

Rättningsguide laboration 4 D0010E för labbhandledare – Pandemiversion med Canvas och Zoom–

- **OBS 1!** Ska göras i de grupper Håkan bestämt (normalt i par).
- **OBS 2!** Rätta endast helt klara arbeten. **Avbryt** om detta inte är fallet.
- Gå igenom allt nedan med båda studenterna.
- Ställ ungefär lika många frågor till båda gruppmedlemmar.
- Notera under rättningen (t ex på ett papper) helt kort brister då de upptäcks så du kommer ihåg dem. Brister kan t ex finnas i:
 - Programmets funktion och utseende (GUI)
 - Övergripande struktur och programmeringsdetaljer
 - Dokumentation (inre kommentarer och javadoc)
 - Gruppmedlemmarnas förmåga att beskriva och förklara
- Sammanfatta sen bristerna för alla i gruppen så det blir helt klart vad som måste åtgärdas. Det är viktigt (för inläringen) att de förstår bristerna.
 - Kom ihåg att anteckna bristerna i Canvas.
- Efter sammanfattningen får de börja åtgärda brister (inte under tiden). Det är viktigt för deras inläring att de inte åtgärdar fel och brister medan du rättar utan efteråt och själva. (Vårt mål är ju inte att deras program ska vara felfritt utan att de ska lära sig.)
- **OBS 3!** Alla i arbetsgruppen står för allt i programmet. Alla ska kunna beskriva och förklara allt, såväl sånt de jobbat mycket med som sånt de jobbat mindre med.

Om det blir tydligt att någon eller några av gruppmedlemmarna inte har riktig koll på vad de gjort så **avbryt rättningen** och uppmana hela gruppen att se till att hela gruppen förstår/kan förklara innan de åter redovisar. Ställ inte för svåra frågor.

Om du är osäker på hur du ska rätta så rådfråga mig eller annan labbhandledare.

A. Test och demo

1. Be studenterna att helt kort testköra med två klienter och spela så det i stora drag framgår att spelets allmänna funktioner och GUI:t (t ex

knappars avaktivering/aktivering, markeringar på spelbrädet och meddelandena) överensstämmer med det som efterfrågas.

- a. Om GUI:t visuellt avviker från bilden i labspecen så be om en noggrann förklaring samt följ upp att GUI:t sen fungerar som studenten beskrivit.
- b. OBS! GUI:t ska vara ungefär som i labspecen. Stora avvikelser ska inte godkännas.

2. Bes studenterna ändra konstanterna

UNIT_SIZE i GamePanel,
DEFAULT_SIZE i GomokuGameState samt
INROW i GameGrid

lite grann och kolla att spelet fortfarande fungerar.

3. Kolla (med de nya värdena på konstanterna som sattes ovan) att GUI-komponenter och GUI-komponenters relativa lägen anpassar sig om storlek och form på spelfönstret ändras genom att dra och förändra fönstret med musen.
 - a. Håll storleken ganska stor och formen inte alltför tunn/smal. GUI:t ska anpassa sig och fungera lika bra som förut vid i vart fall små förändringar.
4. Be studenterna att testa att åter spela med denna ändrade storlek och form på fönstret.
5. Väl slumpmässigt ut två hörnrutor :

En av {*övre vänstra, undre högra*}.
En av {*lägre vänstra, övre högra*}.

Kolla att det går att spela och vinna med rader som inkluderar var och en av dessa två hörnrutor.

B. Javadoc

Den givna koden innehåller redan nästan all javadoc(!) (I labb 5 ingår att skapa och generera websidor av stora mängder javadoc.)

Javadoc för klasserna ska kompletteras med @author-element så det klart framgår vilka som skrivit klassen.

Kolla att javadock för GameGrid har tre publika variabler som är final.

Kolla att kommentarer inte uttryckligen innehåller numeriska konstanter från koden (eller referenser in till andra dolda detaljer i koden).

(Javadoc ska beskriva vad en klass representerar och vad metoder gör, dvs vad dess instanser/dynamiska objekt är och vad man kan göra med dem. Javadoc får inte innehålla implementationsdetaljer; den ska ge en *klientvy* av koden, inte en implementationsvy).

C. Kodgranskning

- *Kod ska ligga i rätt paket och vara utformad efter industristandard.*
- *Kommunikation mellan klienter ska uteslutande göras av den kod som finns i Gomoku.jar (fråga om det är så).*
- *Kolla att koden inte uttryckligen innehåller andra numeriska konstanter än -1, 0 och 1 (för det försvårar framtida ändringar).*
 - *Deklarera och använd istället privata variabler som är final, precis som INROW, UNIT_SIZE och DEFAULT_SIZE.*
- *För läsbarhetens skull ska namn på variabler och metoder säga något väsentligt och relevant om det namngivna.*
- *Allt ska vara private som labspecen/givna javadoc:en inte uttryckligen säger ska vara nåt annat.*

C1. GameGrid

1. Ska ha tre publika variabler EMPTY, ME och OTHER som är final. (Enumerations är OK.)
2. Be studenterna motivera att deras metod isWinner fungerar.

C2. GomokuGameState

1. Be studenterna förklara vilken skillnaden är mellan metoden move i denna klass och move i GameGrid.
2. Och varifrån anropas move i denna klass?

C3. GamePanel

1. Vad gör paintComponent?
2. Nu är rutorna på spelbrädet kvadrater. Vad behöver ändras för att rutorna istället ska kunna vara rektanglar (och spelet sen fungerar som förut)?

C4. GomokuGUI

1. Ingen egen kod för kommunikation ska finnas. Kommunikationskod finns i Gomoku.jar.
2. Be studenterna peka ut en av klassens fyra anonyma lyssnare och förklara vad den har för funktion.