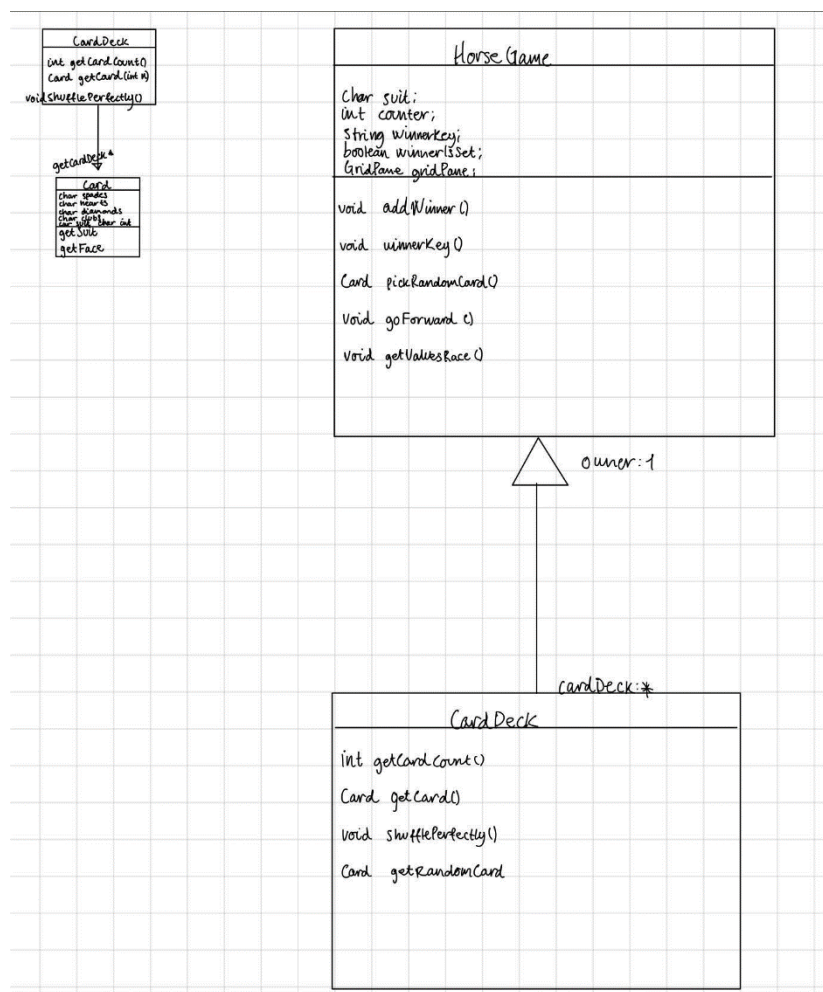


Rapport

Beskrivelse:

Vi har laget spillet hesteveddeløp. Det handler om å være førstemann til mål. Man kommer til mål ved at flest kort av egen sort blir trukket. Kortene blir trukket i en tilfeldig rekkefølge. Vi har fire hovedklasser som hver for seg deler opp logikken i spillet. De er Card, CardDeck, HorseGame og HorseGameController. Vi har også en fxml fil, en app og et interface. Hestene sin posisjon er basert på deres verdi i et hashmap hvor hver hest er en egen nøkkel. Når sorten til en hest/esset blir trukket økes verdien til nøklene/hestene og hesten sin posisjon blir flyttet. Vi har brukt klassene Card og CardDeck fra tidligere øvinger for å forenkle arbeidet vårt og ikke gjøre ting flere ganger. Vi har skrevet mer logikk til spillet enn det som brukes fordi vi har ikke fått det til å funke innenfor tiden, men vi har en visjon om å komplementere spillet med flere metoder i fremtiden. Dette skal jo brukes på fest så det må være skikkelig.



1. Hvilke deler av pensum i emnet dekkes i prosjektet, og på hvilken måte? (For eksempel bruk av arv, interface, delegering osv.)

Vi bruker arv i horsegame til å arve carddeck. Vi bruker også arv i appen vår, da arver vi fra den innebygde application. Vi bruker også interface til addWinner metoden. Ellers så bruker vi kun mer eller mindre programmeringslogikk.

2. Dersom deler av pensum ikke er dekket i prosjektet deres, hvordan kunne dere brukt disse delene av pensum i appen?

Vi har ikke delegering eller observatør/ observert. Delegering kunne vi brukt for å delegerer en make carddeck metode til carddeck/ card, men i stedet bruker vi arv. Observatør/ observert er ikke så relevant fordi vi ikke bryr oss om brukerens interaksjon med brukergrensesnittet.

3. Hvordan forholder koden deres seg til Model-View-Controller-prinsippet? (Merk: det er ikke nødvendig at koden er helt perfekt i forhold til Model-View-Controller standarder. Det er mulig (og bra) å reflektere rundt svakheter i egen kode)

HorseGame, som er logikklassen, har vi det meste av logikk i. Vi har noe logikk i kontrollerklassen, men dette er metoder som kun blir kalt på når brukeren interagerer med brukergrensesnittet. Ellers er klassene våre separert.

4. Hvordan har dere gått frem når dere skulle teste appen deres, og hvorfor har dere valgt de testene dere har? Har dere testet alle deler av koden? Hvis ikke, hvordan har dere prioritert hvilke deler som testes og ikke? (Her er tanken at dere skal reflektere rundt egen bruk av tester)

Vi har veldig få metoder med argumenter så vi har testet metodelogikken i logikklassen vår slik at vi sjekker korrekt tilstandsending. Vi har testet de viktige metodene i logikklassen, samt filbehandlingsklassen og metodene i cardklassen. Vi har ikke testet fxml filen og appklassen. Fxml filen testet vi ikke siden vi ble sterkt frarådet av studass.