Универзитет у Београду

Електротехнички факултет



**Имплементација видео-игре Mummy Maze у развојном оквиру Godot**Дипломски рад

|  |  |
| --- | --- |
| Ментор: | Кандидат: |
| проф. др Марко Мишић | Павле Продановић 2020/0249 |

Београд, септембар 2024. године

Садржај

[Садржај 2](#_Toc178033740)

[1. Увод 3](#_Toc178033741)

[2. Опис игре са постојећим решењима 5](#_Toc178033742)

[2.1. Mummy Maze 5](#_Toc178033743)

[2.2. Пример сличних решења 6](#_Toc178033744)

[3. Коришћени концепти и технологије 8](#_Toc178033745)

[3.1. Entity Component System 8](#_Toc178033746)

[3.1.1. Мане наслеђивања 8](#_Toc178033747)

[3.2. ECS концепти 9](#_Toc178033748)

[3.2.1. Ентитет 10](#_Toc178033749)

[3.2.2. Компонента 10](#_Toc178033750)

[4. Корисничко упутство 11](#_Toc178033751)

[4.1. Поставка игре 11](#_Toc178033752)

[4.2. Играч 12](#_Toc178033753)

[4.3. Мумија 12](#_Toc178033754)

[4.3.1. Бела мумија 12](#_Toc178033755)

[4.3.2. Црвена мумија 14](#_Toc178033756)

[4.4. Кориснички интерфејс 16](#_Toc178033757)

[Литература 18](#_Toc178033758)

[Списак скраћеница 19](#_Toc178033759)

[Списак слика 20](#_Toc178033760)

[Списак табела 21](#_Toc178033761)

1. Увод

Развој рачунара почевши од средине 20. века, довео је до развоја једне од најутицанијих области модерне индустрије забаве – видео-игара. Све је започело од најранијих пројеката, као што је *Tennis for Two* направљен као симулатор тениса на осцилоскопу 1958. године [1]. Убрзо након тога, појављује се *Pong* 1972. године, као прва комерцијално успешна игра [2]. У наредним деценијама област видео-игара доживљава процват, развојем кућних конзола као што је *Atari 2600*. Сама доступност и увећане могућности рачунара довели су до експоненцијалног раста популарности и комплексности видео-игара.

Како су игре постајале популарније и комплексније, тако су и алати за прављење игара постајали софистициранији и приступачнији. Данас су технологије за развој игара толико напредовале да омогућавају појединцима и мањим тимовима да креирају игре које могу конкурисати производима великих студија и освојити престижне награде.

Циљ овог рада јесте развој игре налик на некада популарну видео-игру *Mummy Maze*, али користећи модерне алате и приступе у развоју игара. До пре неколико деценија, алати коришћени за прављење једне игре, углавном нису могли бити поново искоришћени за прављење друге игре. Чак није било могуће користити код за развијање исте игре на више платформи. Сваки студио који развија видео-игре морао је да има скуп својих интерних алата и библиотека, сакупљених заједно у развојни оквир који се зове *Game Engine* [3].

Данас су ови развојни оквири комерцијализовани, па су самим тим и приступачнији јер не захтевају огромна финансијска и временска улагања. *Unreal Engine* и *Unity Engine* су примери развојних оквира који су годинама у назад дефакто стандард у развоју игара, за тимове који не желе да развијају свој софтвер.

Међутим, овај рад ће се фокусирати на *Godot Engine* (у наставку Godot), развојни оквир који је у последњих неколико година доживео велики напредак. Godot је веома популаран у развоју 2D игара, али се све више користи и за 3D развој. Његова популарност произлази из мултиплатформске подршке за десктоп, мобилне и веб платформе, Такође произилази из чињенице да је отвореног кода (*open-source)* [4], односно потпуно бесплатан за развој и објављивање видео-игара, са доступним изворним кодом.

Друго поглавље бавиће се описом игре са постојећим имплементацијама које су направљене од изласка саме игре.

Треће поглавље бавиће се технологијама и концептима потребним за имплементацију игре *Mummy Maze.*

Четврто поглавље служиће као корисничко упутство, које ће детаљно описати игру из угла корисника, а такође и објаснити алгоритме по којима функционише кретање мумија у игри.

1. Опис игре са постојећим решењима

У овом одељку најпре ће бити кратко описана оригинална игра *Mummy Maze* која ће бити имплементирана. Касније ће бити дат пример сличних игара истог концепта или жанра.

* 1. Mummy Maze

Десктоп игра *Mummy Maze* развијенa је 2002. године од стране студија *PopCap Games* (који је такође развио игре као *Plants vs Zombies*) [5]. *Mummy Maze* је логичка игра на потезе (*Puzzle Turn-based game*). Њен главни циљ је да играч, у улози истраживача, успешно побегне из лавиринта, притом избегавајући мумију која му се, по унапред предвидљивом обрасцу, у сваком потезу приближава. Ако мумија сустигне играча, играч је поражен и мора да прелази ниво испочетка.

A screenshot of a video game

Description automatically generated

Слика 2.1.1. Оригинални изглед игре *mummy maze* [6]

Концепт игре је инспирисан је радом Роберта Абота, који је 1990. у својој књизи *Mad Mazes* [7]у једном од поглавља предложио игру *Theseus and the Minotaur,* по узору на причу из грчке митологије*.*

Осим мумија, постоје разни други противници и препреке на које спречавају играча да побегне из лавиринта. Међутим, овај рад ће се ограничити на два типа противника: белу мумију и црвену мумију.

* 1. Пример сличних решења

Током година од изласка игре, неколико пута је покушано прављење игре која ће реплицирати оригиналну игру *Mummy Maze*. Од њих може се издвојити донекле популарна мобилна верзија, објављена од стране тима *NDP Studio* [8]. Изглед мобилне игре можемо видети на слици 2.2.1.

A screenshot of a video game

Description automatically generated

Слика 2.2.1. Мобилна верзија игре *Mummy Maze [8]*

Такође, треба издвојити и имплементацију концепта од ког је *Mummy Maze* инспирисан. То је веб апликација *Theseus and the Minotaur*, коју је направио *github* налог *estivalet [9].* Изглед веб апликације можемо видети на слици 2.2.2.

Обе игре, иако сличне по графичком изгледу, имају исти једноставни концепт лавиринта из ког играч мора побећи, док га непријатељ прогони. Оно што овакве игре разликује је што постоји специфичан тип стратегије потребан за прелажење нивоа. Овакве игре могу бити компликоване за просечног играча, ако није унапред упознат са логиком кретања непријатеља. Број могућих комбинација потеза који играч може одиграти да пређе ниво је мали. Због тога је потребно размишљати неколико потеза унапред о кретању како играча, тако и непријатеља.

*A screenshot of a game

Description automatically generated*

Слика 2.2.2. Веб апликација *Theseus and the Minotaur [9]*

1. Коришћени концепти и технологије

У овом поглављу ....

* 1. Entity Component System

*Entity Component System* (у наставку ECS), је пројектни узорак (енг. *design pattern)* по којем делимично или потпуно ради скоро свака модерна игра. Овај пројектни узорак настоји да наслеђивање замени композицијом, да би се добио флексибилнији и скалабилнији систем [10].

* + 1. Мане наслеђивања

Концепт наслеђивања у објектно-оријентисаном програмирању може да буде изузетно неефикасан и прекомпликован у развоју игара. Језици за развијање игара, као што су C# и Godot скрипт језик *GDScript,* уопште не подржавају вишеструко наслеђивање. Вишеструко наслеђивање се избегава чак и у језицима који подржавају вишеструко наслеђивање, првенствено *C++* [11]*.*

Под претпоставком да користимо само једноструко наслеђивање, на примеру ћемо видети предности које ECS има над наслеђивањем. Као илустративни пример, користиће се игра која се имплементира у овом раду, *Mummy Maze.* Претпоставимо да у нашој верзији постоје следећи елементи:

* Зид – непомерљива препрека постављена на мапи
* Кључ – интерактивни објекат који играч мора да покупи да пређе ниво
* Кутија – интерактивни објекат који играч може да помера по мапи
* Играч – карактер ког контролише корисник
* Мумија – непријатељ који јури играча по мапи

На слици 3.1.1.2. можемо видети пример како би се описани елементи могли имплементирати. Сваки квадрат представља једну класу. „Интерактивни“, „ИнтерактивниПомерајући“ и „Покретљив“ представљају функционалности. Оне могу бити имплементиране као класе или као интерфејси, у зависности од језика. Плавом бојом обојени су претходно наведени елементи.

Објекат који је наткласа свим осталим класама, такође садржи информацију о позицији елемента на мапи. С обзиром да је игра направљена тако да је подељена као шаховска табла, довољно је да објекат садржи редни број реда и колоне где се објекат налази.

У овом примеру већ видимо како за најелементарније функционалности, морамо да направимо сложен систем са неколико нивоа наслеђивања. Шта би се десило када бисмо желели да играч може да буде анимиран или да садржи звучне ефекте који ће се извршити током ходања? Могли бисмо да направимо класу ПокретљивАнимиранСаЗвуком која би садржала све три ове функционалности. Таква класа би била потпуно неорганизована и немогућа за одржавање.

A diagram of a company

Description automatically generated

Слика 3.1.1.2. Дијаграм користећи концепт наслеђивања

* 1. ECS концепти

Пошто традиционално наслеђивање не даје добре резултате, потребно је користити архитектуру која ће бити флексибилна за потребе прављења видео-игара. Због тога ћемо увести концепте које користи ECS:

1. Ентитет
2. Компонента
3. Систем
   * 1. Ентитет

Ентитет представља инстанцу једног објекта у игри. Ако пратимо пример из претходног поглавља, то може бити једна кутија, играч, мумија или кључ. Ентитету није битно која је његова намена. Намена ентитета јесте да служи као омотач за низ одређених компоненти.

Оно што ентитет садржи у себи јесте уникатни идентификатор преко којег можемо да му приступимо и низ идентификатора ка компонентама које он садржи.

* + 1. Компонента

Компонента представља класу која садржи податке одређене функционалности, али не садржи саму логику за њено извршавање. Компоненте се додају ентитетима, а онда можемо мењати податке компоненти ентитета, и тиме прилагодимо тај ентитет нашим потребама. Основни разлог овакве организације јесте раздвајање података и логике извршавања. Овакав приступ је посебно важан у развоју видео-игара јер омогућава ефикасно чување великог броја компоненти у низовима у меморији. На тај начин се може управљати већим бројем ентитета, уз ефикасно коришћење рачунарских ресурса.

На слици 3.2.2.1. можемо видети неке примере компоненти. Компоненте наведене овде користе се за функционалности поменуте у одељку са наслеђивањем. Наведени чланови ових класа су само илустративног карактера, а сама компонента може садржати и сложеније типове података.

A screenshot of a computer

Description automatically generated

Слика 3.2.2.1. Примери компоненти

* + 1. Систем

Систем служи да за једну компоненту или један скуп компоненти у сваком фрејму одради неку акцију, која је дефинисана у зависноти од потребе тих компоненти. То може бити исцртавање, освежавање корисничког интерфејса и тако даље. Систем ради ове акције за све компоненте које то затраже тако што итерира кроз све њих. То је још једна од предности због којих се компоненте чувају заједно у низу.

На пример желимо да имамо систем који ће извршити кретање играча за једно по дефинисаном смеру. Тај систем треба да има информације о тренутној позицији играча, и о жељеном правцу кретања и брзини кретања. Онда у сваком фрејму може да промени положај играча за неки померај, док играч не стигне до жељене локације.

Када су сви основни ECS појмови дефинисани, можемо погледати како изгледа наш систем из примера користећи овај пројектни узорак. На слици 3.2.3.1. имамо инстанце 4 ентитета за све типове елемената игре, осим за зид. Зид неће имати неке од компонената које смо раније набројали, а такође, пошто је статичан објекат, он се реализује на другачији начин.

На овом упрошћеном дијаграму можемо видети да су објекти и даље наслеђени из класе објекат. То је нормална ствар, зато што ECS не забрањује извођење, и оно треба да постоји кад за то има смисла. У овом случају, сви објекти игре се налазе на табели, па можемо имати наткласу која ће садржати компоненту о позицији ентитета.

Међутим, овај дијаграм не представља класе за кутију, играча, мумију и кључ. Сви ентитети су део једне класе која се зове ентитет, наслеђене из класе објекат. Након што су инстанце направљене, у сваку од њих су додате референце ка одговарајућим компонентама. Да не бисмо морали да сваки пут додајемо празан ентитет па онда компоненте у њега, можемо да направимо *prefab. Prefab* се користи да дефинише један заједнички тип ентитета који ће се понављати у игри. Свака његова инстанца ће при прављењу добити исте вредности у компонентама. Када се измени *prefab,* измениће се и све његове инстанце и тиме ћемо знатно смањити потребу за ручним променама. Такође је могуће да одрадимо промене локално, само за једну инстанцу.

A diagram of a company

Description automatically generated

Слика 3.2.3.1. Дијаграм са 4 ентитета и њиховим компонентама

Сада можемо да видимо колико је заиста помогло коришћење овакве архитектуре. У играча ћемо додати компоненту за анимацију и звук. Такође можемо да додамо компоненту за звук и у кључ, да бисмо пустили звучни ефекат када играч интерагује са кључем. Измењени дијаграм може се видети на слици 3.2.3.2. За додавање 3 нове компоненте је потребно додати 3 референце у одговарајуће ентитете ка тој компоненти. Јасно је да би иста оваква модификација путем наслеђивања била много комплекснија.

A diagram of a company

Description automatically generated

Слика 3.2.3.2. Дијаграм са 4 ентитета и новододаним компонентама

1. Корисничко упутство

У овом поглављу биће детаљно објашњена видео-игра из угла корисника. Прво ће бити дат кратак опис игре и њених елемената. Након тога ћемо објаснити како функционише истраживач ког корисник контролише. Након тога ћемо објаснити мумију и њен алгоритам кретања. За крај ће бити приказан преглед корисничког интерфејса.

* 1. Поставка игре

Игра је подељена на нивое, где сваки ниво представља један спрат лавиринта. Сваки спрат има облик квадрата и подељен на мање квадрате, налик на шаховску таблу. Укупна димензија табле је 6x6. Спрат се састоји од зидова и степеница. Зидови су статичне препреке кроз које не могу проћи ни мумија, ни играч. Степенице су циљ нивоа до којих играч мора доћи да би успешно прешао ниво. Сваки ниво почиње тако што играч и мумија заузимају своје позиције на предодређеним квадратима.

A video game with a game of a game

Description automatically generated with medium confidence

Слика 2.1.1. Први ниво игре

Играч и мумија имају право да се крећу у 4 смера: горе, лево, доле или десно, померајући се по квадратима, ако их у томе зид не спречава. Први на потезу је увек играч, који се креће командама задатим преко тастатуре. Након играчевог потеза, на потезу је мумија, која се креће по предодређеном алгоритму, у зависности од типа мумије.

* 1. Играч

Играч у свом потезу има право на једно или ниједно померање по лавиринту. Свака могућа акција играча мапирана је на тастере на тастатури, као што се може видети у табели 2.2.1.1. Акција покушана у смеру у ком није дозвољено кретање или током извршавања претходне акције неће бити узимана у обзир.

Табела 2.2.1.1. Акције и њихово мапирање на тастатури

|  |  |
| --- | --- |
| **Акција** | **Тастери на тастатури** |
| Кретање на горе | "W", "↑" |
| Кретање у десно | "A", "←" |
| Кретање на доле | "S", "↓" |
| Кретање у лево | "D", "→" |
| Прескакање потеза | "Space" |

* 1. Мумија

Мумија у свом потезу има право на максимално два кретања по мапи. Смер кретања директно зависи од унапред дефинисаног алгоритма за ту мумију. Алгоритам за сваку мумију је такав да она настоји да се приближи играчу. Мумија никад неће изабрати кретање тако да се не приближава играчу по некој оси (хоризонталној или вертикалној). Постоје два типа мумија:

* Бела мумија
* Црвена мумија
  + 1. Бела мумија

Бела мумија се појављује у игри од 1. до 4. нивоа. Креће се тако што приоритизује кретање по хоризонталној оси, па тек онда по вертикалној, ако се по хоризонталној оси не може приближити играчу. Алгоритам кретања беле мумије је приказан на слици 2.3.1.1.

A diagram of a flowchart

Description automatically generated

Слика 2.3.1.1. Алгоритам кретања беле мумије

Пример алгоритма кретања можемо видети на слици 2.3.1.2. У свом првом делу потеза мумија ће ићи на десну страну, јер јој је приоритет да се приближи играчу хоризонтално. У другом делу потеза ће исто одлучити да се креће хоризонтално, иако јој то "затвара" директан пут ка играчу вертикално.

A screenshot of a video game

Description automatically generated

Слика 2.3.1.2. Пример кретања беле мумије

* + 1. Црвена мумија

Црвена мумија се појављује у игри на 5. и 6. нивоу. Креће се тако што приоритизује кретање по вертикалној оси, па тек онда по хоризонталној, ако се по вертикалној оси не може приближити играчу. Алгоритам кретања беле мумије је приказан на слици 2.3.2.1.

A diagram of a flowchart

Description automatically generated

Слика 2.3.2.1. Алгоритам кретања црвене мумије

Пример кретања можемо видети на слици 2.3.2.2. У првом делу потеза мумија се креће вертикално, као по алгоритму, да би се приближила играчу. У другом делу потеза мумија не може да се приближи играчу ни вертикално ни хоризонтално, па због тога прескаче свој потез.

Занимљиво је запазити да би у овој истој ситуацији бела мумија ишла на десну страну па на горе и тиме би поразила играча. Због тога је важно знати разлику између кретања црвене и беле мумије и према томе правити стратегију кретања по мапи.

A screenshot of a video game

Description automatically generated

Слика 2.3.2.2 Пример кретања црвене мумије

* 1. Кориснички интерфејс

Кориснички интерфејс приказан је током целог трајање игре, на панелу са леве стране. У наставску следе објашњења свих елемената корисничког интерфејса, приказаних на слици 2.4.1.

1. "Vreme": време проведено на тренутном нивоу. Ресетује се само кад кликнемо дугме "Resetuj Nivo", ако мумија порази играча наставља да тече неометано.
2. "Rekord": рекордно време за које је пређен тренутни ниво. Ако ниво још увек није пређен, уместо времена стојаће ознака "/". Сва рекордна времена чувају се на корисничкој машини док год корисник не ресетује своје сачуване податке.
3. "Resetuj Nivo": ресетује тренутни ниво тако што тренутно време врати на 0 s, такође враћа мумију и играча у своје почетне положаје.
4. "Izaberi Nivo": скуп дугмади са којима можемо прелазити са једног пређеног нивоа на други. Дугме за одређени ниво ће бити закључано ако још увек нисмо успели да стигнемо до тог нивоа.
5. "Napusti Igru": чува све корисничке податке и напушта игру.

A video game with a game screen

Description automatically generated with medium confidence

Слика 2.4.1. Прозор игре са нивоом и корисничим интерфејсом

Постоји још једно дугме које није наведено у претходној листи. Дугме "Resetuj Sve" откључава се када пређемо свих 6 нивоа игре. Притиском на дугме ресетујемо све сачуване корисничке податке. То укључује сва рекордна времена и информацију о пређеним нивоима, чиме се поново враћамо на ниво 1.

Литература

1. <https://www.bnl.gov/about/history/firstvideo.php>  
   приступљено 23.09.2024.
2. [*https://en.wikipedia.org/wiki/Pong*](https://en.wikipedia.org/wiki/Pong)приступљено 23.09.2024.
3. <https://www.gameopedia.com/game-engines-all-you-need-to-know-about/>  
   приступљено 23.09.2024.
4. <https://github.com/godotengine/godot>  
   приступљено 23.09.2024.
5. [*https://en.wikipedia.org/wiki/List\_of\_PopCap\_games*](https://en.wikipedia.org/wiki/List_of_PopCap_games)приступљено 23.09.2024.
6. [*https://www.youtube.com/watch?v=qa8H7386Ddc&*](https://www.youtube.com/watch?v=qa8H7386Ddc&)приступљено 23.09.2024.
7. Robert Abbott, *Mad Mazes: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People,* 1990.
8. [*https://play.google.com/store/apps/details?id=com.resinviet.mummy&hl=en*](https://play.google.com/store/apps/details?id=com.resinviet.mummy&hl=en)приступљено 23.09.2024.
9. <https://estivalet.github.io/theseus-minotaur/>,  
   приступљено 23.09.2024.
10. <https://www.umlboard.com/design-patterns/entity-component-system.html>  
    приступљено 24.09.2024.
11. <https://cs.smu.ca/~porter/csc/common_341_342/notes/oop_multiple_inheritance.html>  
    приступљено 23.09.2024.

Списак скраћеница

2D *– two-dimensional*

3D – *three-dimensional*

ЕCS – *entity component system*

Списак слика

[Слика 2.1.1. Оригинални изглед игре *mummy maze* [6] 5](#_Toc178033691)

[Слика 2.2.1. Мобилна верзија игре *Mummy Maze* 6](#_Toc178033692)

[Слика 2.2.2. Веб апликација *Theseus and the Minotaur* 7](#_Toc178033693)

[Слика 3.1.1.2. Класни дијаграм користећи концепт наслеђивања 9](#_Toc178033694)

[Слика 3.2.2.1. Примери компоненти 10](#_Toc178033695)

[Слика 2.1.1. Први ниво игре 11](#_Toc178033696)

[Слика 2.3.1.1. Алгоритам кретања беле мумије 13](#_Toc178033697)

[Слика 2.3.1.2. Пример кретања беле мумије 14](#_Toc178033698)

[Слика 2.3.2.1. Алгоритам кретања црвене мумије 15](#_Toc178033699)

[Слика 2.3.2.2 Пример кретања црвене мумије 16](#_Toc178033700)

[Слика 2.4.1. Прозор игре са нивоом и корисничим интерфејсом 17](#_Toc178033701)

Списак табела

[Табела 2.2.1.1. Акције и њихово мапирање на тастатури 6](#_Toc177925578)