

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА, НОВИ САД

# Документација за пројекат

---

Bubble Trouble

1/15/2019

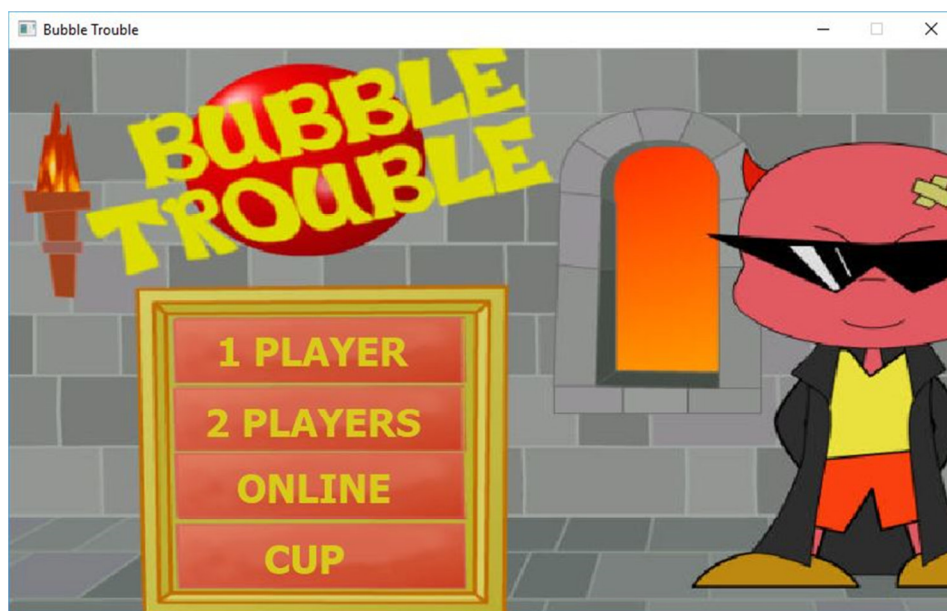
Босиљка Братић,  
Обрен Милошевић,  
Небојша Кнежевић,  
Данило Милошевић

## КЛИЈЕНТ

Постоје две врсте клијената у апликацији, а то су:

- Клијент за игру преко мреже и
- Клијент за *офлајн* мод

Клијентска логика у ова два мода се доста разликује јер клијент за игру преко мреже не врши никакве калкулације, то за њега ради сервер, а клијент за офлајн мод мора сам то да ради. Бирање мода у коме ће се играти врши се из главног менија који се појављује при покретању апликације. За имплементацију цијелог пројекта коришћена је *PYQT5* библиотека.



Слика 1 – Главни мени игрице

### *Клијент за офлајн мод*

Након покретања менија кориснику се нуде три могућности играња у *офлајн* режиму: један играч, два играча и турнир.

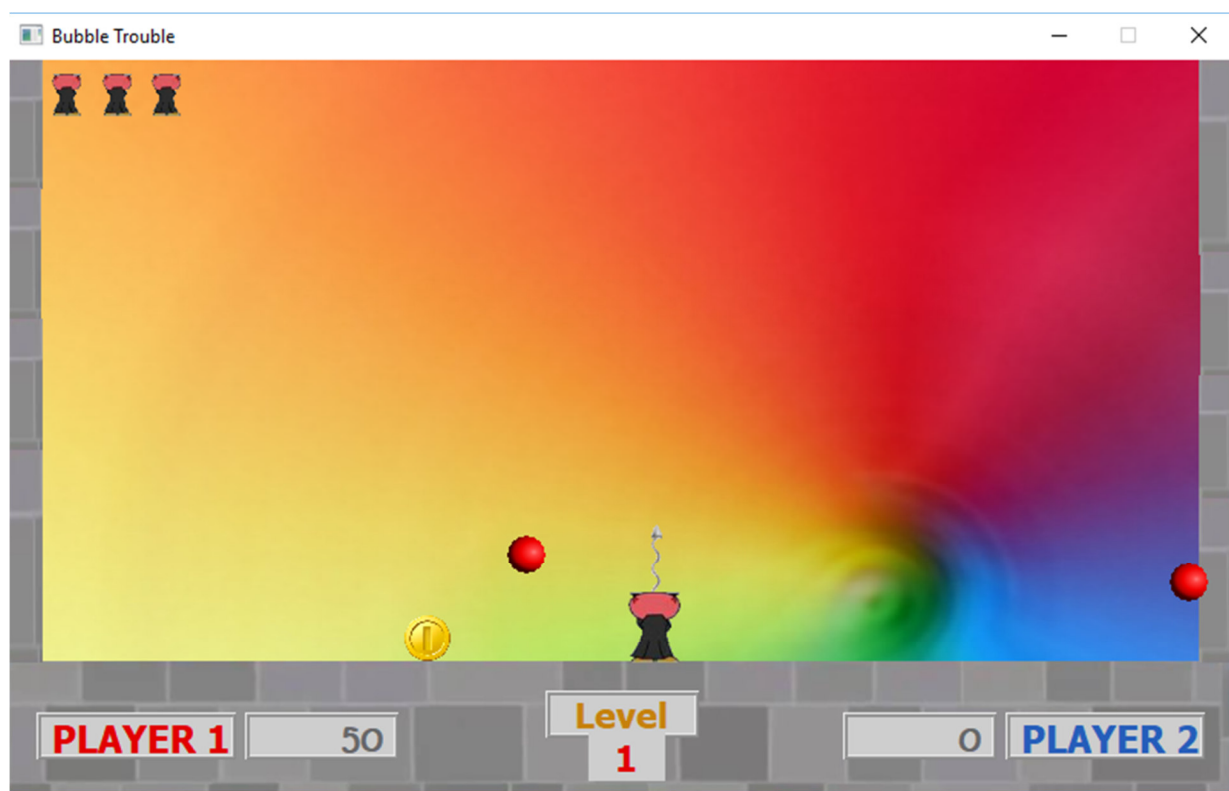
Игрица почиње тако што се појављују играчи (зависно од изабраног мода) и једна лоптица. Први играч се креће користећи тастере са стрелицама на

тастатури, а други користећи тастере *A* и *D*, док оружје први користи притиском на дугме *space*, а други *shift*.

Кретање играча контролисано је из методе *update* коју има у себи сваки играч. Ажурирање позиције се врши у тајмеру који позива методу на сваке 32 милисекунде. Тада се играч зависно од смјера кретања помјери за одређени број пиксела, уз ажурирање слике играча према смјеру кретања.

Помјерање лоптице ради се из истог тајмера према формули  $|sin(ax)|$ . Лоптица се у случају колизије са оружјем дијели на двије мање, осим у случају када је једнака минималној величини и када је коначно уништена.

Сваки играч има своје оружје, које иде од величине играча до врха ако не погоди ниједну лоптицу, а ако погоди онда нестаје. Оружје се помјера према врху из тајмера којег сваки играч има.



Слика 2 – Оружје, подијељене лоптице, бонус

Играч на почетку има 3 живота, а уколико дође до колизије са лоптом он губи живот и ниво се рестартује. Разбијајући лопте играч сакупља поене и може добити и неки бонус. Постоје двије врсте бонуса: новчић и замрзавање оружја. Ако играч покупи новчић добија одређени број поена, а ако покупи други бонус не може да пуца у наредне 2 секунде.

Када све лоптице нестану ниво је пређен. У следећем нивоу може се појавити једна већа лопта или једна више, исте величине, у односу на претходни ниво. Ако играч изгуби сва три живота игра је завршена. Постоји бесконачно нивоа, а сваки следећи ниво је тежи од претходног.

На офлајн клијенту постоје и два процеса који врше калкулације везане за поене и животе и 4 *queue*-а, по два за оба процеса. У један *queue* се смјештају подаци које треба обрадити, а у други процес смјешта нове израчунате вриједности које се из њега касније читају уз ажурирање корисничког интерфејса.

### *Клијент за игру преко мреже*

За овог клијента све калкулације ради сервер на кога се он повезује, а његово је само да тумачи поруку коју му сервер шаље и све исцрта на свом екрану.

Овај клијент покреће два *thread*-а, један за пријем и један за слање порука. Он се повезује на сервер преко задате адресе и порта када се кликне на *Online* мод у менију. Прва порука коју шаље јесте јављање серверу, а све остале поруке садрже само притиснути тастер.

Клијент када прими поруку стави је на свој *queue* и ажурира интерфејс.

## СЕРВЕР

Сервер је компонента која се покреће приликом одабира *онлине* мода унутар менија апликације. У том контексту, серверов посао је да отвори *хост*, чека клијентске конекције (двије) и при успјешном успостављању везе, покреће програмску нит која ће да врши калкулације везане за кретање тог играча у оквиру прозора апликације и освјежавање вриједности промјенљивих везаних за апликацију (игрицу). Такође, сервер при успостави везе са клијентом, чува ту клијентску конекцију у листу конектованих клијената.

Приликом подизања сервера, покреће се још један процес који ће да обрађује клијентске команде које може да шаље играч (тастери: стрелица лијево, стрелица десно, А, Д, размак, *shift*). Притиском на неки од ових тастера, клијент иницира помијерање свог играча за дефинисан корак (одређени број пиксела) и испаљивање оружја чиме покушава да погоди лопте које се крећу по екрану.

Сервер покреће и двије додатне програмске нити које обрађују комуникацију са клијентом. Једна нит прима податке (*RecvThread*) и по приспећу, поруку смјешта у ред (*queue*). Ова структура се просљеђује предходно наведеном процесу како би процес ишчитавао ове команде и обрадио их. Овим се обезбјеђује да се све команде које клијент иницира обраде редом једна по једна. У овој имплементацији, користи се блокирајући режим *socketa* при чему се ова нит која прима податке блокира док клијент не пошаље неку команду.

Друга нит врши слање повратне поруке клијенту из које клијент чита податке односно координате *UI* елемената. Овај посао се врши на сљедећи начин:

Процес који прима команде, обрађује исте тако што провјерава које је дугме притиснуто и у зависности од тога креира поруку која представља дио одговора клијенту пошто он само ажурира тренутну позицију играча на прозору апликације и поставља одређене флегове код испаљивања оружја и такву поруку ставља у ред;

Из тога реда, сервер узима поруке и провјерава флегове за оружје и на основу тога додаје остале податке који су везани за рад апликације (позиције оружја, лопти, бонуса). Након креирања комплетне поруке, сервер емитује сигнал са овом поруком на нит која шаље податке (*SendThread*).

Дакле, серверски процес креира један нови процес за обраду команди, затим за сваку клијентску конекцију креира по једну *send i receive* нит, и једну нит која врши периодично ажурирање елемената које не контролише клијент тј кретање лопти, бонуса, ажурирање оружја, нивоа, поена играча као и провјеру колизија играча са лоптом или лопте са оружјем тако да када дође до колизије играча са лоптом ресетује се тренутни ниво и ажурирају животи играча (иницијално 3 живота), док се при колизији оружја са лоптом или играча са бонусима ажурирају поени које је играч остварио.

Постоје два типа бонуса а то су додатни поени (*coin*) и блокирање оружја на 2 секунде.