Del 1: Beskrivelse 150-200 ord

Applikasjonen jeg har laget heter «Airline Manager». Det er et spill der man oppretter et flyselskap og oppgaven er å utvide selskapet med nye fly og reise rundt hele verden. Spillet er inspirert av «Pocket Planes» (Wikipedia, 2022). Man starter med en kapital som tillater at man kan kjøpe sitt første fly og begynne å frakte passasjerer til de nærmeste flyplassene. Hver passasjer betaler en viss sum for reisen. Denne er beregnet basert på avstanden til destinasjonen. Etter hvert som man tjener penger, kan man kjøpe seg flere fly. Det finnes ulike flytyper som kan ha med flere passasjerer, fly lengre, kjappere, og mere effektivt. Når man starter en flytur beregnes flytiden basert på avstand og hvor fort flyet kan fly. Når flyet har landet mottar man pengene fra passasjerene som har ankommet riktig flyplass. Har man med passasjerer som skal videre blir de værende i flyet.

På hver flyplass er det et maks antall reisende. De skal til forskjellige destinasjoner så det kan være at man ikke klarer å fylle opp flyet med passasjerer for å maksimere profitt. Ønsker man å fylle opp flyet med passasjerer som skal til samme destinasjon, vil flyplassene oppdatere reisende det hvert femte minutt.

Del 2: Kun ett diagram

Del 3:

1. Hvilke deler av pensum i emnet dekkes i prosjektet, og på hvilken måte? (For eksempel bruk av arv, interface, delegering osv.)
   1. Comparator / Comparable
      1. Airport
      2. Distancecomparator
   2. Arv
      1. City
   3. Iterator
      1. PlaneIterator
   4. Delegering
      1. Klokke
      2. FlightDistance
   5. Interface
      1. GameSave
      2. GameFileLoader
      3. SecondClockListener
   6. Observable/Observed
      1. Klokke, fly kontroller og spill
2. Dersom deler av pensum ikke er dekket i prosjektet deres, hvordan kunne dere brukt disse delene av pensum i appen?
   1. Abstact ??
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 for å få full uttelling på dette spørsmålet. Det er mulig (og bra) å reflektere rundt svakheter i egen kode)
   1. Ingen intern tilstand annet enn spillet
   2. Har prøvd å flytte all logikk til Game (Se over hva som eventuelt kan endres på..
   3. Usikker på hvordan man skal implementere en klokke til spillet. Jeg ville ikke koble spillet direkte til kontrollen, så jeg tenkte det beste var å gjøre controllen en SecondClockListener og at klokka kan kalle på metoden tick for å holde kontrollen oppdatert.
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)
   1. Ikke valgt å teste gettere og settere da disse er enkle.
   2. Tester de viktigste delene av spillet. Passe på at interne regler følges, så ikke et flyselskap kan kjøpe fly uten penger, ta av uten penger, at reisende oppdateres, at userInput ikke er ulovlige, at posisjonen til objekter er riktig (Passasjer), regler for takeoff og landing for å unngå at andre deler av koden feiler.
   3. Har ikke klart å få en test til å funke. Får visuelt riktig resultat, men ikke kodevis.
5. Har dere møtt på noen utfordringer i løpet av prosjektet? Hva ville dere gjort annerledes en annen gang?
   1. Tanken var å opprette flyene og airline ved load samtidig, og kun tillate at kontrukøren kunne ta inn fly uten betaling, men det var ingen god måte å gjøre det på med restriksjonene rundt filskriving og lesing. Derfor la jeg på at metodene for å reopprette data måtte ha med signaturen til en gamesavehandler for å fungere.
   2. Koden er preget av at jeg ikke har mye erfaring med javaFX, og at noe funksjonalitet ville vært skrevet på en helt annen måte hadde jeg visst mulighetene til JavaFX. Visste f.eks ikke at «button» kunne ta vare på objektet i seg.
   3. Burde definitivt ha valgt et mindre prosjekt. Ble massivt, med mange filer, og koblinger som gjorde at filbehandlingen er knotete. Har gjort klart for funksjonalitet jeg ikke rekker å lage, som liveries til flyene. Men det var noe jeg var interessert i å få til å funke, og ikke noe som var kjedelig å jobbe med.

Referanser

Wikipedia. (2022, April 20). *Pocket Planes*. Hentet fra Wikipedia.org: https://en.wikipedia.org/wiki/Pocket\_Planes