# РАСПРАЦОЎКА РЭЛЯЦЫЙНАЙ БАЗЫ ДАДЗЕНЫХ, АПТЫМІЗАВАНАЙ ПАД ПАТРЭБЫ ПРАГРАМАВАННЯ ГУЛЬНЯЎ

**К. І. Тамашэвіч**

*Беларускі дзяржаўны ўніверсітэт, г. Мінск;*
*konstantin.tomashevich@gmail.com*
*нав. кір. – У. М. Гошка, старэйшы выкладчык*

З-за больш высокіх патрабаванняў камп'ютарных гульняў да хуткасці тэхналогій захавання і апрацоўкі дадзеных, выкрыстоўваемыя звычайнымі настольнымі і вэб-прыладамі, рэляцыйныя базы дадзеных не падыходзяць для распрацоўкі гульняў. Але з развіццём так званых «арыентаваных на дадзеныя» шаблонаў распрацоўкі гульнёвай логікі з'яўляецца ўсё больш паралеляў паміж механізмамі гэтых шаблонаў і механізмамі рэляцыйных баз дадзеных. У гэтай працы былі разгледжаны найбольш выкарыстоўваемыя шаблоны праектавання гульнёвай логікі і знойдзены паралелі паміж гэтымі шаблонамі і аналагічнымі механізмамі ў рэляцыйных базах дадзеных. Таксама былі разгледжаны механізмы індэксацыі і алгарытмы злучэння ў рэляцыйных базах дадзеных, неабходныя для рашэння задачы эфектыўнай пастаўкі існасцяў сістэмам у шаблоне Існасць-Кампанент-Сістэма. Пасля гэтага былі пастаўлены патрабаванні да эксперыментальнай убудаванай базы дадзеных, арыентаванай на працу ў якасці пастаўшчыка існасцяў сістэмам у шаблоне Існасць-Кампанент-Сістэма; выканана яе праектаванне і рэалізацыя. Магчымасць выкарыстання распрацаванай эксперыментальнай базы дадзеных была праверана з дапамогай напісання невялікай гульні. У выніку чаго зроблена выснова, што атрыманая эксперыментальная база дадзеных мае добрыя перспектывы да развіцця ў паўнавартасны прадукт.

***Ключавыя словы:*** распрацоўка гульняў, праектаванне гульнёвых сімуляцый, шаблон Існасць-Кампанент-Сістэма, рэляцыйныя базы дадзеных, C++.

## УВОДЗІНЫ

Камп'ютарныя гульні з'яўляюцца дыскрэтнымі сімуляцыямі ў рэальным часе з адной ці больш асобамі, якія прымаюць рашэнні незалежна адна ад адной. Пры гэтым да кроку сімуляцыі прымяняюцца дастаткова жорсткія патрабаванні: у сучасных гульнях крок сімуляцыі павінен завяршацца за 17мс у 99.9% выпадкаў, калі сістэма адпавядае патрабаванням гульні. З-за такіх патрабаванняў звычайныя для класічных настольных і вэб-прыладаў методыкі і тэхналогіі не могуць выкарыстоўвацца для распрацоўкі гульняў, бо гэтыя тэхналогіі арыентаваны пад прадастаўленне вялікай колькасці карыстальнікаў доступу да вялікай колькасці дадзеных з больш мяккімі патрабаваннямі да часу выканання. Такім чынам, арыентацыя на прадастаўленне невялікай групе карыстальнікаў адносна невялікага аб'ёму дадзеных, але з жорскімі патрабаваннямі да хуткасці і стабільнасці часу выканання,

задала іншы вектар развіцця тэхналогіям працы з дадзенымі ў сферы распрацоўкі гульняў.

Тым не менш, у апошнія гады ўсё больш папулярным становіцца аналіз і пошук паралеляў паміж шаблонамі праектавання ў распрацоўцы гульняў і ў звычайных бізнес-прыладах. На аснове гэтых даследванняў быў створаны шаблон праектавання Існасць-Кампанент-Сістэма, які ідэалагічна значна бліжэй да рэляцыйных структур, чым стандарт індустрыі – шаблон Сцэна-Вузел-Кампанент. Шаблон Існасць-Кампанент-Сістэма актыўна развіваецца і знаходзіць прымяненне ў індустрыі: на ім кампанія Blizzard распрацавала гульню Overwatch, а для вядомага гульнёвага рухавіка Unity быў распрацаваны эксперыментальны пакет Unity DOTS. Пры гэтым у шаблоне Існасць-Кампанент-Сістэма на дадзены момант няма адзінай спецыфікацыі алгарытму пошуку патрэбных існасцяў і перадачы іх сістэмам. Таму ў межах гэтай працы было вырашана паставіць эксперымент і напісаць убудаваную базу дадзеных, якая будзе выконваць задачы пошуку і перадачы існасцяў сістэмам.

**ШАБЛОН ІСНАСЦЬ-КАМПАНЕНТ-СІСТЭМА**

Пералічым асноўныя прынцыпы шаблона Існасць-Кампанент-Сістэма:

- Логіка і дадзеныя павінны быць поўнасцю адлучаны адно ад аднаго.
- Кампанент – набор дадзеных без канкрэтнай логікі іх апрацоўкі.
- Існасць – звязаны з унікальным ідэнтыфікатарам набор кампанентаў.
- Сістэма – функцыя ці набор функцый, якія выклікаюцца кожны кадр і выконваюць праход па ўсіх існасцях, якія маюць адпаведныя ўмовам сістэмы кампаненты. Пры гэтым сістэма можа чытаць, мадыфікаваць, ствараць і выдаляць кампаненты.
- Парадак выканання сістэмаў фіксіраваны і не можа змяняцца падчас выканання, але набор актыўных сістэм у агульным выпадку можа змяняцца падчас выканання сімуляцыі.

Поўнае адзяленне дадзеных ад логікі спрашчае паралелізацыю алгарытма сімуляцыі і рэалізацыю механізма абмену дадзенымі ў многакарыстальніцкіх гульнях. Таксама лічыцца, што выкарыстанне гэтага шаблона спрашчае падтрымку праектаў з вялікай кодавай базай. Механізмы гэтага шаблона даволі блізкія да механізмаў рэляцыйных баз дадзеных, таму пры праектаванні базы дадзеных было вырашана ў першую чаргу арыентавацца на гэты шаблон.

## ЗАДАЧА ЭФЕКТЫЎНАЙ ПАСТАЎКІ ІСНАСЦЯЎ СІСТЭМАМ

Асноўнай задачай, якую павінна рашаць распрацоўваемая эксперыментальная база дадзеных, з'яўляецца перадача існасцяў сістэмам у шаблоне Існасць-Кампанент-Сістэма. Разгледзім гэту задачу больш падрабязна. Пры кожным кроку сімуляцыі адбываецца выклік усіх сістэм, якія чакаюць перадачы патрэбных ім існасцяў для выканання аперацый над імі. У залежнасці ад складанасці сімуляцыі колькасць сістэм ў ёй змяняецца ад 10 у простых казуальных гульнях да 200 у гульнях з мноствам складаных механік. Пры гэтым адзін крок павінен выконвацца за 17 мілісекунд у большасці выпадкаў. Дадаткова чакаецца, што большасць адведзенага часу будзе выдзелена на выкананне аперацый над існасцямі, а не на пошук існасцяў для сістэм.

Запыт на існасці ад сістэмы звычайна складаецца з набора пакампанентных патрабаванняў: сістэма ўказвае, якія кампаненты ёй патрэбны для чытання і якія для запісу, а таксама можа дадаваць патрабаванні да значэнняў канкрэтных палёў кампанентаў. Пры гэтым для аптымізацыі звычайна патрабуецца, каб запыт быў канстантай. Адказам на такі запыт з'яўляецца плынь, якая, у залежнасці ад рэалізацыі, прадстаўляе сістэме ці спасылкі на існасці, у якіх ёсць усе запатрабаваныя кампаненты, якія адпавядаюць перададзеным крытэрыям, ці самі наборы запатрабаваных кампанентаў, якія знаходзяцца ў адной існасці.

## ПАТРАБАВАННІ ДА ЭКСПЕРЫМЕНТАЛЬНАЙ БАЗЫ ДАДЗЕНЫХ

Спачатку да базы дадзеных былі пастаўлены агульныя патрабаванні, неабходныя для працы ў любой гульнёвай сімуляцыі, а не толькі ў межах шаблона Існасць-Кампанент-Сістэма:

- База дадзеных павінна быць убудаванай, бо перадача дадзеных па каналу будзе займаць залішне шмат часу.
- Картэжы базы дадзеных павінны наўпрост інтэгравацца з тыпамі дадзеных C++. Гэта дазволіць зрабіць працу з палямі картэжа значна больш хуткай.
- База дадзеных павінна падтрымліваць аналагічныя SELECT, UPDATE, INSERT і DELETE аперацыі.
- База дадзеных павінна падтрымліваць даданне і выдаленне трыгераў, напісанных на C++.
- База дадзеных павінна дазваляваць кэшыраваць указальнікі на картэжы для перадачы модулям, якія не арыентаваны на працу з базамі дадзеных.

- Пры перадачы вынікаў аперацый неабходна пазбягаць капіявання картэжаў, бо капіяванне прыводзіла б да дадатковага выкарыстання памяці і працэсарнага часу.

Для таго каб эксперыментальная база дадзеных магла працаваць як пастаўшчык існасцяў сістэмам, яна павінна мець магчымасць эфектыўна выконваць аперацыі злучэння запісаў. Таму былі дададзены наступныя патрабаванні:

- База дадзеных павінна падтрымліваць стварэнне шчыльных ўпарадкаваных індэксаў і аўтаматычна падтрымліваць іх карэктнасць.
- База дадзеных павінна мець агульны інтэрфэйс для працы з індэксаванымі і неіндэксаванымі палямі.
- База дадзеных павінна адтрымліваць аперацыю злучэння з аўтаматычным падборам алгарытма рэалізацыі.

## ПРАВЕРКА ЭКСПЕРЫМЕНТАЛЬНАЙ БАЗЫ ДАДЗЕНЫХ

Для праверкі магчымасці практычнага выкарыстання распрацаванай эксперыментальнай базы дадзеных у якасці пастаўшчыка існасцяў сістэмам было вырашана распрацаваць аднолькавую трывіяльную гульню ў двух варыянтах: з дапамогай распрацаванай базы дадзеных і з дапамогай EnTT – вядомай рэалізацыі шаблона Існасць-Кампанент-Сістэма з адчыненым зыходным кодам. У распрацаванай для праверкі гульні задачай гульца з’яўляецца пошук прадуктаў у супермаркеце падчас эпідэміі: каб не заразіцца і паспяхова завяршыць гульню, неабходна мінімізаваць кантакты з кіруемымі штучным інтэлектам гульцамі. У выніку гэтай праверкі было вырашана, што хоць у распрацаванай базы дадзеных ёсць некаторыя недахопы, яна мае добрыя перспектывы для развіцця і дапрацоўкі. Такім чынам галоўным вынікам гэтай працы з’яўляецца ўбудаваная база дадзеных, якую ўжо можна выкарыстоўваць як пастаўшчыка існасцяў сістэмам у шаблоне Існасць-Кампанент-Сістэма ў межах эксперыментальных праектаў.

### Бібліяграфічныя спасылкі


1. Gregory J. Game Engine Architecture. Boca Raton : CRC Press, 2018.
2. Nystrom R. Game Programming Patterns. Seattle : Genever Benning, 2014.
3. Harrington J. L. Relational Database Design and Implementation: Clearly Explained. Waltham : Morgan Kayfmann, 2016.
4. Landenmaki T. Relational Database Index Design and the Optimizers. New Jersey : Wiley-Interscience, 2005.