

Испит траје 3 сата.

Задатак	1	2	3	4	5	6	7	8	9	10	1	Укупно
Максималан број поена	5	5	5	5	5	5	5	5	5	5	50	100

ТЕОРИЈСКИ ДЕО

...

ПРАКТИЧНИ ДЕО

1. Написати апликацију коришћењем програмског језика C++ и библиотеке Qt која симулира борбе робота. Слика 1 илуструје како програм треба да изгледа приликом покретања. У наставку следи опис рада програма.

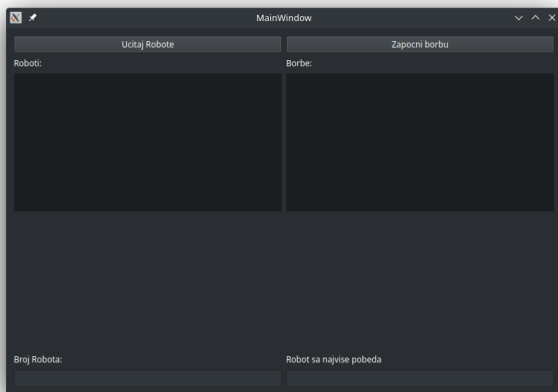
- (а) Графички кориснички интерфејс имплементирати тако да изгледа као интерфејс приказан на слици 1. Елементе прозора распоредити тако да се приликом промене његове величине, компоненте аутоматски померају да задрже приказани распоред. Поље **Роботи** је типа `QListWidget`, а поље **Борбе** је типа `QTableWidget`. Пољима “Број Робота” и “Робот са највише победа” онемогућити кориснички унос.
- (б) Сваког робота представити као објекат класе **Robot**. Објекти ове класе се састоје од наредних атрибута: (1) име робота, (2) његова минимална снага напада као неозначен цео број, (3) његова максимална снага напада као неозначен цео, (4) његови максимални животни поени као неозначен цео број и (5) тренутни животни поени као неозначен цео број. Обезбедити наредни јавни интерфејс ове класе:
 - Метод `fromQVariant` десеријализује податке о једном роботу из прослеђеног аргумента (`QVariant`). Аргумент садржи податке о свим атрибутима класе осим о тренутним животним бодовима који се постављају на максималне.
 - Метод `toQString` израчунава текст са подацима о роботу у формату који је приказан на слици 2.
 - Метод `attack` који извршава напад. Као аргумент се прослеђује робот који је нападнут и враћа да ли је нападнути робот убијен. Бодови који се одузимају нападнутом роботу се рачунају тако што се изабере псеудонасумични број из распона нападачеве снаге напада. Пошто оштећени робот удара слабије, та вредност се умањује сразмерно проценту животних бодова који недостају нападачу ограничено на 50%. Бодове одузети од тренутног броја животних бодова нападнутог робота. Робот је мртав ако му је број животних поена 0.

Дозвољено је додавање произвољних потребних поља и метода.

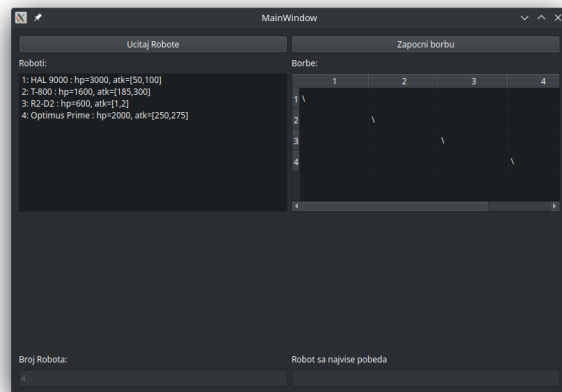
- (в) Кликом на дугме “Учитај роботе”:
 - Учитати из текстуалне датотеке податке о роботима који учествују у борбама. Текстуална датотека је записана у JSON формату (видети пример датотека `robots.json` у поставкама; број робота се може разликовати). Податке учитати као објекте класе **Robot**.
 - Приказати `robote` у `QListWidget` контроли “Роботи” као на слици 2.
 - У поље “Број робота” уписати информацију о броју робота који учествују у борбама.

Корисник може више пута учитавати податке пре започињања борби.

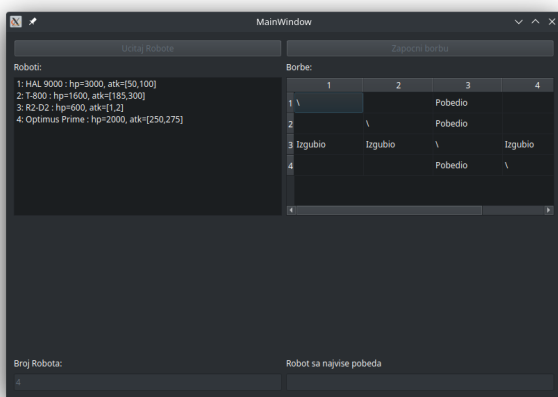
- (г) Борбе се започињу кликом на дугме “Започни борбе”. Уколико је листа робота празна, не радити ништа. У супротном онемогућити кликање на дугмад “Учитај Роботе” и “Започни борбе”. Испразнити поље “Робот са највише победа”. `QTableWidget` контролу “Борбе” иницијализовати као на слици 2 где свака колона и свака врста представља редни број робота као у `QListWidget` контроли “Роботи”. Како би апликација конкурентно симулирала све борбе за сваку борбу покренути по једну нит. Робот се не може борити сам против себе. Редослед робота није битан тј. ако је покренута нит за борбу А-Б, не покреће се за борбу Б-А.
- (д) Након што је покренута, нит животне бодове оба робота поставља на њихов максимум. Наредни кораци се понављају докле год један од робота не буде побеђен. Нит спава 0.25 секунди. Псеудонасумично се бира нападач. Оба робота имају подједнаке шансе да буду нападач. Нападач напада другог робота позивом метода `attack`. Након завршене борбе, обавестити главну нит о резултату.
Један робот не може у исто време да се бори у више борби. Све нити морају да конкурентно оперишу над истим подацима.
- (ђ) Сваки пут када главна нит добије информацију о завршеној борби, она треба да ажурира податке у `QTableWidget` контроли “Борбе” као на слици 3, где се приказује да ли је робот из врсте *i* табеле победио робота из колоне *j* табеле. Када се све борбе заврше, дугмад “Учитај Роботе” и “Започни Борбе” се поново омогуће, у пољу “Робот са највише победа” се уписује име робота који је остварио највише победа, ако више робота дели прво место уписати произвољног и главни прозор изгледа као на слици 4.
- (е) Осигурати се да не долази до проблема у конкурентном окружењу и водити рачуна о раду са динамичким ресурсима.



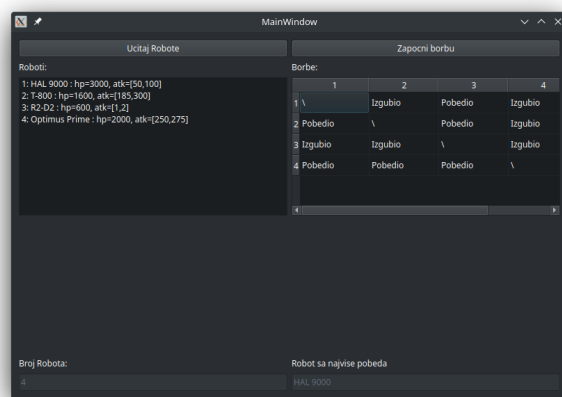
Слика 1: Приказ графичког интерфејса.



Слика 2: Приказ након учитавања робота.



Слика 3: Приказ током борби. Апликација приказује тренутно стање борби.



Слика 4: Приказ након што се заврше све борбе. Апликација приказује резултате и робота са највише победа.