

## РАЗВОЈ АРКАДНЕ 2Д ИГРИЦЕ КОРИШЋЕЊЕМ ПРОГРАМСКОГ ЈЕЗИКА ЛУА – АРХИТЕКТУРА, АЛАТИ И РЕШЕЊЕ

Saša Lazarević, Biljana Vulićević

**РЕЗИМЕ:** Lua је скриптни језик који постаје све популарнији у индустрији видео игара. Поред мале величине и добрих перформанси, главна предност је компатибилност са другим језицима. Већина игара има *game engine* у C++ који се бави графиком и физиком, а сама игра је креирана изнад тог *game engine*-а у Луи. Међу корисницима мобилних телефона, најпопуларнији жанр игара су аркадне игре. Овај жанр у први план ставља саму радњу игре, нудећи играчу кратке нивое који постају све изазовнији и тежи како игра напредује. У раду је приказан процес развоја аркадне игре, од дизајна до имплементације у развојном оквиру Corona SDK. Разматрају се и предности и мане коришћења скриптног језика Lua.

**КЉУЧНЕ РЕЧИ:** скриптни језик, Lua, аркадна игра, Corona SDK

**ABSTRACT:** Lua is a scripting language that is becoming more popular in the gaming industry. Apart from its small size and performance, Lua's major advantage is the compatibility with other programming languages. Most games have a game engine written in C++ that deals with graphics and physics, while the game itself is created in Lua, above that game engine. Arcade games are the most popular mobile games genre. This genre focuses on the action itself. It offers short levels that become more challenging as the game progresses. This paper shows the proces of developing an arcade game, from design to implementation in Corona SDK. Advantages and disadvantages of using Lua are also considered.

**KEY WORDS:** scripting language, Lua, arcade game, Corona SDK

### 1. УВОД

Lua је моћан и ефикасан скриптни језик, дизајниран у Бразилу, 1993. године. Поддржава процедурално програмирање, објектно оријентисано програмирање, функционално програмирање. На португалском Lua значи Месец. [5] Један од оснивача Луе, Роберто Јерузалимски рекао је о Луи: "Луа је језик који је од почетка креиран као скриптни језик који ће се користити заједно са другим језицима. Дозвољава занимљиве конструкције и нове начине програмирања. Поред мале величине и добрих перформанси, главна предност Луе је њена компатибилност са другим језицима. Већина игара има *game engine* у C++ који се бави графиком и физиком, а сама игра је креирана изнад тог *game engine*-а у Луи.[8]" Луа је постала веома популаран програмски језик и сада постоји значајан број програмера у индустрији видео игара који могу да преносе знање између компанија. Овоме је допринела брзина и лакоћа са којом девелопери могу да уграде Lua-у у *game engine* и да прошире њену функционалност екстензијама писаним у другим језицима, као и брзо извршавање и лакоћа савладавања језика. Многе видео игре су писане користећи Lua-у као скриптни језик. Међу најпознатијима су World of Warcraft и Angry Birds.

### 2. ДЕФИНИЦИЈА ИГРЕ

Ралф Костер дефинише игру као интерактивно искуство које пред играча ставља низ захтевних шаблона које он или она учи и на крају савлада. То значи да су учење и превазилажење препрека срж онога што зовемо забава (као што виц постане смешан оног тренутка када га схватимо).

Свет је пун система којима можемо да приступимо као игри. Игре су слагалице које треба решити, као и све на шта наилазимо у животу. Као када учимо да возимо аутомобил, свирамо неки инструмент, таблицу множења. Учимо патерне по којима ствари функционишу, савладамо их, и на крају их складиштимо тако да их можемо поново користити по потреби. Највећа разлика између игре и реалности је у томе што су улози много мањи код игре[3].

Већина 2Д и 3Д игара су "меке" интерактивне компјутерске симулације у реалном времену, засноване на агентима.[1]

Симулација - У играма је неки део реалног света представљен преко математичког модела како би компјутер могао да управља њиме. Модел је апроксимација и симплификација реалности, па је игра симулација имагинарног или реалног света игре.

Симулација заснована на агентима је она у којој више ентитета врши интеракцију. У видео играма ови ентитети могу бити возила, карактери, ватрене кугле итд.

Интерактивна временска симулација - Видео игре су динамички системи, стање у свету игре се мења кроз време. Игра такође мора да одговори на непредвидиве потезе играча.

Меке симулације. Срж сваког система у реалном времену је концепт крајњег рока, ограничења. На пример, у видео играма постоји потреба да се екран ажурира бар 24 пута у секунди да би постојала илузија покрета, физичка симулација мора да се ажурира 120 пута у секунди да би остала стабилна док аудио библиотека мора да се позове бар 60 пута у секунди да би аудио бафери били пуни и да звук не би "сецкао". Мека симулација значи да пробијање рока неће имати катастрофалне последице. Са друге стра-

не, у чврстим реалним системима, пробијање ограничења може да значи озбиљне повреде или чак смрт људског оператора. Примери чврстих реалних система су авионски систем у хеликоптеру или контролни систем у нуклеарној електрани.

Игре су дискретни системи, стање се одређује у сваком дискретном тренутку времена. Главна петља игре се стално понавља, и током сваке итерације петље се израчунава и ажурира стање система који чине игру (вештачка интелигенција, логика игре, физичка симулација). Резултати су онда рендеровани приказивањем графике, емитовањем звука итд.

### 3. ЖАНРОВИ РАЧУНАРСКИХ ИГАРА

#### 3.1 Аркадне игре.

Овај жанр у први план ставља саму радњу игре, нудећи играчу кратке нивое који постају све изазовнији и тежи како игра напредује. Основне карактеристике су кратко време трајања, тренутни изазови и једноставна радња. Аркадне игре су најпопуларније међу онима које се играју на мобилним телефонима. [13] Тркачке аркадне игре имају поједностављену симулацију физичких објеката (physics engine) и захтевају мање возачких вештина од правих тркачких симулација. Аркадни аутомобили могу да скрену брзо без кочења, а противнички аутомобили су често програмирани да увек буду близу играча [17].

#### 3.2 Акционе видео игре

Моторне вештине и координација око-рука су најбитније у овом жанру који наглашава физичке изазове.

First-Person Shooters (FPS) - Један од најпопуларнијих акционих жанрова у коме играч има улогу стрелца. Играчи користе разно оружје да би учествовали у акцији која се одвија у даљини. Већина укључује насиље и смртоносна оружја којима се наноси штета противницима. Стрелачке игре су обично једне од технолошки захтевнијих за развијање, јер овај тип игре има за циљ да играчима приушти детаљну и верну илузију хипер реалног света. [1]

Платформери - Радња се врти око скакања и пењања по окружењу да би се избегле препреке и непријатељи. Вертикална компонента је пренаглашена, како би приказ терена давао више прилика за прескакање или пад. Кретање ликова пркоси гравитацији, они могу да скоче јако високо/далеко и да контролишу своје кретање у ваздуху како би превазишли препреке.

Борилачке видео игре - Обично су игре за два играча који се боре у некаквој врсти ринга. Технологија борилачких игара инсистира на разноврсним анимацијама борилачких покрета и тачној детекцији удараца а улазни систем је способан да детектује комплексне комбинације дугмади и joystick-а. Графички напредне борилачке игре верно приказују борце чак до детаља модрица и капи зноја, покрета праменова косе и одеће. [1]

#### 3.3 Авантуристичке видео игре

Насупрот авантуристичким филмовима, авантуристичке игре нису дефинисане причом или концептом, већ начином играња у коме победа не зависи од рефлекса или акције. Од играча се очекује да реши разне загонетке кроз интеракцију са људима или окружењем, без сукобљавања и акције.

#### 3.4 Играње улога

Видео игре са играњем улога вуку порекло од класичних игара као што су Тамнице и змајеви. Играч се ставља у једну од улога авануриста који су специјализовани за одређену групу вештина (нпр борба прса у прса или бацање чини) док напредују кроз унапред смишљен заплет. Овим улогама се обично управља са главног подручја (*overworld*) на коме су чудовишта, и са кога играч има приступ важнијим локацијама игре као што су градови, тамнице, градови. Овај жанр је међу првима стекао популарност од распрострањености персоналних рачунара. [17]

#### 3.5 Стратегије у реалном времену

У овом жанру играч стратешки управља својим војним јединицама на великом бојном пољу и покушава да порази противника. Свет игре је обично приказан из птичје перспективе.

Остали жанрови укључују спортске игре, игре са картама, слагалице, неформалне.

### 4. РАЗВОЈ АРКАДНЕ 2Д ИГРЕ

У првој фази развоја игре одређује се какав ће бити кориснички интерфејс и контроле играча, које ће платформе постојати, дизајнирају се нивои. Прави се документ дизајна игре у коме се детаљно описују сви аспекти игре - циљна група, радња игре, правила и структура игре, графика, ликови, нивои. [12]

#### 4.1 Документ дизајна игре

##### 1. Опис игре

###### 1.1. Концепт

Корисник игра нинџу који покушава да се попне до врха куле успут избегавајући препреке.

###### 1.2. Жанр

Аркадна игра

###### 1.3. Циљна публика и платформа

Власници андроид телефона, старости 15-30 година

##### 2. Gameplay и механика

###### 2.1. Gameplay

###### 2.1.1. Циљеви

Избегавати непријатеље и преживети што је дуже могуће.

###### 2.1.2. Ток игре

На почетку игре непријатељи су птице које се појављују са леве или десне стране и лете ка играчу, као и камење које пада на доле. После 200 поена, играч прелази на ниво 2, и игра се убрзава. Осим камења и птица, играча нападају и ватрене кугле одозго. После 400 поена, играч прелази на ниво 3, игра се додатно убрзава. Камење и птице се крећу брже и теже их је избећи. Ватрене кугле падају укоса ка играчу. У сваком нивоу се повремено на екрану појављује штит, који када га играч покупи, даје имунитет који траје 5 секунди.

2.2. Механика

2.2.1. Кретање

Играч може да мења стране, са леве на десну и обрнуто, као и да се спусти да би избегао удар птице.

2.2.2. Губитак живота

Играч има три живота на почетку игре. Сваки пут када се судари са неким од непријатеља, губи по један живот. Када изгуби сва три живота, игра је завршена.

3. Интерфејс

3.1. HUD

3.1.1. Аватар главног карактера

3.1.2. Приказ броја поена

3.1.3. Приказ здравља

3.1.4. Приказ нивоа

3.1.4. Дугме за искључивање/укључивање звука

3.2. Контроле

3.2.1. Додиром екрана играч мења страну

3.2.2. Превлачењем на доле играч може да се спусти

3.3. Аудио, музика и звучни ефекти

3.3.1. Позадинска музика у току игре

3.3.2. Звукови удараца када се лик судари са птицом или каменом

3.3.3. Звук ватре када на лика падне ватрена кугла

3.3.4. Звук скока када играч мења страну

4. Game Art

4.1 Позадина

Две паралелне куле са мердевинама преко којих се лик пење ка врху

4.2 Главни лик и анимације

Играч може да бира између мушког и женског лика.

4.2.1 секвенца анимације на почетном екрану

4.2.2 секвенца анимације пењања

4.3.3 секвенца анимације падања/умирања

4.3 Непријатељи и анимација

4.3.1 птице које лете лево-десно

4.3.2 камење које пада са врха

4.3.3 ватрене кугле које падају са врха

4.4 HUD (Head's Up Display)

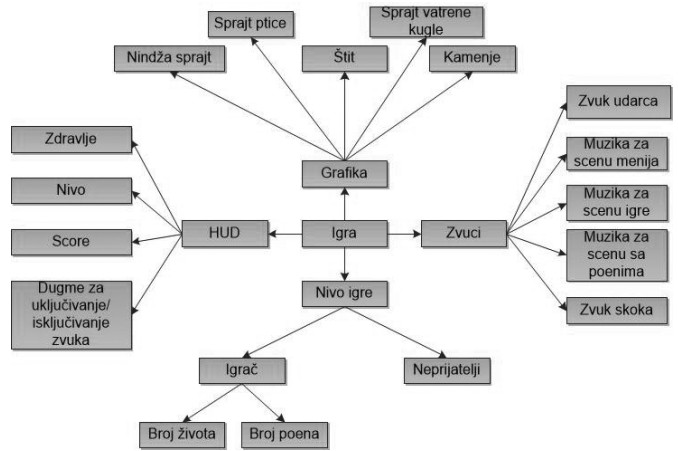
4.5 Почетни екран

4.6 Страница помоћ

4.7 Страница са бројем поена

4.8 Страница са информацијама о игри

На Слици 1 Елементи игре приказани су елементи који сачињавају игру.

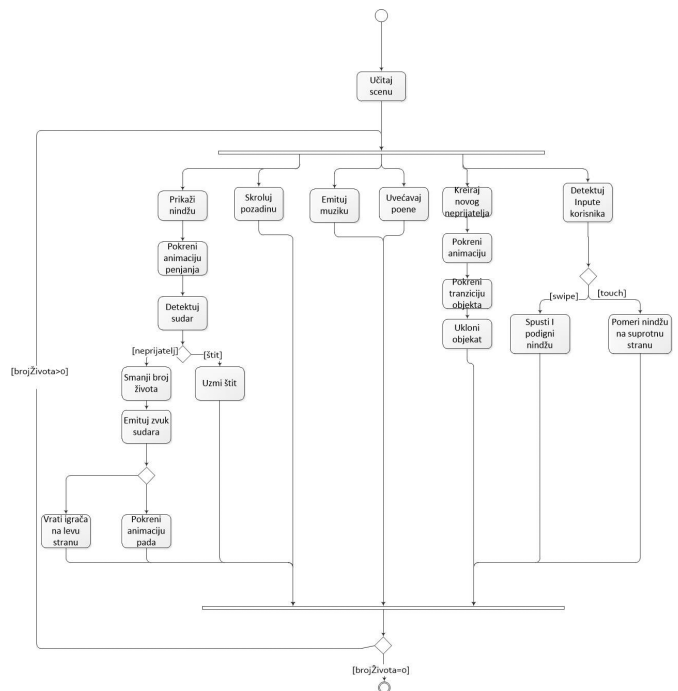


Слика 1 Елементи игре

4.2 ДИЈАГРАМ АКТИВНОСТИ ИГРЕ

Дијаграм активности приказује који се то паралелни процеси одвијају у току трајања игре

- Праћење стања нинце, анимације и детекција колизије
- Позадина се са сваком фрејмом помера на горе
- Емитује се музика
- Сваке секунде се повећава број поена
- Креирају се нови непријатељи, покреће се анимација и транзиција тих објеката. Када такав објекат нестане са екрана, брише се из меморије
- Обрађују се инпути корисника преко корисничког интерфејса



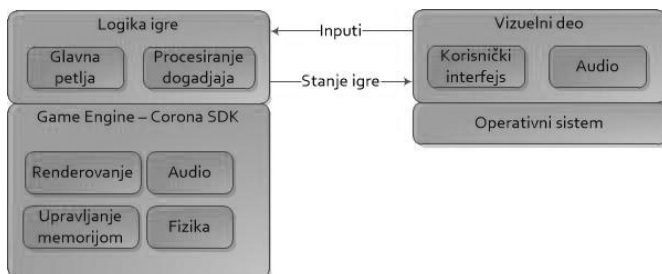
Слика 2 Дијаграм активности игре

### 4.3 Архитектура игре

На Сlici 3 је приказана архитектура игре.

Corona је развојни оквир који се користи за брзи развој апликација и игрица за мобилне телефоне и desktop рачунаре. Пројекат креиран у Corona-и може да функционише на разним типовима уређаја укључујући Apple iPhone и iPad, Android телефоне и таблете, Amazon Fire, Mac рачунаре, Windows рачунаре, итд. Заснован је на програмском језику Lua.[6]

Развојни оквир у себи садржи софтверске компоненте за приказивање графике, физичку симулацију и емитовање аудио садржаја. Логика игре је написана у скриптном језику Lua, користећи многобројне Corona API-је. Инпути корисника као на пример додир екрана или притисак дугмета представљају догађаје који утичу на игру. Ови догађаји се процесирају и кориснику се на уређају приказује измењено стање игре.



Слика 3 Архитектура игре

### 4.4 Понашање главног лика

Догађаји су покретачи игре. А у контексту игре догађај је све оно што се дешава у току игре а да има некакав утицај на стање игре – на пример експлозија, непријатељ је уочио играча, играч је притиснуо дугме на улазном уређају, истекло је време предвиђено за неки део итд. Игра мора да обавести објекте од интереса да се десно догађај и да омогући да ти објекти одговоре. Реакција игре на догађај је један од кључних аспеката понашања игре. Објекти игре комуницирају тако што пошљалац креира структуру података event која садржи тип и аргументе који се шаљу. Event се даље прослеђује примаоцу који позива eventhandler функцију. У Corona-и постоји два типа догађаја, локални и глобални (Runtime).[2]

Runtime догађаји су активни током целе дужине трајања апликације. Ови догађаји нису усмерени на неки одређен објекат, већ се одашиљу ка свим заинтересованим ослушкивачима. Неки од најчешће коришћених Runtime догађаја су

- enterFrame догађаји, који се се окидају са фреквенцијом број оквира у секунди
- system догађаји, који се шаљу када се десе неки екстерни догађаји, као нпр. када уређај изађе из апликације – applicationStart, applicationExit, applicationSuspend, applicationResume, applicationOpen

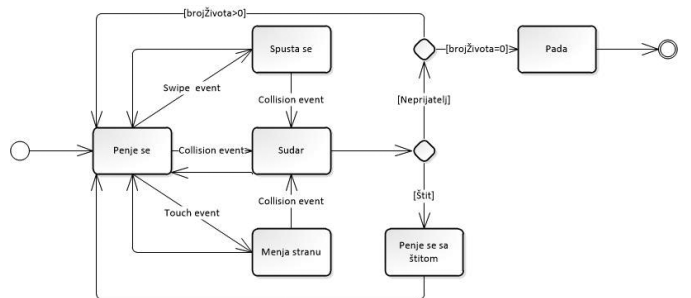
- resize догађаји, дешавају се када се висина или ширина апликације промени
- orientation догађаји, дешавају се када се промени оријентација уређаја из portrait у landscape и обрнуто

Локални догађаји се шаљу само једном ослушкивачу а не глобалном Runtime ослушкивачу. Примери локалних догађаја

- tap и touch догађаји се дешавају када корисник додирне екран уређаја
- collision догађаји се дешавају када се два физичка објекта сударе
- timer догађаји када истекне тајмер
- audio догађаји, нпр када се заврши аудио фајл

Ослушкивачи могу да буду функције или таблице/display објекти. Функцији ослушкивачу се као параметар прослеђује табела event.

Основа игре описане у документу дизајна је главни лик који се констатно пење на горе, успут избегавајући препреке. Да бисмо створили илузију кретања у простору, помераћемо позадину. EnterFrame је глобални догађај који се окида са учесталошћу од 30 или 60 оквира у секунди, у зависности од тога како је подешено у конфигурационом фајлу config.lua. Како би позадина била без краја, направимо две променљиве bg1 и bg2 које ће користити исти ресурс (слику), али ће се оне наизменично смењивати на у оси кретања се на горе.



Слика 4 Дијаграм промене стања

У току игре главни лик може да буде у различитим стањима у зависности од догађаја који се детектују. Ако играч има преосталих живота, ница се пење док се не детектује неки други догађај. Корисник додиром екрана скаче са једне на другу кулу, и спушта се превлачењем на доле. Када се деси локални догађај touch, тада се извршава функција onTouchScreen која испитује разлику између почетне и крајње позиције параметра event на x оси. Ако је разлика већа од 10 пиксела, реч је о swipe покрету и позива се функција swipe\_down која помера главни лик за 200 пиксела на доле, а затим га током 0.8 секунди враћа на почетни положај. У супротном се позива функција change\_side која главни лик помера на супротну кулу од оне на којој се тренутно налази. Када систем детектује колизију нинце са неким другим објектом - ако је други објекат штит, нинца преузме штит и настави да се пење са имунитетом који траје 5 секунди. У супротном, ако је други објект

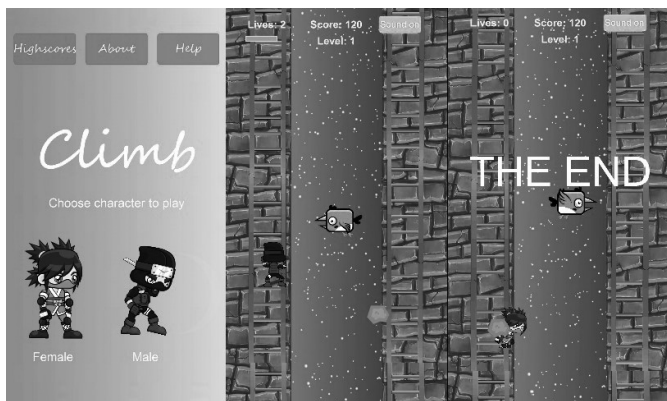
кат непријатељ (птица, камен или ватрена кугла), играчу се одузима један живот. Ако нема више живота, нинџа пада и игра се завршава. У супротном, наставља да се пење.

#### 4.5 Пројектовање корисничког интерфејса

Са почетног екрана корисник притиском на дугме може да оде на странице

- Highscores, која садржи табелу са поенима
- About, на којој су опште информације о игри
- Help, која садржи упутства за игру

Са ових страница се корисник враћа назад на почетни екран притиском на дугме *Back*. На почетном екрану корисник бира лик са којим ће играти и започиње игру.



Слика 5 Детаљи из игре

## 5. ПРЕДНОСТИ И НЕДОСТАЦИ ПРОГРАМСКОГ ЈЕЗИКА LUA ЗА РАЗВОЈ ИГАРА

Мала је. Lua заузима јако мало простора. Код најновије верзије 5.3.5, Lua интерпретер са свим стандардним библиотекама заузима око 247 килобајта а изворни код се састоји од око 24000 линија писаних у C-у. Ово је посебна предност код апликација за мобилне телефоне, који често имају ограничену меморију и сваки килобајт је важан. Минимално коришћење меморије је важно чак и код телефона са већим ресурсима јер је саче меморија процесора мала, и приступ меморији је спор у поређењу са процесорима десктоп рачунара.[7]

Брза је. Lua је стекла репутацију због својих перформанси. Многа истраживања показују да је Lua најбржа у домену интерпретираних скрипних језика, и значајни делови великих апликација су написани у овом програмском језику. Додатну брзину може донети коришћење LuaJIT (just in time) компајлера.[5]

Портабилна је. Lua ради на свим платформама које имају стандардан C компајлер. То подразумева све дистрибуције Unix-а и Windows-а, мобилне уређаје које користе Android, iOS, BREW, Windows Phone итд.

Бесплатна је. Lua је бесплатни open-source софтвер, дистрибуирана је под MIT лиценцом. Може се користити у било коју сврху, чак и комерцијално, без икаквих трошкова.

Проширива је. Lua је дизајнирана да ради заједно са C-ом, и нуди флексибилан C API. Повезивање са C библиотекама је лако и обично у пар линија кода.

Лако се учи. Lua има једноставну али моћну синтаксу, семантика је конзистентна и интуитивна. Постоји добра документација и почетници могу одмах да почну да је користе, са било којим текстуалним едитором, нема потребе за великим интегрисаним развојним окружењем. Са друге стране, Lua има и напредне структуре као што су closure, ко-рутине, мета механизми, могућа је примена објектно оријентисаног програмирања. Ове предности је чине привлачним за професионалце, али не отежавају коришћење почетницима.[7] Значајна предност Lua је и постојање великог броја game engine-а који користе Lua-у.

Lua као додатни језик - Једна од предности Lua, да је развијена да буде компатибилна са другим језицима и да се лако може уградити, може се посматрати и као недостатак. Није развијана са циљем да буде примарни језик самосталних апликација, већ секундарни језик.

JIT и iOS - Конзоле и iOS не дозвољавају JIT (Just In Time) компајлирање, па Lua може бити спорија од C++, нарочито за математички захтевне операције. На овим платформама LuaJIT се извршава само као интерпретер, што тада чини LuaJIT само 3 пута бржом од обичне Lua (са JIT компајлирањем LuaJIT је 5-100 пута бржи од Lua).[11]

Нема статичких типова - Динамичко типизирање је често бржи начин током развоја, међутим показало се као отежавајућа околност у фази одржавања. Lua је лака за писање делом зато што тип варијабле не мора да се специфицира, већ интерпретер сам одређује тип варијабле приликом извршавања. Велики део грешака које се јаве у току извршавања програма везане су баш за тип варијабле, на пример користимо варијаблу мислећи да је број а испостави се да је стринг.

Табеле су једина структура података у Луи и користе се за имплементирање многих типова података. Суштински табела у Луи је асоцијативни низ, који може бити индексан не само бројевима већ и стринговима, и било којом вредношћу осим нил. Преко табела се могу креирати низови, матрице, листе, глобалне варијабле, модули, објекти и класе. Међутим, као што и сами дизајнери Lua-е наводе, имитација неке структуре података није једнако добра као оригинал.[9]

## 6. ПРЕДНОСТИ И НЕДОСТАЦИ РАЗВОЈА У ОКВИРУ CORONA SDK

Предности

- Користећи Corona SDK одједном се пише код који ће радити на више платформи, па је самим тим брже и јефтиније него развијати нативне апликације за сваку платформу посебно. [14]
- Интерактивни Corona симулатор одмах реагује на промене у коду, и у реалном времену пружа приказ како би апликација изгледала на правом физичком уређају[6]

- Нема интегрисано развојно окружење, за развој је потребан само текстуални едитор[14]
- Скалабилност – лако је креирати апликацију која ради на много различитих величина екрана
- Може се користити било која нативна библиотека (C/C++/Obj-C/Java)
- Добра документација
- Аудио и сликовне датотеке се лако приказују уз пар линија кода
- Све што креирате у Corona-и је ваше власништво и имате пуно право на код пројекта

#### Недостаци

- Подржава само 2Д, не и 3Д
- Потребна је интернет конекција за build пројекта
- Не подржава конзоле (Playstation, Xbox)
- Апликације које су развијане за више платформи имају мању респонзивност од нативних апликација[16]
- Нативне апликације могу у потпуности да искористе могућности платформе и уређаја, док су апликација за више платформи ограничене могућностима развојног оквира у коме су креиране
- Корисници android и iOS имају различите навике и очекивања од дизајна. Ако се развија једна апликација за обе платформе мора се или направити у компромис у коме није све по вољи обе стране, или направити две специјализоване апликације[16]

#### 7. ЗАКЉУЧАК

Са Corona SDK развојним оквиром, у исто време настаје код који је компатибилан са свим платформама, што значајно убрзава развој. Интерактивни Corona симулатор одмах реагује на промене у коду, и у реалном времену пружа приказ како би апликација изгледала на правом физичком уређају.[6] Lua је брза и заузима мало меморије, што је посебно важно код уређаја са ограниченом меморијом, као нпр код мобилних телефона. Кроз коришћење мета-механизама, могуће је имплементирати концепте класе и наслеђивања.

#### 8. ЛИТЕРАТУРА

[1] Jason Gregory, Game Engine Architecture, A.K. Peters, 2009  
 [2] Michelle M. Fernandez, Corona SDK Mobile Game Development, Packt Publishing, 2015

[3] Ralph Coster, Theory of Fun for Game Design, O'Reilly Media, 2014  
 [4] Programming in Lua, Roberto Ierusalimschy, Lua.org, December 2003  
 [5] About Lua, <https://www.lua.org/about.html>, датум последњег приступа септембар 2018  
 [6] Corona docs, <https://docs.coronalabs.com/>, датум последњег приступа септембар 2018  
 [7] Advantages Lua, <http://lua-users.org/lists/lua-l/2007-11/msg00248.html>, датум последњег приступа септембар 2018  
 [8] Lua Founding Developer Shares the Scripting Language's Journey and Advantages for App Configuration and Data Management, <https://www.hostingadvice.com/blog/lua-founding-developer-shares-language-advantages/>, датум последњег приступа септембар 2018  
 [9] Lua workshop 2016, <https://www.lua.org/wshop16/Ierusalimschy.pdf>, датум последњег приступа септембар 2018  
 [10] Why Lua, [https://help.autodesk.com/view/Stingray/ENU/?guid=stingray\\_help\\_creating\\_gameplay\\_scripting\\_with\\_lua\\_why\\_lua\\_html](https://help.autodesk.com/view/Stingray/ENU/?guid=stingray_help_creating_gameplay_scripting_with_lua_why_lua_html), датум последњег приступа септембар 2018  
 [11] Goodbye Lua, <https://realmensch.org/2016/05/28/goodbye-lua/>, датум последњег приступа септембар 2018  
 [12] Tutorial Lua, <https://www.tutorialspoint.com/lua/>, датум последњег приступа септембар 2018  
 [13] Overview of 10 Most Popular Mobile Game Genres, <http://rstylelab.weebly.com/blog/overview-of-10-most-popular-mobile-game-genres>, датум последњег приступа август 2018  
 [14] Corona advantages, <https://dzone.com/articles/is-corona-sdk-the-best-mobile-game-development-pla>, датум последњег приступа август 2018  
 [15] Corona pros and cons, <http://inquisiment.com/blog/2014/08/corona-pros-cons.html>, датум последњег приступа август 2018  
 [16] Cross platform and native apps, <http://www.businessofapps.com/cross-platform-vs-native-app-app-development-pros-and-cons/>, датум последњег приступа август 2018  
 [17] Game genres, [https://en.wikipedia.org/wiki/List\\_of\\_video\\_game\\_genres](https://en.wikipedia.org/wiki/List_of_video_game_genres), [https://en.wikipedia.org/wiki/List\\_of\\_video\\_game\\_genres](https://en.wikipedia.org/wiki/List_of_video_game_genres)



**Биљана Вулићевић**, студент мастер студија на Факултету организационих наука, смер Софтверско инжењерство и рачунарске науке  
**Контакт:** [vulicevicbiljana2009@gmail.com](mailto:vulicevicbiljana2009@gmail.com)  
**Области интересовања:** Развој Interactive Voice Response апликација, Развој мобилних апликација, Базе података, Информациони системи



**Проф. Др Саша Лазаревић**, професор на Факултету организационих наука  
**Контакт:** [slazar@fon.rs](mailto:slazar@fon.rs)  
**Области интересовања:** софтверско инжењерство, информациони системи, базе података, системи за управљање документацијом, .NET платформ

