# Guess the price

## 1. feladat

View:

A Mainwindow-n belül a táblázat alapján alakítsd ki a sorokat és az oszlopokat. Az első feladatban csak egyes cellákat kell elkészíteni (lesz még „üres” cella).

A „cellákban” található szavak a ViewModel-ben használt Property-k nevét jelzik (label szövege ezen property-k értéke legyen) , ahol pedig „gomb:” előtag szerepel ott a ViewModel-ben használt RelayCommand nevét (és ezek gombok legyenek).

|  |  |
| --- | --- |
|  | |
| MinPrice | MaxPrice |
| ActualPrice | |
| gomb:NextRoundCommand | |
|  |  |

EventArgs:

Hozz létre (int) minimum ár, (int) maximum ár, (string) tényleges ár publikus adattagot, a konstruktorban add át a paraméterül kapott értékeket a változóknak.

Model:

A Model-ben szerepeljen (int) minimum ár, (int) maximum ár, (string) tényleges ár és egy Random private adattag.

Hozz létre egy publikus event-et „SetUpNextRound” néven, használjon GuessThePriceEventArgs-ot.

A konstruktorban inicializáld a Random adattagot.

Hozz létre egy void NextRound metódust, a metódusban generálj 1-200 közötti számot a min árnak, 900-1200 közötti értéket a max árnak és min-max közötti árat a tényleges árnak.  
Hívd meg a SetUpNextRound event Invoke metódusát és paraméterezd az EventArgs-ot.

ViewModel:

A ViewModel-ben szerepeljen Model, (int) minimum ár, (int) maximum ár, és (string) tényleges ár private adattag. Az utóbbi háromnak készíts public Property-t a táblázatban megadott nevek alapján, a set résznél ne felejtsd el „update”-elni is az adattagot.

Hozd létre a RelayCommand -ot (get,set), a táblázat alapján.

A konstruktorban inicializáld a RelayCommand -ot és hívd meg vele a Model NextRound metódusát.  
 Iratkozz fel a Model SetUpNextRound event-re egy metódussal amelyben állítsd be a min ár, max ár és a tényleges ár értékét a paraméterül kapott GuessThePriceEventArgs alapján (a property-nek adj értéket).   
Végül hívd meg a Model NextRound metódusát.

## 2. feladat

|  |  |
| --- | --- |
|  | |
| MinPrice | MaxPrice |
| ActualPrice | |
| gomb:NextRoundCommand | |
| <Opponents’ guess>  OpPriceGuess |  |

View:

Hozd létre a táblázat alapján a kettő label-t az egyik csak az „Opponents’ guess:” szöveget tartalmazza, a másik pedig mint az első feladatban az OpPriceGuess property értékét mutassa (rendezéshez segíthet a stackpanel használata).

EventArgs:

Hozz létre egy public adattagot az ellenfél tippjére és ezt is a kapott paraméter alapján állítsd be.

Model:

Hozz létre egy private adattagot az ellenfél tippjére.  
A NextRound metódusban adj értéket az új adattagnak, a min és max érték közötti random (egészítsd ki a SetUpNextRound eventargs-ját is ezzel az új értékkel).

ViewModel:

Hozz létre egy private adattagot az ellenfél tippjére, és hozz létre egy public property-t a táblázat alapján.  
A SetUpNextRound-et lekezelő metódust egészítsd ki az új adattag értékadásával.

## 3. feladat

|  |  |
| --- | --- |
|  | |
| MinPrice | MaxPrice |
| ActualPrice | |
| gomb:NextRoundCommand | |
| <Opponents’ guess>  OpPriceGuess | PlayerPriceGuess  gomb: PlayerGuessCommand |

View:

Hozd létre a táblázat alapján a textbox-t (textbox ’Text’-et bindold) és a gombot, a gomb szövege legyen „Make Guess”(rendezéshez segíthet a stackpanel használata).

Model:

Hozz létre egy publikus event-et AfterGuess néven, használjon GuessThePriceEventArgs-ot.

Hozz létre egy void AfterPlayerGuess metódust, ami paraméterül egy int –et kapjon, ez a szám lesz a játékos tippje. A játék úgy működik, hogy ha mindkét játékos tippje kisebb, mint a tényleges „ár ” akkor az nyer, aki közelebb van az árhoz. Ha csak az egyik játékos van a tényleges ár értéke alatt, akkor az a játékos nyert. Ha mindkét játékos tippje több mint a tényleges ár senki sem nyert.  
Csinálj három ágat, az előző esetek alapján, a játékos nyer, az ellenfél nyer és hogy nincs nyertes. Mindegyik ágban hívd meg az „AfterGuess” event Invoke metódusát és paraméterezd az eventargs-ot.

A NextRound metódusban, a SetUpNextRound hívásánál ne a tényleges árat adjuk át az eventargs-nak hanem „????” –t, a tényleges árat csak a játékos tippje után mutatjuk majd

ViewModel:

Hozz létre egy private adattagot a játékos tippjére, és hozz létre egy public property-t a táblázat alapján.

Hozd létre a RelayCommand -ot (get,set), a táblázat alapján.

A konstruktorban iratkozz fel a Model AfterGuess event-jére és az event-et kezelő metódusa a kapott eventargs-ból állítsa be a tényleges ár értékét.

Az új Command-ot is inicializáld és hívja meg a model AfterPlayerGuess metódusát. A PlayerPriceGuess-t kezdetben állítsd 0-ra.

## 4. feladat

|  |  |
| --- | --- |
| WinnerText | |
| MinPrice | MaxPrice |
| ActualPrice | |
| gomb:NextRoundCommand | |
| <Opponents’ guess>  OpPriceGuess | PlayerPriceGuess  gomb: PlayerGuessCommand |

View:

Hozd létre a táblázat alapján egy label-t ami a WinnerText property értékét mutassa.

EventArgs:

Hozz létre egy public adattagot a nyertes szövegnek és ezt is a kapott paraméter alapján állítsd be.

Model:

A NextRound metódusban egészítsd ki a SetUpNextRound hívás eventargs-ját egy üres stringgel, mivel a kör elején nem akarjuk látni a szöveget.

Az AfterPlayerGuess event hívásait is egészítsd ki a következő szövegekkel "Player Won this round!”, "Opponent Won this round!", "Both players went over!".

ViewModel:

Hozz létre egy private adattagot a nyertes szövegnek, és hozz létre egy public property-t a táblázat alapján.

Hozz létre egy private adattagot és a hozzátartozó property-t a „NextRound” gomb „IsEnabled” beállítására. A NextRound metódusban ezt állítsd „false”-ra, az AfterGuess-t lekezelő metódusban állítsd be „true”-ra.

Mindkét event-et lekezelő metódushoz add hozzá a nyertes szöveg értékadását az eventargs-ból.

## 5. feladat

Model:

Hozz létre 3 private adattagot, a játékos pontjaira, ellenfél pontjaira és a körök számára.  
A pontok eléréséért hozz létre 2 public property-t (pl.: PlayerPoints, OpponentPoints) ezek adják vissza a változók értékét (csak get).  
Hozz létre egy EventArgs-ot használó GameEnd event-et.  
A pontokat állítsd 0-ra a körök számát 1-re a konstruktorban.  
A NextRound metódus hívásában növeld a körök számát eggyel.  
Az AfterPlayerGuess metódusban, ha a játékos nyert az ő pontját, ha az ellenfél akkor az ellenfelét növeljük eggyel, harmadik esetben senki se kapjon pontot.  
Az AfterPlayerGuess végén nézd meg hogy a körök száma elérte e a 10-et, ha igen hívd meg a GameEnd Invoke metódusát üres EventArgs-al.

ViewModel:

Iratkozz fel a Model GameEnd event-jére. Az event-et kezelő metódusban vizsgáld meg hogy ki nyert (esetleg döntetlen) és egy messagebox-ban jelenítsd meg az alábbi szövegek egyikét:

"After 10 rounds you have won with "+\_model.PlayerPoints+" points!"

"After 10 rounds your opponent has won with " + \_model.OpponentPoints + " points!"

"After 10 rounds both player have " + \_model.PlayerPoints + " points, so it is a draw!"