Електротехнички факултет Универзитета у Београду
Катедра за рачунарску технику и информатику

Развој библиотеке за извођење лекција у виртуелном свету

дипломски рад

Ментор

др Игор Тартаља

Студент

Милош Ђоковић

Београд, 2011.

Апстракт

Овај рад описује библиотеку *Спарк* (енг. *Spark – Simple Physics And Rendering Kit*), која представља софтверско језгро за развој апликација за едукацију у окружењу виртуелне стварности и апликацију *Моја трпезарија*, развијену уз помоћ ове библиотеке.

Библиотека *Спарк* је развијена спајањем више софтверских модула нижег нивоа, од којих су најзначајнији *Ogre* (библиотека за приказ 3D графике) и *Bullet* (библиотека за симулацију закона физике, односно закона механике), те њиховом надоградњом новим концептима. *Спарк* имплементира и решења специфична за проблем дефинисања и извршавања лекција у виртуелном окружењу, као и обуку корисника кроз те лекције.

Апликација *Моја трпезарија* представља пример примене библиотеке *Спарк* и састоји се из корисничког интерфејса за коришћење библиотеке и лекција чији задаци се извршавају у виртуелној кухињи и трпезарији. Апликација *Моја трпезарија* нуди могућност избора корисничког профила под којим ће се извршавати лекција, као и избор лекције и режима у којем ће се она извршавати. Ова апликација представља и део система апликација осмишљеног да истражи могућност коришћења рачунара у обуци деце ометене у развоју. Спецификација овог система је дата у прилогу.

Садржај

1. Увод

Развој рачунарске графике пружио је могућност креирања тродимензионалног виртуелног света у рачунарским апликацијама, чиме је отворен простор за креирање широког спектра рачунарских симулација, од рачунарских игара, до софтвера за обуку астронаута. Виртуелни свет ствара код корисника осећај присуства симулираној активности, чиме се он инспирише да јој посвети већу пажњу. Код образовног софтвера, ово својство се може искористити како би корисник стекао боље разумевање симулираних активности, односно како би боље упамтио градиво. Осећај присуства у виртуелном свету се код корисника може у великој мери појачати увођењем закона физике у виртуелни свет.

Деца ометена у развоју често имају проблема са учењем врло једноставних задатака, па је потребно да велики број пута понове задатак како би га упамтили. Ово често захтева ангажовање инструктора који ће обучавати децу и надгледати их док обављају задатке како не би дошло до њиховог повређивања. Овакав начин обуке захтева већи број инструктора, односно мање групе које ће инструктор обучавати. У случајевима када није доступан довољан број инструктора, долази до неефикасности у обуци, јер одређени број деце мора да чека да инструктор постане расположив да ради са њима.

Овај рад описује апликацију развијену са циљем испитивања могућности примене виртуелне стварности у обуци деце ометене у развоју. Претпоставка је да ће развијена апликација унапредити обуку тако што ће ученици моћи да уче и увежбавају извршавање задатака у безбедном окружењу виртуелне стварности чиме би се припремили за извођење задатака у стварном свету. За време обуке на рачунару један инструктор може надгледати знатно већу групу него за време практичне обуке, док би време потребно за практичну обуку било значајно скраћено, захваљујући предзнању ученика стеченом при извођењу симулације. Централни део развијене апликације је језгро за извођење лекција у окружењу виртуелне стварности, над којим је креирана апликација *Моја трпезарија* за извођење задатака у кухињи и трпезарији.

У развоју језгра апликације су коришћена два велика софтверска система отвореног кода (енг. *open-source*), *Ogre* (*Object-oriented Graphics Rendering Engine*) и *Bullet*. *Ogre* је коришћен за приказ графичког окружења, док је *Bullet* коришћен за симулацију физичких закона у виртелном свету. Поред ових система, коришћено је и неколико мањих библиотека, чија ће употреба бити описана касније у тексту.

Резултат овог рада је софтверско језгро за извођење лекција у виртуелном свету, са имплементираним основним функционалностима за учитавање и извођење лекција и великим потенцијалом за даље проширивање и усавршавање. Језгро је конципирано као библиотека класа.

У остатку рада ће детаљније бити описане функционалности, њихова имплементација и архитектура језгра. Друго поглавље детаљно описује решавани проблем. Треће поглавље описује функционалности развијеног софтвера из перспективе корисника. У четвртом поглављу су описани софтверски системи и библиотеке коришћене за развој апликације, док пето и шесто поглавље описују архитектуру, односно поједине детаље имплементације које је аутор сматрао занимљивим.

2. Проблем

Задатак овог рада је креирање језгра за развој алата који би требало да унапреди обуку деце ометене у развоју, као и провера концепата таквог језгра кроз развој конкретне апликације. Циљ апликације је да омогући овој деци да део своје обуке обаве на рачунару, у безбедном окружењу виртуелне стварности. Овакав начин рада треба да доведе до растерећења инструктора и отварања простора за подизање квалитета обуке.

Циљна апликација треба да чини део програмског система *Моја трпезарија*, чија детаљна спецификација је дата на пратећем диску. Намена ове апликације је да омогући приказ и извођење лекција дефинисаних у формату датом у прилогу.

Уводни екрани апликације треба да омогуће избор корисника који обавља лекцију, избор лекције и избор режима извођења лекције. Лекција се може изводити у режиму

- *показивања*, када апликација сама извршава задатке по унапред дефинисаном сценарију,
- *вежбања*, када корисник извршава задатке са циљем увежбавања, те се резултати не оцењују,
- *тестирања*, када се оцењују резултати извођења задатака.

Виртуелни простор у којем се изводе лекције треба да пружи могућност кретања кроз просторије дефинисане лекцијом. Ове просторије треба да садрже објекте са којима корисник може да интерагује и да их користи за обављање лекције. Корисник тако може отварати врата и фиоке, преносити објекте са једног места на друго, укључивати и искључивати електричне уређаје, пуштати воду из чесме, или палити и гасити светло у просторији. Неке од ових акција мењају начин приказа објеката (као што је пуштање воде из чесме на судопери).

Потребно је и да апликација упозори корисника уколико га његово кретање кроз виртуелни простор може довести у опасност или направити штету. Пример оваквог случаја је приближавање укљученом шпорету или удаљавање од отворене чесме.

Апликација треба да омогући постављање задатка кориснику приказом текста и слике упутства, као и давањем гласовних инструкција. Задатак који се поставља кориснику може бити, на пример, постављање једног или више места за столом. Када апликација детектује успешно обављен задатак прелази на следећи или завршава лекцију, ако она не садржи више необављених задатака.

Уколико је лекција изведена у режиму тестирања, апликација треба да сними податке о резултатима извођења задатака. Инструктор касније може анализирати ове податке како би стекао слику о напредовању ученика и прилагодио свој рад њиховим потребама.

3. Функционална спецификација

У овом поглављу ће бити дат преглед функционалности развијеног софтвера. Прво ће бити представљене функционалности које пружају уводни екрани и односе се на припрему за извођење лекције, а затим ће бити представљене функционалности доступне кориснику у току извођења лекције.

3.1. Избор корисника

По покретању апликације, приказује се листа регистрованих корисника са њиховим сликама и именима. Кликом на поље са сликом и називом корисника се бира корисник који ће извести изабрану лекцију.

У горњем десном углу овог екрана се приказује дугме са сликом врата које служи за излаз из апликације. Кликом на ово дугме, кориснику се поставља питање да ли жели да изађе из апликације. Потврдним одговором се затвара апликација, док се одричним одговором апликација враћа на приказ екрана за избор корисника.

Апликација неће приказати екран за избор корисника уколико је у конфигурационом фајлу дефинисан подразумевани корисник.

3.2. Избор лекције

По избору корисника који ће извести лекцију, апликација приказује екран за избор лекције. На овом екрану се приказује листа лекција доступних апликацији. Кликом на поље са сликом и називом лекције се бира лекција коју ће корисник изводити.

На екрану за избор лекције у горњем десном углу је такође доступно дугме са сликом врата које служи за излаз из апликације. Лево од дугмета за излаз из лекције се налази дугме за повратак на претходни екран, у овом случају екран за избор корисника.

Уколико је у корисничком профилу изабраног корисника изабрана подразумевана лекција, апликација неће приказати екран за избор лекције, већ ће аутоматски изабрати лекцију и прећи на следећи екран.

3.3. Избор режима извођења лекције

Овај екран приказује три дугмета за избор режима извођења лекције. Доступни режими су учење, вежбање и тестирање. На екрану су, као и на екрану за избор лекције, доступна дугмад за повратак на претходни екран (екран за избор лекције) и излаз из програма.

Уколико корисник изабере режим учења, апликација сама изводи задатке по унапред дефинисаном сценарију.

У режиму вежбања корисник сам изводи задатке, али се његове акције не бодују.

У режиму тестирања корисник обавља задатке и добија одређени број бодова за њихово комплетирање.

3.4. Извођење лекције у режимима учења и тестирања

У току извођења лекције, корисник контролише анимирани лик (актера) смештен у тродимензионални виртуелни свет. Преко сцене се исцртавају додатне информације, као што су број бодова (само у режиму тестирања), слика и текст упутства текућег задатка, упутство за отварање екрана са списком контрола и прекид лекције, као и садржај инвентара. Инвентар представља ограничен скуп објеката које је корисник узео са сцене и тренутно их "држи" у рукама. Курсор миша је у овом режиму фиксиран у центру екрана, па корисник поставља курсор на неки објекат усмеравајући поглед ка том објекту. Слика 1 приказује изглед корисничког екрана са две сцене, једном из кухиње и једном из трпезарије.

Слика 1 – Поглед на кухињу и трпезарију

Циљ корисника је да комплетира постављени задатак. Када корисник комплетира све постављене задатке, лекција се завршава, а бодови се смештају у одговарајући фајл (уколико је лекција изведена у режиму тестирања).

Задаци се обављају интеракцијом са сценом. Како би читалац разумео начин интеракције корисника са сценом, потребно је увести појам мете. Мета је објекат облика квадра, невидљив за корисника, који служи да ограничи део простора сцене на који корисник може кликнути. Мета може имати придружену акцију и може детектовати спуштање објекта у простор који ограничава.

Корисник интерагује са сценом активирањем акција и узимањем и спуштањем објеката на сцену. Активација акција се постиже кликом левим дугметом миша (леви клик) на мету којој је акција придружена. За активирање неке акције може бити захтевано да корисник у тренутку клика на мету "држи" одређени предмет у рукама, односно да одређени предмет буде изабран у инвентару корисника. Објекат се са сцене може узети кликом десним дугметом миша (десни клик). Објекат који је изабран у инвентару се може спустити десним кликом на неку тачку у простору сцене. Објекат који је изабран у инвентару се исцртава у пољу које се налази у доњем левом углу екрана. Уколико је ово поље празно, ни један објекат није изабран. Тренутно изабрани објекат се мења притиском тастера *Таб* или

размак на тастатури. Притиском једног од ових тастера, бира се објекат који се у инвентару налази десно од тренутно изабраног објекта. Уколико је тренутно изабран последњи објекат у инвентару, притиском тастера за промену изабраног објекта, избор се мења тако да ни један објекат није изабран. Притиском тастера за промену изабраног објекта када ни један објекат није изабран, биће изабран први објекат у инвентару.

Резултат извршења акције може бити промена видљивости објекта (пуштање воде из чесме), укључење или искључење објекта (укључење шпорета), промена положаја објекта (отварање врата и фиока), укључење светла или комбинација ових операција. Апликација помоћу одговарајућих сензора детектује извршење сваке од ових операција, као и узимање објекта и постављање објекта на мету. Сензор је објекат који мења стање када корисник изврши операцију коју дати сензор прати. Када корисник изврши скуп операција које чине задатак, апликација означава задатак као извршен и прелази на следећи.

Сваки задатак може имати једну или више секвенци сензора. Активирање секвенце сензора може успешно завршити задатак и прећи на нови (секвенца комплетирања), или може поништити задатак и прећи на неки од претходних (секвенца ресетовања). Успешно комплетирање задатка апликација сигнализира штриклирањем упутства и одговарајућим звуком, док се поништавање задатка сигнализира прецртавањем текста задатка и одговарајућим звуком.

3.5. Извођење лекције у режиму показивања

За разлику од режима учења и тестирања у којима корисник обавља задатке интеракцијом са виртуелним светом, у овом режиму апликација сама извршава задатке по унапред дефинисаном сценарију, учећи корисника како се задаци обављају.

Апликација наводи актера по задатој путањи и у одређеним тачкама извршава акције, односно узима/спушта објекте. Како би кориснику било јасно где треба да кликне, апликација пре извршења акције наизменично пали и гаси приказ мете која активира ту акцију.

Пре сваког задатка кориснику се приказује комплетно упутство и слика тог задатка, а извршавање задатка се паузира. Извршавање задатка почиње када корисник кликне на дугме *Start* испод упутства задатка.

3.6. Упозоравање корисника на опасност

Како би корисник научио који објекти могу представљати опасност, апликација пружа могућност дефинисања две врсте опасних објеката и упозоравање корисника на улазак у опасну зону или излазак из безбедне зоне.

Врсте опасних објеката које се могу дефинисати су објекти са опасном зоном и објекти са безбедном зоном. Објекти са опасном зоном су они којима је опасно прићи, као што је укључена рингла шпорета, док су објекти са безбедном зоном они од којих се не треба удаљавати, као што је отворена чесма судопере.

Апликација упозорава корисника да улази у опасну зону, или излази из безбедне зоне, тако што мења интензитет црвене компоненте светла сцене. Интензитет црвене компоненте светла је јачи када је корисник дубље у опасној зони.

4. Постојећи алгоритми и алати за развој

У данашње време, када су за већину честих рачунарских проблема доступне библиотеке отвореног кода, ретко се може наћи добро оправдање за развој апликативног софтвера "од нуле". Иако решења која подразумевају имплементацију апликативног софтвера коришћењем библиотека и API-ја нижег нивоа могу у раним фазама развоја бити примамљива због велике слободе коју има програмер и једноставности писања кода, оваква решења би у каснијим фазама захтевала велики труд за имплементацију напреднијих својстава, као што су анимација, симулација закона физике и сложено текстурисање објеката. Поред тога, овакво решење би било подложније појави програмских дефеката ("багова") и архитектуралних проблема који би спречавали даље проширивање апликације. Библиотеке отвореног кода са великим групама корисника су обично "отпорније" на овакве проблеме, јер их пројектује и тестира велики број корисника пре њиховог објављивања.

Због свега овога, основни проблем програмера постаје избор и спајање библиотека које решавају специфичне проблеме нижег нивоа и имплементација логике специфичне за конкретан апликативни софтвер.

Апликација *Моја трпезарија* није изузетак. Имајући претходне ставове у виду, аутор је изабрао да искористи неколико популарних библиотека за решавање проблема као што су цртање графике, симулација физике и генерисање звука. Неке од ових библиотека су коришћене и у вишемилионским комерцијалним пројектима (нпр. *Bullet* је коришћен за симулацију специјалних ефеката у филму 2012.), што је додатни аргумент у прилог њиховог квалитета. У наставку ће бити дати описи коришћених библиотека и разлози њиховог избора.

4.1. Ogre

Библиотека *Ogre* (*Ogre – Object-oriented Graphics Rendering Engine*) је основа библиотеке *Спарк*, а самим тим и апликације *Моја трпезарија*. Ова библиотека, као што јој и име каже, служи за приказ рачунарске графике (првенствено 3D графике). *Ogre* апстрахује системске интерфејсе за комуникацију са графичким хардвером (*OpenGL* і *Direct3D*), чиме омогућава програмеру рад на вишем нивоу апстракције.

Ogre омогућава приказ унапред дефинисаних модела и анимација, управљање сценом, текстурисање модела на основу скриптова, управљање светлом и камерама и решава друге уобичајене проблеме из области рачунарске графике као што је приказ сенки, приказ система честица (енг. *particle systems*) и управљање приказом транспарентних објеката. Поред својстава која се односе на приказ графике, *Ogre* садржи и библиотеке за рад са векторима и матрицама, као и библиотеку за рад са ресурсима, управљање меморијом и рад са ZIP и PK3 архивама.

Ogre је добро документована библиотека, са великом и врло активном заједницом корисника што је у великој мери допринело избору *Ogre*-а за развој библиотеке *Спарк*. Избор *Ogre*-а је у великој мери предодредио избор библиотека за приказ

GUI-ја и рад са звуком, па су изабране библиотеке *CEGUI* и *OgreOggSound*, које проширују *Ogre* и додају му ове функционалности.

Лоше стране избора *Ogre*-а су последица високог нивоа апстракције рада са графичким хардвером. Како *Ogre* посматра сцену као граф објеката, а не као низ слика, недостаје подршка за цртање примитива на одређеној слици. Аутор је наишао на проблем цртања линије које је потребно за имплементацију *debug* приказа физичких модела. Наиме, једини начин да се у *Ogre*-у нацрта линија је креирање објекта који садржи ту линију. Када се ово понавља за сваку слику, потроши се доста процесорског времена, јер је креирање објеката скупа операција. Аутор је у одређеној мери успео да оптимизује овај процес, али је и даље *debug* приказ остао скупа операција. Такође, *Ogre* зароби курзор у прозору апликације, те док је симулација у току, мишем се не може изабрати неки други прозор или екрански објекат изван текуће апликације.

4.2. Bullet

Bullet је библиотека отвореног кода која служи за симулацију физичких закона у виртуелном свету. У библиотеци као што је *Спарк* довољно би било детектовати колизију између објеката, па се може рећи да је укључивање *Bullet*-а нека врста непотребног луксуза, али симулација закона физике је значајно повећала интерактивност и реалистичност приказа виртуелног света, креираног коришћењем библиотеке *Спарк*.

Поређење перформанси различитих библиотека за симулацију закона физике превазилази оквире овог рада, па се аутор при избору ове библиотеке водио другим радовима и демонстрацијама доступним на Интернету. Избор је пао на *Bullet* због перформанси описаних у тестовима[3] и приказаних у демонстрацијама доступним на званичном сајту библиотеке[4], чињенице да је библиотека отвореног кода, као и чињенице да већ постоје пројекти који су успешно комбиновали *Ogre* и *Bullet*[5],[6].

За разлику од *Ogre*-а, *Bullet* није подједнако добро документован и нема тако активну заједницу корисника, али се ипак уз мало стрпљења може добити одговор на већину питања, постављених путем форума на званичном сајту. Примећени су и проблеми у симулацији објеката који имају бар једну од ивица окружујућег квадра краћу од 0,2 јединице, односно појава подрхтавања и тунел-ефекта при симулацији оваквих објеката (ова врста проблема је превазиђена у библиотеци *Спарк*).

4.3. CEGUI

Библиотека *CEGUI*[7] служи за приказ графичког корисничког интерфејса (GUI) и првенствено је намењена за приказ интерфејса рачунарских игара. Библиотека може користити *Ogre* или *Irrlicht*[8] за рендеринг, али може и директно користити *Direct3D* или *OpenGL*. За овај рад је од интереса подршка за *Ogre*.

Иако прилично комплексна, ова библиотека постоји још од 2003. године и активно се користи у заједници која користи библиотеку *Ogre*. Чак и званичне демо апликације библиотеке *Ogre* користе *CEGUI* као GUI библиотеку.

CEGUI подржава дефинисање комплетног изгледа и распореда елемената корисничког интерфејса путем података, односно слика и XML фајлова, док се само обрада догађаја мора налазити у коду апликације која користи ову библиотеку. Ово практично значи да је могуће променити изглед или распоред елемената без поновног превођења апликације која користи *CEGUI*.

Лоше стране ове библиотеке су осредња документација и помало једноличне унапред дефинисане теме.

4.4. OgreOggSound и OpenAL

OgreOggSound[9] је софтверски "омотач" библиотеке *OpenAL*[10], имплементиран тако да омогући једноставно додавање подршке за 2D/3D звук апликацијама које користе *Ogre*. *OpenAL* је платформски независан 3D аудио API. *OpenAL* моделује изворе звукова као објекте који се крећу у 3D простору, које слуша један пријемник, такође постављен у неку тачку простора.

OgreOggSound има могућност пуштања звука из ogg и wav фајлова, пуштање звукова из меморије или из фајла, везивање извора и пријемника звука у граф сцене, коришћење више програмских нити за пуштање звука и друге напредне могућности. *Спарк* користи само основне могућности ове библиотеке, за пуштање звука упутства задатка и звучног упозорења на комплетиран односно поништен задатак. Уколико постоји потреба за напредним функционалностима ове библиотеке, библиотека *Спарк* се лако може проширити да користи и те функционалности.

Недостатак ове библиотеке је такође оскудна документација, али ако се узме у обзир да је реч о релативно малој библиотеци отвореног кода, овај недостатак се може превазићи. Уколико корисник ипак наиђе на проблем који не може савладати сам, на располагању му је тема званичног *Ogre* форума, посвећена овој библиотеци.

4.5. TinyXML++

TinyXML++ или *ticpp*[11] како је још називају, је једна од мањих библиотека коришћених у реализацији библиоитеке *Спарк*, али то не умањује њен значај. Ова библиотека се користи за рад са XML фајловима, односно учитавање лекција и конфигурације, као и за снимање остварених бодова при извршавању лекције.

TinyXML++ је C++ омотач за библиотеку *TinyXML* и задржава њен скуп функционалности, али доноси C++ конструкте као што су темплејти и изузеци. Библиотека парсира XML документ у стабло чији су чворови елементи XML фајла. Ово стабло се може претраживати, односно може се приступати чворовима по називу одговарајуће ознаке или атрибута.

TinyXML++ има и бројна ограничења, међу којима су немогућност провере ваљаности XML документа према задатој шеми, недостатак подршке за стилове и просторе имена (енг. *namespaces*).

5. Пројекат софтвера

У овом поглављу ће бити описана архитектура библиоитеке *Спарк* и апликације *Моја трпезарија*. Библиотека *Спарк* је задужена за учитавање и извршавање лекција, док апликација *Моја трпезарија* имплементира GUI за избор корисника, лекције и начина извођења лекције, као и GUI за приказ статусних информација, као што су садржај инвентара и упутство за обављање лекције.

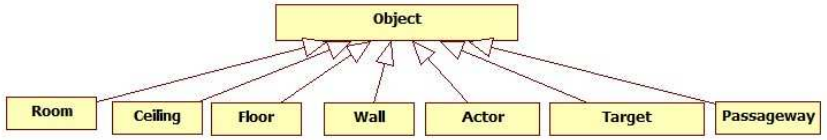
5.1. Контрола тока симулације и интерфејс ка екстерним модулима

Основна класа библиотеке *Спарк* која контролише ток симулације је класа `WorldManager`. Ова класа садржи метод `stepSimulation(long dt)` који служи за инкрементирање времена симулације. Када дође до померања симулације по временској оси, класа о томе обавештава регистроване посматраче. Ова класа је одговорна за креирање и уништавање објекта класе `Actor`, која представља анимирани лик под контролом корисника.

Класа `WorldManager` уз класе `LectureManager`, `UserManager`, `Configuration` и `WorldCreator`, представља интерфејс који могу користити екстерни модули за комуникацију са библиотеком *Спарк*. Класа `LectureManager` се користи за контролу учитавања и тока лекције. Класа `UserManager` се користи за издавање команди за учитавање података о кориснику, избор корисника који извршава лекцију и снимање података о оствареним бодовима. Класа `Configuration` служи за учитавање конфигурације језгра, која дефинише локацију фолдера са лекцијама, локацију фолдера са корисничким профилима, назив подразумеваног корисничког профила (ако је постављен) и локацију фајла са бодовима. Класа `WorldCreator` се користи за иницијализацију библиотека које користи библиотека *Спарк*.

5.2. Представљање објеката, спој графичког и физичког света

Сви објекти који се приказују на сцени припадају класи `Object`, или класама које су из ње изведене (Слика 2). Класа `Object` спаја графички приказ објекта са његовим физичким моделом. Како се захтеви симулације физичких својстава тела и њиховог графичког приказа знатно разликују, разликују се и модели тела у графичком и физичком свету. У случају графичког модела, корисник очекује сложени приказ изгледа модела, па он мора што више подсећати на објекат који представља. Физички модел корисник не може да види већ о њему сазнаје на основу интеракције са сценским објектом. Са друге стране, за графички модел није потребно обавити сложене прорачуне кретања сваке тачке модела између два временска тренутка, а прорачуне приказа у великој мери може убрзати специјализовани графички хардвер. Због тога се коришћењем сложеног графичког и упрошћеног физичког модела постиже значајно побољшање перформанси, док је губитак реалистичности приказа занемарљив. Наиме, ако физички модел чаше заменимо ваљком, корисник то вероватно неће ни приметити, јер ће чаша и даље моћи да се преврне и откотрља. Насупрот томе, ако графички модел чаше заменимо простим ваљком, корисник неће знати да модел представља чашу.

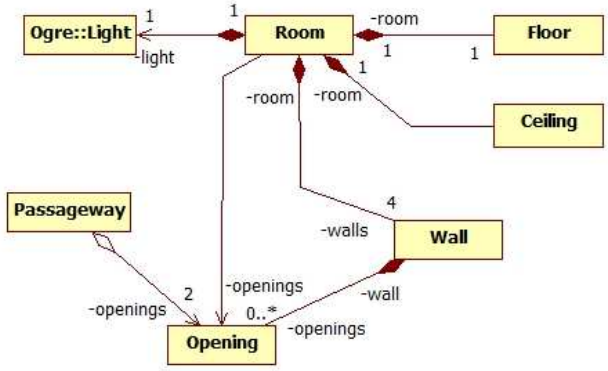


Слика 2 – Хијерархија сценских објеката

Како би симулација била реалистична, потребно је да се положаји графичког и физичког модела подударају у сваком тренутку. У библиотеци *Спарк* је за ажурирање положаја графичког модела објекта на основу положаја његовог физичког модела, искоришћен *Bullet*-ов механизам `btMotionState`. О овом механизму ће бити више речи у одељку 6.1, а овде је довољно рећи да *Bullet* на свакој слици ажурира положај графичког модела.

5.3. Представљање просторија

Лекције које приказује библиотека *Спарк* су смештене у просторије, међусобно повезане пролазима. Усвојено је да су просторије правоугаоне основе, са зидовима паралелним равнима које дефинишу осе координатног система. Такође је усвојено да сви зидови једне просторије имају исту текстуру, док под просторије може имати другу текстуру. Зидови просторије могу имати произвољан број отвора. Под просторије је смештен у *xOz* раван координатног система, док је y-оса усмерена ка плафону просторије. Оваква орјентација просторије је усвојена због начина на који *Ogre* подразумевано поставља камеру. У центру просторије, на плафону, је смештено светло просторије.



Слика 3 – Простор сцене

Библиотека *Спарк* аутоматски генерише моделе просторија на основу података задатих у фајлу лекције. Подаци потребни за генерисање просторије су величина

просторије и њен положај у односу на координатни почетак, текстуре зидова и пода и величине и положај отвора у односу на угао просторије са минималним координатама. Дијаграм класа које представљају простор сцене у којем се одвија лекција приказује Слика 3.

5.4. Својства објеката, аларми, контролери и сензори

Због генеричке природе класе `Object`, понашање објекта који се јавља у лекцији се дефинише конфигурацијом инстанце ове класе. При креирању генеричке инстанце објекта, додељују му се својства дефинисана лекцијом и придружују му се објекти одговарајућих контролера и сензора.

Класа `Controller` и њене потклассе представљају контролере који служе за вршење операција над објектом. Ове операције могу бити промена статуса укључености, промена видљивости, промена положаја и свака је дефинисана одговарајућом потклассом класе `Controller`. Међу контролерима треба издвојити контролер светла, који се не везује за генерички објекат, већ се везује за просторију и управља стањем светла просторије.

Класа `Sensor` и њене потклассе представљају сензоре који служе за детектовање промене стања објекта који прате. Промене стања објеката је потребно пратити како би апликација детектовала извршење задатка, јер су задаци дефинисани као низ промена стања објеката које корисник треба да изазове. Сензори имплементирају пројектни узорак посматрач (енг. o*bserver*). Инстанце класе `Sensor` су посматрачи, док су инстанце класа `Object`, `Actor`, `Target`, `Inventory`, `Room` субјекти. Субјекат такође може бити и други сензор, па се сензори могу организовати у граф чији су чворови сензори посматрачи других сензора. Ови "везни" сензори врше логичке операције И, ИЛИ и НИЛИ над стањима повезаних сензора и активирају се уколико је резултат логичке операције истинит. На овај начин се може представити произвољна логичка операција, чији су операнди стања сензора. Поред поменутих везних сензора који представљају логичке операције, сензори могу бити повезани и посебном врстом сензора који се активира када је активан задат број повезаних сензора.

Контролу упозоравања корисника на опасност врше објекти класе `Alarm`. Објекти ове класе се придружују сценским објектима који су означени као опасни. Када се аларм активира, мења интензитет црвене компоненте светла просторије у зависности од удаљености актера од опасног објекта.

5.5. Кретање кроз виртуелни простор, актер и инвентар

Присуство актера у виртуелном свету је симулирано класом `Actor`. Ова класа имплементира пројектни узорак уникат (енг. *singleton*). Класа `Actor` имплементира приказ и контролу анимираног лика који представља корисника, као и интеракцију са сценом. Како би кретање актер било што реалније, односно како би се симулирали судари са објектима на сцени, актеру је придружен физички модел капсуле на који корисник делује силом како би га померао.

`Actor` садржи инстанцу класе `Inventory` која представља инвентар у којем се налазе сценски објекти које је корисник узео са сцене. Сам приказ инвентара дефинише апликација која користи библиотеку *Спарк*, а библиотека *Спарк* креира текстуру са приказом објеката у инвентару. Апликација *Моја трпезарија*, између осталог, имплементира пример приказа инвентара.

5.6. Моделовање интеракције са објектима, мете и акције

Корисник са објектима интерагује узимањем и спуштањем објеката са сцене или активацијом контролера који су придружени објекту. Како су корисничке акције често сложене, а контролери извршавају само просте промене стања објеката, потребно је груписати и истовремено активирати више контролера. Ову функцију обављају инстанце класе `Action`, која представља сложену корисничку акцију, као што је укључење шпорета (промена стања у укључен, промена видљивости црвене сигналне лампице).

Посебна врста објеката су мете. Мете су објекти невидљиви за корисника који служе да ограниче простор на који се може кликнути, односно у који се може спустити неки предмет. Метама је придружен сензор локације који се активира када се на мету спусти одговарајући објекат, дефинисан лекцијом. Активирањем мете се покреће акција (описана класом `Action`) која је придружена мети, а посредством акције се активирају контролери који мењају стање објекта, што доводи до промене стања сензора.

5.7. Учитавање лекције

Опис лекције се дефинише XML фајлом чији је формат дат у прилогу. Овај фајл је спакован у .lkc архиву, заједно са другим фајловима потребним за приказ лекције, као што су графички модели, текстуре, фајлови са звучним упутством и други. Архива у коју је спакована лекција је заправо .zip фајл са екстензијом карактеристичном за библиотеку *Спарк*.

Команда за учитавање лекције се издаје инстанци класе `LectureManager` која позива класу `LectureReaderXML`. `LectureReaderXML` чита XML фајл и за сваку секцију позива одговарајућу потклассу класе `BaseLoader` која имплементира читање те секције, стварање објекта одговарајуће класе и конфигурисање тог објекта.

5.8. Контрола тока лекције

Када је учитавање завршено, `LectureManager` покреће лекцију. Лекција садржи сценарио (`Scenario`) који се састоји од задатака које корисник треба да обави. Када се покрене задатак, укључује се звучно упутство (под условом да је звучно упутство омогућено конфигурацијом), а апликација која користи библиотеку може приказати и текстуално, односно сликовно упутство.

По окончању давања упутстава, активирају се секвенце ставки задатка. Ове секвенце су посматрачи листе сензора. Како би се секвенца сматрала

комплетираном, потребно је да се сви сензори на листи активирају задатим редоследом. Секвенца ставки задатка може бити секвенца комплетирања или секвенца поништавања и служи да се дефинише редослед којим корисник треба да активира сензоре како би успешно комплетирао задатак, односно поништио неки успешно обављен задатак. У зависности од типа комплетиране секвенце, апликација одлучује који ће следећи задатак активирати и да ли ће остварени бодови бити уписани у фајл са бодовима.

5.9. Демонстрација извођења лекције

Како би се кориснику библиотеке омогућило да научи како се одређени задатак извршава, имплементиран је механизам за демонстрацију извођења задатка. Када се активира демонстрација извођења лекције, корисник нема контролу над актером, већ се контрола предаје класи `DemoSequence`. `DemoSequence` управља актером наводећи га по тачкама задатим инстанцама класе `DemoPoint`. Сваки задатак може имати дефинисану једну `DemoSequence`. Детекција комплетирања задатка се врши на исти начин као када лекцију извршава корисник.

5.10. Управљање корисницима

Управљање корисницима се обавља посредством класе `UserManager`. Ова класа, као и остале управљачке класе имплементира пројектни узорак Уникат, како би јединствена инстанца ове класе била лако доступна из свих делова апликације у којима се обавља рад са корисницима, односно корисничким профилима и како би спречила вишеструке појаве корисника у извођењу лекције.

Кориснички профил је у библиотеци *Спарк* представљен класом `User`. Објекти ове класе садрже податке специфичне за корисника који утичу на начин извођења лекције, односно начин приказа упутстава и дозвољене режиме извођења лекције.

Класе `Score` и `TotalScore` представљају бодове остварене комплетирањем задатка, односно лекције. Класа `TotalScore` имплементира упис остварених бодова у фајл са бодовима.

5.11. Звучна упутства и ефекти

Подршка за звук је имплементирана у класама библиотеке *OgreOggSound*, а креирање и пуштање звукова контролише класа `Task`. Ова класа имплементира методе `void Task::playTaskSound()` и `void Task::playSound(const String& soundName)`, које служе за репродукцију звучног упутства задатка, односно звучне поруке успешно или неуспешно обављеног задатка.

5.12. Коришћење библиотеке из екстерног модула и GUI систем

Како би екстерни модули који користе библиотеку *Спарк* исправно приказивали статусне информације, дефинисан је механизам посматрача лекције. Класе које наслеђују класу `LectureObserver` могу се регистровати да примају поруке о

догађајима насталим у току извршавања лекције. Ове поруке се шаљу када лекција почне или се заврши, односно када корисник отпочне нови задатак или га заврши.

Апликација *Моја трпезарија* дефинише класе које имплементирају GUI систем. Централна класа је `CeguiGuiSystem` која контролише иницијализацију *CEGUI* библиотеке, креирање класа које представљају екране и дијалоге и њихово повезивање. Како је ток корисничких екрана линеаран, за сваки кориснички екран се дефинише претходни и следећи екран. Сам изглед корисничких екрана је дефинисан у *CEGUI* XML фајловима, док класе које представљају корисничке екране заправо имплементирају интеракцију корисника са елементима корисничког интерфејса, као што су листе и дугмад.

6. Имплементација софтвера

Аутор је изабрао да у овом поглављу укратко представи механизам који спаја графичке и физичке моделе сценских објеката, као основни елемент библиотеке *Спарк*. Овде ће такође бити дат преглед техничких карактеристика развијеног софтвара. Под техничким карактеристикама је подразумеван број класа, број линија кода, број фајлова и велична .exe и .dll фајлова.

6.1. Спајање *Ogre* и *Bullet*, `btMotionState`

Један од основних проблема које библиотека *Спарк* решава је спајање графичког и физичког модела објекта. За ове потребе, било је неопходно наследити класу `btMotionState` и редефинисати методе `getWorldTransform(btTransform&)` и `setWorldTransform(btTransform&)`, које *Bullet* позива за дохватање, односно враћање тренутне трансформације физичког модела.

Иста функционалност је могла бити обезбеђена и другачије. Било је могуће да се пре исцртавања сваке слике прође кроз листу физичких објеката и ажурира њихов положај. Употреба btMotionState има неколико предности над оваквим решењем:

- ажурирање трансформације само оних графичких модела чији су се физички модели померили у претходној итерацији,
- могуће је извршавање произвољних операција када дође до померања тела.

```
void SparkMotionState::getWorldTransform(btTransform &worldTrans) const {
       worldTrans = getBulletTransform(mVisibleobj);
}

void SparkMotionState::setWorldTransform(const btTransform &worldTrans) {
       if(NULL == mVisibleobj) return; // silently return before we set node
       Quaternion orientation = convert(worldTrans.getRotation());
       Vector3 position = convert(worldTrans.getOrigin());
       SceneNode* parent = mVisibleobj->getParentSceneNode();
       if (parent && parent != mSceneManager->getRootSceneNode()) {
              orientation = orientation *
                     parent->_getDerivedOrientation().Inverse();
              position = position - parent->_getDerivedPosition();
       }
       mVisibleobj->setOrientation(orientation);
       mVisibleobj->setPosition(position);
}
```

Листинг 1 – Редефинисање `SparkMotionState`

Додатну потешкоћу је представљало то што је усвојено да је јединична дужина у графичком свету једнака дужини од 1 метар у стварном свету. Како *Bullet* има проблема са симулацијом објеката чија је дужина странице око 0,02 јединице, а неки објекти коришћени у лекцији могу бити и мањи (нпр. нож), било је неопходно скалирати физички свет и објекте у њему. Ово је једноставно имплементирано, јер је једина додирна тачка графичке и физичке представе објекта класа

SparkMotionState, која је изведена из `btMotionState`. Листинг 1 приказује имплементацију метода `getWorldTransform` и `setWorldTransform` у класи `SparkMotionState`.

6.2. Техничке карактеристике имплементираног софтвера

Преглед техничких карактеристика библиотеке *Спарк* и апликације *Моја трпезарија* даје Табела 1.

	Библиотека Спарк	Апликација Моја трпезарија
Број класа	87	14
Број линија кода	12140	1697
Број фајлова	121	6
Величина извршног модула (exe или dll)	860 KB	462 KB
Број линија демо лекције (постављање стола)	-	3671
Величина демо лекције са пратећим моделима	-	3,6 MB

Табела 1 - Техничке карактеристике развијеног софтвера

Поред наведених карактеристика, може се још напоменути да комплетна инсталација апликације са свим пратећим библиотекама и демо лекцијом, заузима 125 MB простора на диску.

7. Закључак

Став аутора је да је библиотека *Спарк* поставила стабилну основу за развој образовних апликација у виртуелном простору. Библиотека се може лако проширивати проширивањем скупа функционалности *Ogre*-а и *Bullet*-а које библиотека *Спарк* користи.

Иако формат лекције може испрва деловати комплексно, заправо је врло једноставан, а ширина спектра лекција које се њиме могу представити оправдава труд потребан за његово савладавање. Процес креирања лекције би био додатно олакшан развојем графичког едитора лекција који је описан у спецификацији система апликација *Моја трпезарија*, која је дата на пратећем диску.

Апликација *Моја трпезарија* је проверила ваљаност библиотеке *Спарк*. Она представља алат који треба да испита могућности вршења обуке деце ометене у развоју у виртуелном окружењу. Пре него што се овај алат употреби у пракси потребно је додатно унапредити контролу актера и механизам давања инструкција. Такође треба развити алат за анализу резултата извођења лекције, како би инструктор могао што лакше да прати учинак ученика.

Што се тиче унапређења библиотеке *Спарк* аутор сматра да се релативно једноставно може унапредити подршка за звук, додавањем звукова сцене који би се активирали акцијама корисника. Ова функционалност се може имплементирати проширивањем формата лекције и додвањем нове врсте контролера.

Највеће достигнуће развијеног софтвера није скуп функционалности који пружа, већ могућност проширивања и додавања нових функционалности.

8. Литература

[1] OGRE – Open Source 3D Graphics Engine, http://www.ogre3d.org/

[2] Bullet Physics Library, http://www.bulletphysics.org/

[3] Boeing, A, Bräunl, T, “Evaluation of real-time physics simulation systems”, http://www.adrianboeing.com/pal/papers/p281-boeing.pdf

[4] PAL demo and benchmark, http://www.adrianboeing.com/pal/benchmark.html

[5] gamekit, http://code.google.com/p/gamekit/

[6] OgreBullet, http://www.ogre3d.org/tikiwiki/OgreBullet

[7] CEGUI, http://www.cegui.org.uk/

[8] Irrlicht Engine, http://irrlicht.sourceforge.net/

[9] OgreOggSound, http://www.ogre3d.org/tikiwiki/OgreOggSound&structure=Libraries

[10] OpenAL, http://connect.creativelabs.com/openal/

[11] TinyXML++, http://code.google.com/p/ticpp/

9. Прилози

У прилозима су дати описи формата фајлова које користи библиотека *Спарк*. Нормалним фонтом је написан текст који се дословно појављује у описаним фајловима. Овде спадају називи XML тагова и атрибута. Курзивом је написан опис вредности која се јавља на датом месту. Пример за ово је путања до фолдера са профилима корисника у конфигурационом фајлу. Тагови и атрибути који су подвучени су опциони, односно могу се изоставити. Уколико се на неком месту јавља једна од вредности из неког скупа, те вредности су наведене нормалним фонтом и раздвојене знаком '|'. На местима где је аутор сматрао да је из назива тага или атрибута јасно која се вредност на том месту јавља, уместо описа вредности је стављен знак '...'. За означавање понављања претходног тага произвољан број пута, коришћен је симбол +++. Додатна објашњења су дата у XML коментарима.

Аутор се одлучио за овакав начин описа формата фајлова, уместо за неку од стандардних синтаксних нотација, јер је сматрао да је овакав опис разумљивији за неупућеног корисника који би могао пожелети да промени конфигурацију апликације, креира или измени неку од лекција, прочита број остварених бодова, или креира новог корисника, пре него што буде развијен графички алат за ту сврху.

9.1. Прилог I – Опис формата фајла конфигурације

```
<?xml version="1.0" ?>
<konfiguracija>
        <korisnici> фолдер који садржи профиле ученика </korisnici >
        <rezultati> локација фајла са резултатима тестирања </rezultati>
        <lekcije> фолдер са лекцијама </lekcije>
        <podrazumevani> ознака подразумеваног корисника </podrazumevani>
</konfiguracija>
```

9.2. Прилог II – Опис формата фајла лекције

```
<?xml version="1.0" ?>
<lekcija id="..." naziv="...">
        <scena>
                <prostorije>
                        <prostorija naziv="назив просторије">
                                <velicina>
                                        <x> апсолутна величина по x-оси </x>
                                        <y>...</y>
                                        <z>...</z>
                                </velicina>
                                <lokacija>
                                        <x>...</x>
                                        <y>...</y>
                                        <z>...</z>
                                </lokacija>
```



```
                                <materijal>
                                        <pod>
                                                <tekstura naziv="назив материјала">
                                                        <velicina>
                                                                <s> релативна величина
                                                                        по s-оси </s>
                                                                <t>...</t>
                                                        </velicina>
                                                        <polozaj>
                                                                <s> офсет по s-оси </s>
                                                                <t> офсет по t-оси </t>
                                                        </polozaj>
                                                </tekstura>
                                        </pod>
                                        <zid>
                                                <tekstura naziv="назив материјала">
                                                        <velicina>
                                                                <s> релативна величина
                                                                        по s-оси </s>
                                                                <t>...</t>
                                                        </velicina>
                                                        <polozaj>
                                                                <s> офсет по s-оси </s>
                                                                <t> офсет по t-оси </t>
                                                        </polozaj>
                                                </tekstura>
                                        </zid>
                                </materijal>

                                <otvori>
                                        <otvor naziv="...">
                                                <velicina>
                                                        <x> апсолутна величина по x-оси </x>
                                                        <y>...</y>
                                                        <z>...</z>
                                                </velicina>
                                                <lokacija>
                                                        <x>...</x>
                                                        <y>...</y>
                                                        <z>...</z>
                                                </lokacija>
                                        </otvor>
                                        +++
                                </otvori>
                        </prostorija>
                        +++

                        <prolaz>
                                <materijal>
                                        <tekstura naziv="назив материјала">
                                                <velicina>
                                                        <s> релативна величина
                                                                по s-оси </s>
                                                        <t>...</t>
                                                </velicina>
```

```
                        <polozaj>
                            <s> ofset po s-osi </s>
                            <t> ofset po t-osi </t>
                        </polozaj>
                    </tekstura>
                </materijal>
                <prostorija naziv="naziv prostorije" otvor="naziv otvora"/>
                <prostorija naziv="..." otvor="..."/>
        </prolaz>
</prostorije>

<akter>
        <domet> maksimalno rastojanje između aktera i mete ili
                predmeta, na kojem je dozvoljena interakcija </domet>
        <lokacija>
                <x>...</x>
                <y>...</y>
                <z>...</z>
        </lokacija>
        <orijentacija>
                <x> rotacija modela u stepenima oko x-ose </x>
                <y>...</y>
                <z>...</z>
        </orijentacija>
</akter>

<objekti>
        <objekat naziv="..." vrsta="...">
                <grafickiModel> naziv fajla sa modelom </grafickiModel>
                <fizickiModel>
                    <oblik>BOX | MESH | SPHERE | CYLINDER |
                        CAPSULE | CONVEX</oblik>
                    <masa> masa objekta </masa>
                </fizickiModel>
                <roditelj naziv="naziv objekta kojem je posmatrani
                    objekat prikljucen" />
                <svojstva>
                    <vidljiv> DA | NE </vidljiv>
                    <prenosiv/>
                    <ukljucenje> ako je tag naveden,
                        podrzava ukljucivanje, a vrednost definise
                        da li je inicijalno ukljucen
                        DA | NE </ukljucenje>
                </svojstva>
                <velicina> podrazumevana velicina je (1, 1, 1)
                    <x> relativna velicina po x-osi </x>
                    <y>...</y>
                    <z>...</z>
                </velicina>
                <lokacija>
                    <x>...</x>
                    <y>...</y>
                    <z>...</z>
                </lokacija>
                <orijentacija>
```



```
                    <x> rotacija modela u stepanima oko x-ose </x>
                    <y>...</y>
                    <z>...</z>
                </orijentacija>
        </objekat>
        +++
</objekti>

<kontrola>
        <senzori>
            <senzor naziv="...">
                <tip naziv="I | ILI | NILI | BROJ"> broj aktivnih
                    senzora potrebnih za aktivaciju senzora
                    tipa BROJ </tip>
                <senzori>
                    <senzor naziv="..."/>
                    +++
                </senzori>
            </senzor>
            +++
        </senzori>

        <kontroleri>
            <kontroler>
                <objekat naziv="..."/>
                <!-- definise se ili translacija ili rotacija -->
                <translacija>
                    <x>...</x>
                    <y>...</y>
                    <z>...</z>
                </translacija >
                <rotacija>
                    <ugao> ugao rotacije u stepenima </ugao>
                    <osa>
                        <x>...</x>
                        <y>...</y>
                        <z>...</z>
                    </osa>
                    <ofsetOse>
                        <!-- vektor pomeraja ose rotacije u
                            odnosu na centar objekta -->
                        <x>...</x>
                        <y>...</y>
                        <z>...</z>
                    </ofsetOse>
                </rotacija>
            </kontroler>
            +++
        </kontroleri>

        <mete>
            <meta naziv="...">
                <roditelj naziv="naziv objekta za koji je vezana
                    lokacija mete"/>
                <!-- navodi se ili naziv ili vrsta objekta -->
```

```
                <objekat naziv=”назив објекта који се може
                        спустити на мету”
                        vrsta=“врста објекта који се може
                        спустити на мету“/>
                <velicina>
                        <x>...</x>
                        <y>...</y>
                        <z>...</z>
                </velicina>
                <lokacija>
                        <x>...</x>
                        <y>...</y>
                        <z>...</z>
                </lokacija>
                <orijentacija>
                        <x> ротација мете у степенима око x-осе </x>
                        <y>...</y>
                        <z>...</z>
                </orijentacija>
        </meta>
        +++
</mete>

<alarmi>
        <alarm>
                <objekat naziv=”назив објекта за који се
                        везује аларм”/>
                <senzor naziv="назив сензора чија активност
                         условљава укључење аларма"/>
                <tipZone>bezbedna | opasna</tipZone>
                <r> радијус зоне у метрима </r>
        </alarm>
        +++
</alarmi>

<okidaci>
        <okidac naziv="...">
                <akcija naziv="назив акције коју активира окидач"/>
                <senzor naziv="..."/>
                <vreme> време у милисекундама за које је
                        закашњена активација акције </vreme>
        </okidac>
        +++
</okidaci>

<akcije>
        <akcija naziv="...">
                <objekat naziv="назив објекта који треба да
                        држи актер приликом активирања акције"
                        vrsta="врста објекта који треба да
                        држи актер приликом активирања акције"/>
                <senzor naziv="сензор чија активност условљава
                        извршење акције"/>
                <kontroleri>
                        <!-- листа контролера које активира акција -->
```



```
                        <kontroler naziv="..."/>
                        +++
                </kontroleri>
        </akcija>
        +++
</akcije>
</kontrola>
</scena>

<scenario pocetniZadatak=”назив почетног задатка”>
        <zadatak naziv=”...”>
                <zvuk> назив фајла са звучним упутством </zvuk>
                <slika>...</slika>
                <uputstvo>...</uputstvo>

                <bodovi>
                        <max> максималан број бодова који носи задатак </max>
                        <min> минималан број бодова који се добија за извршење
                                задатка по истеку задатог времена </min>
                        <vreme> време за извршење задатка у секундама </vreme>
                </bodovi>

                <sekvence>
                        <sekvenca koeficijent=”коефицијент којим се множи број
                                        остварених бодова”
                                tip=”kompletiranje | resetovanje”>
                                <senzor naziv=”...”/>
                                <okidac naziv=”...”/>
                                <senzor naziv=”...”/>
                                +++
                                <zadatak naziv=”следећи задатак”/>
                        </sekvenca >
                        +++
                </sekvence>
                <demo>
                        <kontrolnaTacka naziv=”...”>
                                <lokacija>
                                        <x>...</x>
                                        <y>...</y>
                                        <z>...</z>
                                </lokacija>
                                <orijentacija>
                                        <x>...</x>
                                        <y>...</y>
                                        <z>...</z>
                                </orijentacija>
                                <akcija tip="UZIMANJE | SPUSTANJE | IZVRSENJE"/>
                                <meta naziv="назив мете која учествује у акцији"/>
                                <objekat naziv="назив објекта који треба
                                        узети или селектовати у инвентару"/>
                        </kontrolnaTacka>
                </demo>
        </zadatak>
        +++
</scenario>
```

</lekcija>

9.3. Прилог III – Опис формата фајла са бодовима

```
<?xml version="1.0" ?>
<bodovi>
        <test>
                <datumVreme> датум и време тестирања </datumVreme>
                <lekcija id=”…” naziv=”…”/>
                <korisnik id=”…”/>
                <brBodova>
                        <ukupno>…</ukupno>
                        <zadatak  naziv=“...“> број бодова </zadatak>
                        +++
                </brBodova>
                <trajanje>…</trajanje>
        </test>
        +++
</bodovi>
```

9.4. Прилог IV – Опис корисничког профила

```
<?xml version="1.0" ?>
<korisnik id="...">
        <ime> име корисника </ime>
        <grupa> група којој корисник припада </grupa>
        <komentar> коментар који описује корисника </komentar>
        <uputstvo>
                <zvuk/>
                <slika/>
                <tekst/>
        </uputstvo>
        <rezimiIzvodjenja podrazumevani="подразумевани режим, ако је дефинисан">
                <pokazivanje/>
                <ucenje/>
                <testiranje/>
        </rezimiIzvodjenja>
        <lekcija id="..." naziv="назив подразумеване лекције"/>
</korisnik>
```