

1. **Vad är skillnaden på GoldModel och GameModel, och hur är de dem relaterade till varandra?**

GameModel står för grunden till alla spel som skapas, alltså de metoder som alla spel har gemensamt och kommer behöva. GoldModel är en modell på ett specifikt spel. Här återfinns de metoder som gör spelen unika.

Relationen mellan GameModel och GoldModel är att GoldModel ärver GameModel. GoldModel är subklass till superklassen GameModel.

2. **Vilken metod utför själva ritandet av 'Gold coins' i spelet 'Gold coins'?**

Metoden setGameboardState utför ritandet av Gold coins.

Metoden finns i GameModel eftersom att alla spel förmodligen kommer ändra på GameBoard's state.

3. **Vilken metod är det som anropar metoden som ritar ut 'Gold Coins'?**

Det är metoden 'addCoin' som anropar metoden setGameboardState vilket sätter ut en 'coin tile' på en slumpmässigt utvald position.

4. **Vad är syftet med GameFactory och hur kan GameFactory utvidgas för att få med 'Snake game'? (Endast ett kort svar behövs.)**

Det är en klass som innehåller de spel som är implementerade. Vill man implementera ett nytt spel som till exempel snake räcker det att utöka if-satsen med en parameter som innehåller "snake" samt lägga till "snake" i arrayen med spelnamn.

5. **Var återfinns beräkningen och kontrollen av de rörelser som 'Gold eaters' gör?**

Dessa återfinns i klassen GoldModel. I metoden 'updateDirection' återfinns kontrollen av rörelsen där programmet kollar på senaste inmatningen av användaren och sätter hur förändringen ska vara i både x- och y-led. I metoden 'getNextCollectorPos' återfinns beräkningen av nästa position som utför beräkningen utifrån datan av updateDirection.

I metoden gameUpdate görs sedan förflyttningen och de kontroller som t.ex. om spelaren har lyckats ta ett mynt eller nått kanten på spelplanen.

6. **Hur lagras ett spelbräde och i vilken klass?**

Det lagras som två instansvariabler i GameModel. Ett array med arrayer som innehåller spel informationen. En variabel som innehåller dimensionerna på brädet.

7. **Beskriv arbetsflödet av programmet.**  
**(Detta kan göras genom att skissa på hur kontrollen flödar genom olika klasser och metoder.)**

- 7.1. När programmet körs så börjar det i klassen Main. I Main så kommer det skapas ett fönster och ett GUIView med en GameFactory.
- 7.2. I GUIView så lägger programmet till ett gränssnitt i fönstret som innehåller en lista med spel samt en knapp att starta det valda spelet. Listan genereras genom att använda GameFactorys 'getNames' metod som returnerar listan med alla spel.
- 7.3. Utifrån vilket spel användaren väljer laddas det specifika spelets klass när spelaren tryckt på starta knappen. Då skickas en händelse som anropar metoden 'startgame' i GameFactory som returnerar det specifika spelets klass och startar en ny tråd.
- 7.4. Spelet loopar sedan runt och repaintar fönstret tills det slutar.

8. **Ge en plan för hur ni tänker fortsätta med deluppgift 2. Vilka klasser kommer ni att skriva och vad kan ni återanvända?**

Vi behöver skriva en ny class som ärver av GameModel på samma sätt som GoldModel gör. Alla andra klasser går att återanvända men vi behöver göra lite tillägg i GameFactory (som nämndes ovan i fråga 4) för att göra det möjligt att välja det nya spelet.