

Tidigare har det stått: "Dessutom ska koden vara strukturerad, dokumenterad och ni ska visa på att ni förstår och kan använda er av objektorienterad programmering samt grafik med Swing i Java. "

Nu läggs det till som ett explicit krav för både **godkänt** och **väl godkänt** att:

\*Alla klasser och publika metoder ska dokumenteras med Javadoc enligt standard

---

Det har inte funnits någon uttalad deadline i kravspecifikationen för utvecklarna. Deadline för kodning sätts nu till måndag 19e december klo 16.00. Då ska varje grupp ha skickat en länk med sitt GitHub-repo till Martin (och Elias) i sin privata kanal på Systemutvecklingsprojekt-slacken. All kod ska då ligga i master-branschen. Det är den koden Martin kommer att bedöma. Commits till master efter 19e december klo 16.00 kommer inte att tas i beaktning.

Förutom att koden ska ligga uppe på GitHub ska ni även skicka en ZIP-fil via Slack som ska innehålla:

- UML-diagram
  - Ta med namn på klasser du skapat, relationer samt klasser/gränssnitt du direkt ärver från/implementerar
  - Ta **inte** med klassens attribut och metoder
- Dokumentation med *Javadoc*

---

Som ytterligare krav för både **godkänt** och **väl godkänt** så läggs nu till att:

- All kod i ditt spel ska vara skriven av gruppen själva
- Redovisa ärligt hjälp ni fått och källor ni använt er av

Det är dock tillåtet att be om och ge hjälp för ökad förståelse. Så länge som man är ärlig och inte tar någon annans kod rakt av eller skriver kod åt någon annan.

---

Som ett av numera två valbara VG-krav ställs nu att att "Poängen för respektive tärningskombination ska automatiskt räknas ut och för varje slag ska programmet visa vilka kombinationer som du kan sätta poängen på och hur mycket poäng du får"

# Uppdaterad kravlista för utvecklingsdelen av projektet (markerat med rött är det som är nytt)

För både **godkänt** och **väl godkänt** ska spelet minst uppfylla följande krav:

- Ett enklare UML-diagram över klasserna i spelet
- Spelet visar fem tärningar och ett protokoll där poäng för varje spelare och varje tärningskombination visas
- Användaren ska ha möjlighet att välja vilka tärningar som ska slås om
- Poängen för respektive tärningskombination ska automatiskt räknas ut
- Efter alla tärningskombinationer är ifyllda så ska spelet räkna ut eventuell bonuspoäng och meddela den totala poängen
- Alla klasser och publika metoder ska dokumenteras med Javadoc enligt standard
- All kod i ditt spel ska vara skriven av gruppen själva
- Redovisa ärligt hjälp ni fått och källor ni använt er av

Dessutom ska koden vara strukturerad och du ska visa på att du förstår och kan använda dig av objektorienterad programmering samt grafik med Swing i Java. Onödig kodupprening ska i största mån undvikas.

För betyget **godkänt** ska spelet minst uppfylla följande krav:

- Tärningarnas värden visas med text
- Spelet spelas i ordning uppifrån och ner
- Spelet spelas med en mänsklig spelare

För betyget **väl godkänt** ska spelet minst uppfylla följande krav:

- Tärningarnas värden visas med bilder
- Spelet ska fråga om namn på spelare och visa vems tur det är att slå tärningarna
- Spelet har stöd för godtyckligt antal spelare

samt minst **två** av följande krav

- En spelare kan vara mänsklig eller dator
- Spelet kan spelas först övre delen i valfri ordning, sedan nedre delen i valfri ordning
- Spelet kan spelas i valfri ordning
- Poängen för respektive tärningskombination ska automatiskt räknas ut och för varje slag ska programmet visa vilka kombinationer som du kan sätta poängen på och hur mycket poäng du får

Dessutom ska koden tydligt följa en etablerad stilguide. Du kan redogöra och argumentera för att din kod är av hög kvalitet.