Dokumentation - Brädspelet Grupp 2- Swedish test mafia Programmering för testare HT24, labb 2

## Hur man spelar spelet:

När programmet startas möts spelaren av en meny med fyra val där de kan välja att spela följande spel:

- 1. Spela tre i rad mot datorn
- 2. Spela tre i rad mot en motståndare
- 3. Spela fyra i rad mot en motståndare
- 4. Spela fem i rad mot en motståndare

Därefter får spelaren välja om den vill skriva in ett eget namn eller få ett default-namn, *Spelare 1 / Spelare 2*.

Spelaren får sedan välja vilken symbol de vill spela med där de kan välja mellan X eller O. Därefter slumpas det fram vem som får inleda omgången oavsett om du spelar mot datorn eller en annan motståndare.

Brädet presenteras sedan i terminalen där rader och kolumner är presenterade med siffror. Spelaren får sedan först skriva in vilken rad och sedan vilken kolumn som de vill placera sin symbol. (Se bild nedan.)



Därefter läggs en symbol varannan gång och målet med spelet är att placera sina symboler i tre/fyra/fem i rad (beroende på vilket spel som valts). Om spelaren försöker placera på en upptagen ruta eller en ruta utanför spelplanen presenteras ett felmeddelande.

#### Vinstvillkor:

Om någon av spelarna har fått tre/fyra/fem i rad (lodrätt, vågrätt eller diagonalt) avslutas omgången och spelet presenterar vinnaren. Om brädet är fullt avslutas omgången och spelet presenterar att omgången är oavgjord.

Efter avslutad omgång presenteras spelaren med tre val:

- 1. Börja ny omgång
- **2.** Tillbaka till menyn
- **3.** Avsluta spel

### Poängöversikt:

Vid avslutad omgång presenteras spelarnas vinster för aktuellt spel, samt spelarnas vinster totalt sedan spelet startade.

## Klasser:

#### Main:

I Main startas spelet och använder klassen Menu för att visa en meny med de olika spelvalen. Spelaren gör sitt val och en spelarlista skapas. En instans av klassen *Game* skapas. Har metoder för att skapa en spelarlista, skapa och hantera en spelarlista för "vilande spelare" (om spelare 1 vill spela mot datorn i stället för motståndare och vice versa), samt skickar oss vidare till själva spelomgången.

#### Menu:

Klassen *Menu* finns spelets meny och dess fyra olika val för de olika spellägena, *Tre i rad*, *Fyra i rad* och *Fem i rad*. Menyn är uppbyggd av en switch-sats med felhantering om spelaren matar in ogiltiga symboler/siffror. I menyn är olika spelen kopplade till olika ID:n som är unika för varie spel.

### Player:

I klassen *Player* finns metoder för att välja namn på spelaren/spelarna, felhantering vid inmatning av namn, och hantering av spelarens vinststatistik. Det finns även en metod som behandlar att spelaren ska kunna välja vilken symbol de vill spela med.

## Game:

I klassen *Game* hanteras spelflödet i de olika spelen. Denna klassen använder metoder från *Player*-klassen för att hämta aktuell spelare, och från *Board*-klassen för att skapa ett spelbräde, kontrollera brädets status, placera symbol och skriva ut aktuellt bräde i terminalen. Det finns metoder som slumpar ordningen på spelarna, byter spelartur, hantera vinstkriterier, hanterar dators beteende om motspelare skulle vara en dator, samt vad som händer vid avslutad omgång (en meny med val att spela igen, avsluta spel eller gå tillbaka till startmeny).

Ett välkomstmeddelande och med instruktioner skrivs ut vid start av varje spel och ett exit-meddelande vid spelavslut.

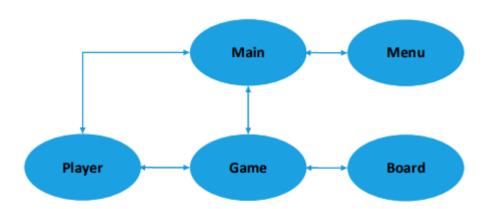
### Board:

I klassen *Board* hanteras spelbrädet, med metoder för att skapa ett bräde, skriva ut brädet i terminalen, kontrollera om en plats är giltig (finns på brädet) samt om en plats är ledig. Även metoder för att placera en symbol samt beräknar om vinstvillkor uppfyllts (kontrollerar om brädet har tre/fyra/fem i rad).

#### **GlobalTools**

GlobalTools innehåller ett inputfilter som endast släpper igenom giltiga värden vid användarinput. Används i klasserna *Main, Menu, Game* och *Player*.

# Tillståndsdiagram



# Klassdiagram

