

(Rühma)töö kirjeldus

Autor: Indrek Raudla

Programmi kirjeldus: Tikumäng. Programmiga saab mängida mängu, kus laual on juhuslik arv tikku. Mängijad võtavad sealt kordamööda ära kas üks, kaks või kolm tikku. Kes võtab viimase tiku, on kaotaja.

Koodis on kommenteeritud iga klassi eesmärk ning nende põhimeetodid. Teen kiire ülevaate ka siin.

Main klass – luua mängu isend ja käivitada see

Põhimeetod on muidugi main(String[] args), mis loob Gameloop isendi ning paneb selle käima

Gameloop klass – mängu töö hoida, peamenüü

Põhimeetodiks on start() mis parameetreid ei taha ning mis käivitab kogu mänguloogika. Oluline on ka meetod setup() millega luuakse soovitatav arv mängijaid ning valitakse, kas nad on inimesed või mängib loodava mängija eest arvuti.

Tikutops klass – vahendab mängijate vahel infot laual olevate tikkude kohta. On piltlikult öeldes see tikukuhi laua peal. Lihtne ja lühike klass get/set, konstruktor ja eemalda (tikke) meetodiga.

Player klass – luuakse mängijaid ning sisaldab arvuti mänguloogikat + inimmängija valikuvõimalusi

Konstruktor vajab mängija nime ning tõeväärtust, kas mängija on inimene. Kui ei ole, siis on tegu arvutiga ning hiljem kasutab arvuti mängimiseks oma loogikat.

Kõige olulisem meetod on play() mis argumenti ei vaja. Selle tööeesmärk on ära teha mängija käik - arvuti puhul arvutada, inimese puhul sisestada soovitatav eemaldatavate tikkude arv ning see tagastada, et see laual olevatest tikkudest eemaldada.

Projekti tegemise protsessi kirjeldus

Esimene etapp toimus kiirelt paberi ja pliiatsiga. Märkisin üles asjad mida ma arvasin, et mul võiks vaja minna, kuidas mänguloogika võiks töötada ning mida mul selleks teha on vaja.

Teine etapp hõlmas endast esialgse koodi kirjutamist. Tegin vajalikud klassid, kuid mitte gameloop klassi. Kogu mänguloogika käis peameetodis. Hiljem otsustasin teha eraldi klassi, mis hõlmaks kogu mängukäiku ise – nii jääb peameetod puhtamaks.

Kolmas etapp oli kõik tööle saada nii nagu mängureeglid tegelikkuses lubavad. Enne seda sammu eemaldas arvuti näiteks alati 5 tikku, mängija sai küll sisestada arvu, aga alati eemaldati samamoodi 5 tikku. Lisasin piirangud erinevatele sisenditele ning tegin arvutile lihtsa loogikaahela tikkude eemaldamiseks, mis sisaldab random() kasutamist.

Viimase sammuna viimistlesin teksti millega kasutajaga suheldakse, et see näeks viisakas välja ning edastaks vajamineva informatsiooni. Kommenteerisin olulisema koodi kõigis klassides.

Tegemise mured.

Kuna tegu on suhteliselt lihtsa projektiga (kõik väljapakutud teemad on suhteliselt sama lihtsad), siis murekohti väga ei olnud. Kõik mured mis tekkisid olid pigem enda hooletusest või lihtsalt ei mäletanud midagi, kuid materjalidest leidsin kõik vajamineva. Oleksin tegelikult tahtnud teha ilusama kasutajaliidesega mitte tekstipõhise, kuid kuna projekti nõuetes oli rangelt soovituslik teha tekstipõhine, siis nii see seekord jäi.

Hinnang tööle.

Üleüldises pildis olen projekti tulemusega suhteliselt rahul. Paar kohta koodis on sellist, mis vajaks natukene „mudimist“ et olla efektiivsemad ja viisakama väljanägemisega. Lisaks on seda arvutit suhteliselt lihtne võita, kui sa tead kuidas ta töötab. Samas ma ei näe mõtet seda arvuti loogikat kuidas ta mängib teha ideaalseks, kuna siis tekib olukord, kus mängija ei võidaks kunagi. Ma arvan, et hetkel on arvuti täpseltpiisavalt tark, et mängijat karistada lihtsate vigade eest, kuid piisavalt rumal, et mitte alati mängijat „mati“ olukorda mängida.

Mis vajab arendust on minu enda java keele oskus. Ma pean õppima koodi paremini struktureerima ja meetodeid paremini ehitama ning kasutama. Hetkel oli vahepeal endal ka tunne, et „siin võiks ju meetod olla, miks ma kogu seda koodi siia kirjutan“. Kuid eks tehes õpib ning see projekt oli tore väike õppetund.

Kuidas programmi testiti?

Tähtsamates kohtades kasutasin ka breakpointe, tihtipeale kasutasin print lauseid. Aga põhiliselt ma tegin mingi osa ära, näiteks arvuti mänguloogika, ning seejärel panin kaks arvutit omavahel erinevate tikupaki suurustega mängima niikaua kuni ma olin veendunud, et nad mängivad õigesti ja kuni ma ise olin rahul tulemustega.

Samamoodi algaasis lasin näiteks luua korraga sadakond mängijat ja kontrollisin kas mängijad ikka lisati nimekirja. Või näiteks menüüs ringi liikumist tegin manuaalselt läbi korduvalt ja korduvalt, kuni olin kindel, et midagi ei lähe valesti.

Üks asi mida ma testisin ka alguses, oli see, kuidas kontrollida kasutaja sisendit ja vajaduse korral uuesti ja uuesti küsida sisendit, kuni saadakse sobiv sisend. Kui see oli selge, siis ülejäänu oli juba lihtne. Tagantjärele mõeldes, oleks saanud kogu selle loogika oma meetodisse panna ja mitte risustada gameloop.start() meetodit, aga eks järgmises projektis saan paremini teha.

Põhimõtteliselt testisin igat osa mis ma lisasin ükshaaval justkui jõumeetodi abil. Surusin teadud väärtusi mis ei tohiks sattuda sinna ja vaatasin, kuidas programm reageerib. Mänguloogikat kontrollida oli kõige lihtsam lihtsalt lasta kahel botil mängida ja vaadata mis toimub. Või kahel mängijal üleüldiselt.