Rühmatöö nr 1

Egert Vinogradov, Eva Meinson, Maris Palgi

**Kirjeldus**

Loodud programm käsitleb mälumängu mängimist, kus mälumängu küsimusi on võimalik võtta kas etteantud failist või kasutades API’t. Kasutajale tutvustatakse programmi alguses, millega tegu ning küsitakse mõningane sisendinfo. Kasutaja saab valida, kas ta soovib küsimusi failist või veebist ning mitu küsimust ta vastata soovib. Kasutaja vastab küsimustele valides õige vastusevariandi. Küsimusele valesti vastates kuvab programm ka õige vastuse. Mängu lõppedes küsitakse, kas kasutaja soovib uut mängu alustada.

**Klassid, nende eesmärgid ja meetodid**

Programmis on kokku 4 erinevat klassi.

* **Main –** Peaklass, kus toimub kogu mängu käivitamine ning mängu tegevus.
* **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

**Projekti tegemise protsessi kirjeldus**

Projekti tegemise protsessi saame mõtteliselt jaotada osadeks:

* Projekti idee genereerimine – Ideid genereerisime kõik koos ning otsustasime lõpuks mälumängu kasuks, kuna tundus selline, millest võiks ka endal kasu olla. Just veebist küsimuste sisselugemisel saab kasutada erinevaid URLe ning selle kaudu ka oma teadmisi täiendada.
* üldise struktuuri paikapanemine – Arutasime üldise struktuuri ning programmi käitumise üle kõik koos koosoleku vormis
* GitHubi ülesseadmine – peamiselt tegi seda Egert koostöös Evaga, kes aitas testida
* klasside ja meetodite programmeerimine – Kõik andsid oma panuse

Ajakulu meeskonnaliikmete kaupa:

* Egert – 8h
* Eva –
* Maris –

**Tegemise mured**

Kuna varem polnud üldse GitHubiga kokku puutunud, siis selle ülesseadmine ja kasutamine võttis aega.

**Hinnang töö lõpptulemusele**

Üldjoontes mäng töötab ning sellega oleme rahul. Kood võiks olla ilusam ning tagantjärele vaadates võiks teha Quiz klassi kaheks erinevaks klassiks, üks faili ja teine veebi jaoks ning nendele ülemklassi. Tõenäoliselt peaks programmi töö jaotama ka rohkemate meetodite vahele, sest mõni meetod meie programmis teeb korraga liiga palju tegevusi, mis omakorda teeb programmi muutmise juba keerulisemaks.

**Testimine**

Testisime programmi tööd ka poole peal ning selle jaoks tegime ka vahepeal ajutisi get meetodeid, et mingid isendid üksikult kätte saada, nägemaks, millisel kujul need täpselt on, kui midagi ei töötanud. Lõpuks testisime programmi erinevate kasutaja sisenditega ning testimise käigus sai programmi ka parandatud.