OOP PROJEKT 2

Autorid: Marten Tiisler ja Kristo Kontse

Projekti põhjalik kirjeldus, kus on kirjas programmi eesmärk ja selgitus

Projekt: Viktoriinimäng

Meie projektiks sai interaktiivne viktoriin, kus kasutaja saab valida erinevate raskustasemete vahel (lihtne, keskmine, raske) ja samuti koostada oma küsimustega viktoriini, mida ta saab mängida. Projekti eesmärk on pakkuda kasutajale viktoriini elamust programmi võtmes ning jagada erinevaid teadmisi või fakte.

Programmi üldisest tööst, vajadusel lühike kasutusjuhis;

Esmalt kuvatakse kasutajale peaekraan, kus saab valida kolme viktoriinitaseme vahel või koostada enda viktoriin ja seda mängida. Kui valitakse mõni meietehtud raskusaste, siis avaneb uus aken, kus kuvatakse küsimus ja valikvastused koos numbritega. Kasutaja saab sisestada valikvastuse numbri ja esitada oma vastuse nupuvajutusega. Seejärel annab programm küsimuse kohta tagasisidet. Kui kõigile küsimustele on vastatud, kuvab programm lõppskoori ja mängijal on võimalik teemast lahkuda “Lahku” nupu abil. Küsimuste aknast saab mängu jooksul samamoodi lahkuda, minnes sama nupu abil tagasi peaekraanile. Oma viktoriini disainimise aknas saab kasutaja lisada küsimuse, kirjutada sellele neli valikvarianti ning märkida väikse ümmarguse nupuga, et milline neist on õige. Kui küsimuseväljad on täidetud, siis peab vajutama nuppu “Lisa küsimus”, et saaks lisada järgmist küsimust. Lisatavate küsimuste arvul pole piiri, kuid kindlasti peab kõige lõpus vajutama nupule “Salvesta”, et küsimused saaksid faili kirjutatud.

Iga klassi kohta eraldi selle eesmärk ja olulisemad meetodid

Klass: Peaekraan - peaekraani aken, mis käivitub esimesena. Selle klassi eesmärk on kuvada kasutajaliides, kus saab valida erinevate raskusastmete vahel või koostada enda viktoriin.

Meetod **start**, millega kuvatakse kasutajaliides ja selle funktsioonid ning disain.

Klass: Küsimus - viktoriiniküsimuste isendiklass, kus on kolm isendivälja (küsimus, valikud, vastuse indeks). Sellest klassist saab get meetoditega (**getTekst,**

getValikud, getÕige) kätte küsimuse, vastusevariandid ja õige vastuse. Samuti on klassis konstruktor koos isendiväljadega.

Klass: Mäng - loob uue viktoriini ja kasutajaliidese akna, kus saab mängida olemasolevaid viktoriine või kasutaja loodud viktoriini. Klassi eesmärk on luua ja hallata kasutajaliidest.

Meetod **loo**, kus kasutaja saab luua ise viktoriini küsimusi. Iga küsimuse puhul on neli vastusevarianti, küsimus ning 1 õige vastus. Küsimused salvestatakse faili muudetav.txt.

Meetod **alusta** (meetod start lihtsalt eesti keeles), avab mängu akna, kus loetakse küsimusi failist.

Meetod **loeKüsimused**, mis loeb küsimused failist listi ja salvestab need Küsimus isenditena listi küsimused.

Meetod **kuvaKüsimus**, mis kuvab küsimused kasutajaliideses ja kontrollib kasutaja sisendit. Samuti lisab uue lõimega viivituse küsimuste vahele. Lõpuks kuvab meetod punktiskoori.

Projekti tegemise protsessi kirjeldus (erinevad etapid ja rühmaliikmete osalemine neis)

Esialgu leppisime kokku projekti teema ning siis hakkasime mõtlema tööjaotuse ja tööjärjekorra peale.

Tööde jaotus ja järjekord:

1. Kristo tegi valmis programmi aluse ehk siis algelise viktoriiniprogrammi koos selleks vajaminevate klasside ja funktsioonidega, mis oskab failist küsimusi kuvada ning anda tagasisidet kasutaja vastuse peale.
2. Marten disainis programmile välimuse ehk siis ühtne stiil, värvid ja disain kõikidel avanevatel akendel. Lisaks valmistas viktoriini loomise funktsionaalsuse ning mõningad väiksemad parandused või lisad (Nt lahkumise nupp).
3. Lõpus vaatasime koos veel üle, kas kõik sobib ja kas miski vajab veel täiendamist.

Nagu tööjaotusest näha on, siis meil ei olnud mingisugust suuremat edasi-tagasi disainimist, vaid saime töö jaotatud niimoodi, et mõlemad saavad ühe korraga oma töö ära teha. Me leiame, et koostöö sujus hästi ning mõlemad panustasid nii idee väljamõtlemisele kui ka programmi kirjutamisele võrdselt.

Iga rühmaliikme panus (sh tehtud klassid/meetodid) ja ajakulu (orienteeruvalt);

Nagu eelnevas küsimuses sai selgitatud, siis otseselt ei saa enamike klasside ega meetodite puhul öelda, et kumbki paariline tegi selle täiesti üksinda valmis. Ainukesteks erandideks on meetod “loo()” (Marten) ja klass “Küsimus” (Kristo). Igal pool mujal on loogika kirjutatud Kristo poolt ning disain või üksikud lisanupud/parandused lisatud hiljem Marteni poolt. Kokku läks mõlemal eraldi umbes 5-6 tundi.

Tegemise mured (nt millistest teadmistest/oskustest tundsite projekti tegemisel puudust);

Alguses oli korraks segadusseajav, et kuidas saaksime ühendada programmi koos kasutajaliidesega, sest programme ja projekte on ju selle aasta jooksul olnud mitmeid, kuid alati on kasutajaliides olnud ehitatud konsooli väljundisse, mitte tehtud eraldi kasutajaliidesena nagu javaFX. Disaini aspektist vajas samuti JavaFX’iga töötamine harjumist, sest juba objektide paigutamine ekraanile ning nende asukoha valimine on väga raske, kui võrrelda seda näiteks tavalise veebidisainiga. Kui aga sellega ära harjuda ning netist erinevate meetodite ja trikkide kohta uurida, siis juba üsna kiiresti hakkab leidma nippe ning viise, kuidas kasutada varasemaid veebidisaini oskusi ära, et luua sarnane asi ka Javas. Samuti oli segane lõimede kasutamine, kuid interneti abiga sai programmi tööle.

Hinnang oma töö lõpptulemusele (millega saite hästi hakkama ja mis vajab arendamist);

Oleme uhked oma programmi üle, kuna kasutajaliides sai kaunis ja programm töötab (hetkel) veatult. Kindlasti saaks lisada programmile funktsionaalsust (nt: nuppudega küsimused, tulemuste edetabel (statistika salvestamine), heli ja animatsioonid). Üldjoontes jäime siiski rahule, kuna lõpp-produkt vastab ülesande nõuetele ja erineb teistest viktoriini mängudest.

Selgitus ja/või näited, kuidas programmi osi eraldi ja programmi tervikuna testisite ehk kuidas veendusite, et programm töötab korrektselt.

- Katsetasime, kas programm teab alati õigesti, et milline vastus on õige.
- Katsetasime, kas programm läheb kuhugi vale asja sisestades katki.

- Kui kasutajaliides sai valmis, siis katsetasime, et mis juhtub, kui akent teha suuremaks või väiksemaks.
- Vaatasime, kas nupud töötavad õigesti.
- Kas viskab erandeid.
- Kas lisatud lõimed töötavad õigesti ja vahetuvad korrektselt.