszerbe.

Jó munkát!

B	Név:	Neptun kód:		
	Gép sorszám:	Értékelő oktató:	Jegy:	

Feladat: Whack-a-mole

Készítsünk egy *WPF* grafikus felületű, *MVVM* architektúrájú alkalmazást a népszerű *whack-a-mole* játékra. A játékos célja, hogy a pályán elhelyezkedő vakondlyukakból véletlenszerűen előbukkanó vakondokat leüsse, és ezzel minél több pontot gyűjtsön össze.

A játékot egy 5×5 -ös sakktábla szerű táblán játsszák, ahol minden második mező vakondlyuk, az ábrán látható módon. A játékos feladata, hogy rákattintson azokra a mezőkre, amelyekből éppen előbújik egy vakond.

H		H		H
	H		H	
H		H		H
	H		H	
H		H		H

Részfeladatok:

- (2 pont)** A program jelenítse meg a játéktáblát, amelyen minden másodpercenben jelenjen meg 40% valószínűséggel 2 vakond, 40% valószínűséggel 1 vakond, 20% valószínűséggel 0 vakond. A játékosnak 2 másodpercig van lehetősége a vakondokat leüthetni (a mezőre kattintva). A vakondok az idő letelte után vagy ha leütik őket, eltűnnek. Új vakond csak olyan lyukon jelenhet meg, amelyen éppen nincsen vakond.
- (1 pont)** Egy játék a programmal tartson 30 másodpercig, minden sikeresen leütött vakonddal 1 pontot szerezzen a játékos, azonban a téves ütések (ahol nincs vakond) -1 pontot érjenek. Az alkalmazás jelenítse meg hány másodperc játékidő van még hátra és ismerje fel, ha vége van a játéknak (letelt az idő). Ekkor jelenítse meg, hogy összesen hány pontot szerzett a játékos, majd automatikusan kezdődjön új játék.
- (1 pont)** Amennyiben egy vakond leütésre került, ugyanazon a lyukon vakond közvetlenül a következő alkalommal ne jelenhessen meg, de utána már újra igen.
- (1 pont)** Az alkalmazásban legyen lehetőség a 10 legtöbb pontot elért játék toplistájának megtekintésére (játék időpontja, pontszám). Ehhez minden játék végén perzisztens módon relációs adatbázisban kerüljön tárolásra az adott játék befejezési időpontja és az elért pontszám. Az adatbáziskezelést objektum-relációs leképezéssel, *Entity Framework* használatával végezd, a mentésért és betöltésért külön *perzisztencia* réteg feleljen.

A megoldást az elfogadást követően **ZIP** formátumban fel kell tölteni a <https://assignment.elte.hu/> beadandókezelő rendszerbe.

Jó munkát!

C	Név:	Neptun kód:
Gép sorszám:	Értékelő oktató:	Jegy:

Feladat: A róka és a sajt

Készítsünk egy *WinForms* grafikus felületű, *MV* architektúrájú alkalmazást a következő egyszemélyes játékra. A játékot egy $n \times n$ -es táblán játsszák, a játékos egy rókával van, amely kezdetben a tábla bal felső sarkában helyezkedik el, és $3n$ életerővel rendelkezik. A róka 4 irányban, balra, jobbra, felfelé vagy lefelé tud lépni, életereje pedig másodpercenként csökken 1-gyel. A pályán szabályos időközönként 1-1 sajt jelenik meg egy véletlenszerű mezőn, majd 2 másodperc után eltűnik. A róka fel tudja venni a pályán lévő sajtot, amitől 2-vel nő az életereje.



A játék célja, hogy a róka minél több sajtot felvegyen, és ne csökkenjen 0-ra az életereje.

Részfeladatok:

- (2 pont)** A program jelenítse meg a játéktáblát, amelynek méretét a felhasználó adja meg három lehetőség közül választva (6×6 , 8×8 , vagy 10×10). A játékosnak legyen lehetősége lépegetni a rókával, illetve felvenni a pályán lévő sajtot, amely ekkor eltűnik a pályáról. A sajt 2 másodpercenként új helyen jelenik meg, attól függetlenül, hogy a róka megette-e. A program csak szabályos lépéseket tegyen lehetővé, illetve folyamatosan jelezze a képernyőn, hogy mennyi életereje van még a rókának.
- (1 pont)** A program ismerje fel, ha vége a játéknak, ekkor jelenítse meg, hogy hány sajtot vett fel a róka, majd automatikusan kezdődjön új játék. Lehessen továbbá bármikor új játékot kezdeni, akár a táblaméret megváltoztatásával.
- (1 pont)** A sajt ne tűnjön el a pályáról a 2 másodperc leteltével, hanem járjon le a szavatossága (ezt jelezzük is a táblán). A romlott sajt nem tűnik el a tábláról, és felszedve 3-mal csökkenti a róka életerejét. 1-1 jó sajt továbbra is jelenjen meg a pályán, de nem kerülhet olyan mezőre, ahol romlott sajt van. Ha a jó sajt nem tud hol megjelenni (minden mezőn vagy a róka vagy romlott sajt van), akkor vége a játéknak.
- (1 pont)** Az alkalmazásban legyen lehetőség az aktuális játékállapot mentésére és a korábbi mentések visszatöltésére külön perzisztencia réteg segítségével.

A megoldást az elfogadást követően **ZIP** formátumban fel kell tölteni a <https://assignment.elte.hu/> beadandókezelő rendszerbe.

Jó munkát!