Rühmatöö nr 2

Egert Vinogradov, Eva Meinson, Maris Palgi

**Kirjeldus**

Programm on jätk 1. rühmatöö raames loodud programmile. Lisasime oma programmile graafilise kasutajaliidese ning mõned funktsionaalsused. Loodud programm käsitleb mälumängu mängimist, kus mälumängu küsimusi võetakse veebist kasutades APIt. Kasutajale tutvustatakse programmi alguses, millega tegu ning küsitakse mõningane sisendinfo. Kasutaja saab valida küsimuste raskusastet (lisandunud funktsionaalsus) ning mitu küsimust ta vastata soovib. Kasutaja vastab küsimustele valides õige vastusevariandi. Küsimusele valesti vastates kuvab programm ka õige vastuse. Lisaks on lisatud programmile “Feeling lucky” funktsionaalsus, kus iga valiku juures (va vastuste) on nupp nimega “Feeling lucky” - sellele vajutades võtab mäng suvalise valiku ning läheb edasi.

**Klassid, nende eesmärgid ja meetodid**

Programmis on kokku 5 erinevat klassi.

* **Main –** Peaklass, mida on võimalik kasutada mängu käivitamiseks ilma graafilise kasutajaliideseta.
* **Question –** Klass küsimuste objektide loomiseks. Question klassil on isendiväljad:
  + question – String tüüpi isendiväli küsimuse hoidmiseks
  + answers – List<String> tüüpi isendiväli vastusevariantide hoidmiseks
  + correctAnswer – String tüüpi isendiväli õige vastuse hoidmiseks

Lisaks on Question klassis konstruktor ning **olulisemad meetodid**:

* getQuestion() – küsimuse kättesaamiseks
* getAnswers() – kõikide vastusevariantide kättesaamiseks
* getCorrectAnswer() – õige vastuse kättesaamiseks
* **User –** Klass mängija objekti loomiseks. User klassil on isendiväljad:
  + score – int tüüpi isendiväli kasutaja mänguskoori meeles hoidmiseks
  + name – String tüüpi isendiväli kasutaja nime jaoks

Lisaks on User klassis konstruktor ning **olulisemad meetodid:**

* getName() – mängija nime kättesaamiseks
* getScore() – mängija skoori kättesaamiseks
* addScore() – mängija skoorile punktide juurde lisamiseks
* **Quiz –** Klass mälumängu objekti loomiseks. Quiz klassil on isendiväljad:
  + questions – List<Question> tüüpi isendiväli küsimuste objektide hoidmiseks
  + user – User tüüpi isendiväli mängija objekti hoidmiseks
  + scan – Scanner tüüpi isendiväli kasutaja sisendi lugemiseks
  + numOfQuestions – int tüüpi isendiväli küsimuste arvu jaoks, mis kasutaja soovib vastata.
  + fileOrWeb – int tüüpi isendiväli küsimuste võtmise asukoha jaoks.

Lisaks on Quiz klassis konstruktor ning **olulisemad meetodid:**

* welcome() – Meetod mälumängu kohta väikse sissejuhatuse tegemiseks ning kasutaja nime ja küsimuste asukoha küsimiseks
* startQuizURL() – Meetod mälumängu alustamiseks, kui küsimused tulevad URList
* startQuizFile() – Meetod mälumängu alustamiseks, kui küsimused tulevad failist. Failis on kokku 20 küsimust, ning mängu jaoks peab valima minimaalselt 3. Kui kasutaja sisestab ebasobiliku küsimuste arvu, genereerib mäng ise 3-st 20ni ühe numbri ning alustab mängu selle arvu küsimustega.
* endQuiz() – Meetod mälumängu lõpetamise jaoks
* readQuestionsFile() – Meetod, et failist küsimused sisse lugeda
* readQuestionsURL() – Meetod, et URList küsimused sisse lugeda
* presentQuestions() – Meetod küsimuste kuvamiseks kasutajale ning vastuse lugemiseks
* presentAnswers() – Meetod vastuste kuvamiseks kasutajale
* getFileOrWeb() – Meetod kasutaja eelistuse kättesaamiseks
* setLevelOfDifficulty() – Meetod raskusastme määramiseks
* setNumOfQuestions() – Meetod küsimuste arvu määramiseks
* getNumOfQuestions() – Meetod küsimuste arvu kättesaamiseks
* getLevelOfDifficulty() – Meetod raskusastme kättesaamiseks
* getQuestions() – Meetod küsimuste listi kättesaamiseks
* **QuizGame –** Klass graafilise kasutajaliidese jaoks. Kogu mängu graafiline pool ning sellega seonduv loogika on siin klassis.
  + splitSentence() – meetod pikkade küsimuste poolitamiseks.
  + showAlert() – meetod erinevate teadete näitamiseks.
  + setEnterKeyPressAction() – meetod, kui vajutatakse nupu asemel ENTERit, siis töötaks see nagu nupule vajutamine

**Projekti tegemise protsessi kirjeldus**

2. rühmatööks otsustasime täiendada oma olemasolevat tööd lisades sellele graafilise kasutajaliidese. Kuna praksides seda ei käsitlenud, siis pidime kõik selle endale iseseisvalt selgeks tegema, mis võttis omajagu aega. Seejärel panime paika üldise graafilise liidese väljanägemise ning hakkasime korda mööda programmeerima. Töö käigus arutasime

Ajakulu meeskonnaliikmete kaupa:

* Egert – 12h
* Eva – 8h
* Maris – 8h

**Tegemise mured**

Gradle käimasaamine ning selle projekti implemeteerimine võttis omajagu aega. Lisaks läks palju aega viimistlemisele ning ka erinevate nuppude uurimisele, et mida mis nupuga teha saab ja kuidas kasutada.

**Hinnang töö lõpptulemusele**

Üldjoontes mäng töötab ja näeb ka üsna ok välja ning sellega oleme rahul. Kood võiks olla ilusam ning tõenäoliselt saaks erinevad funktsionaalsust jagada ka rohkem meetodite vahel ära, et kood loetavam oleks.

**Testimine**

Testisime oma programmi jooksvalt kohe kui midagi valmis sai ning samuti lõpus erinevate küsimuste arvude ja raskusastmetega. Testimise käigus tegime ka muudatusi nii funktsionaalsuse kui ka välimuse osas.