

- U4TG1: Beskrivning av skatter och fällor ska finnas
 - Typer av objekt:
 - Gem tar en position/knapp, den har en form av en punkt eftersom den bara har ett liv. Man får ett poäng från att gräva denna objekt.
 - Jewelry tar två positioner/knappar, den har en form som ser ut som ett i, den har två liv. Man får ett poäng från att gräva denna objekt.
 - Goldbar tar tre positioner/knappar, den har en form som ser ut som ett L, den har tre liv. Man får ett poäng från att gräva denna objekt.
 - Special Stone tar fyra positioner/knappar, den har en form som ser ut som en fyrkant, den har fyra liv. Man får ett poäng från att gräva denna objekt.
 - Treasure Chest tar fem positioner/knappar, den har en form som ser ut som ett J, den har fem liv. Man får två poäng från att gräva denna objekt.
 - Trap den tar ett position/knapp, den har en form som ser ut som ett punkt eftersom den bara har ett liv. Om man trycker på fällan och har poäng så förlorar man ett poäng, men om man inte har poäng alls, ändra din tur till nästa spelare.
- U4TG2: Motivera valet av interface och/eller generalisering för att hantera olika typer av nedgrävda föremål (skatter och fällor)
 - För den uppgiften valde jag att använda mig av en abstrakt klass. Anledningen till detta valet har med att klasser som ärver från abstracta klassen får redan metoderna i abstrakta klassen färdig implementerade, detta spara mycket tid och arbete, då man slipper gå i varje klass och skriva samma metod. Klasser som ärver från den abstrakta klassen ser också mer läsbara ut, då metoderna som de ärver finns inte synliga i klassen.
 - Nackdelen med att använda abstract klasser är att subklasserna endast kan ärva från en abstrakt klass, vilket gör det jobbigt om man vill använda subklasserna på andra sätt genom att ge dem andra funktionalitet.

Detta problem träffade jag när jag ville skapa min trap-klass. Jag ville att min trap-klass skulle ha funktionaliteten att byta tur och reducera spelaren poäng, men detta betyder att alla de andra klasserna av typen skatt skulle få denna funktionaliteten, vilket jag inte ville, så jag var tvungen att lösa detta på annat sätt. Trap-klassen var tvungen att ärva från min abstracta klass för att kunna lägga till på brädan tillsammans med skatterna.
- U4TG3: Beskrivning av tur omgångar med aktivitetsdiagram
 - Min lösning till detta problem är att när spelaren trycker på en knapp, så ska det finnas en metod som håller koll på vad som finns under knappen. Om det inte fanns något objekt under knappen så ska turen ändras till nästa spelare, men om det finns ett objekt så ska det finnas en metod som hanterar prick. Denna metod som hanterar prick ska kolla efter vad för objekt som blev

prickat, skatt eller fälla? Om det var fälla så ska spelarens poäng kontrolleras. Om spelaren har poäng, så ska den reduceras och turen ska ändra till nästa spelare. Men om det skulle vara så att spelaren inte har poäng, så ska turen ändras till nästa spelare.

- U4TG4: Förbered minst två frågor till den grupp som redovisar samtidigt.
 - Min första fråga är, hur gruppen löste problemet med att lägga ut skatterna i en form
 - Min andra fråga är hur gruppen löste tur omgång mellan spelare 1 och spelare 2.