Висока школа електротехнике и рачунарства струковних студија



**Видео игра: Precision Trainer**

Предмет: Пројектовање рачунарских игара

Лука Челебић - ИС - 27/18

Јануар 2020.

Садржај

1. [Увод 3](#_TOC_250012)
   1. [Основни подаци 3](#_TOC_250011)
2. [Циљеви, правила и структура игре 4](#_TOC_250010)
3. [Ток игре (Gameplay), дизајн и механика игре 5](#_TOC_250009)
   1. [Gameplay контроле 7](#_TOC_250008)
   2. [Кориснички интерфејс 8](#_TOC_250007)
   3. [Камера и осветљење 9](#_TOC_250006)
4. [Играч 10](#_TOC_250005)
5. [Додатни ефекти у игри и звук 10](#_TOC_250004)
   1. [URP Shader graph 10](#_TOC_250003)
   2. [Звук 11](#_TOC_250002)
6. [Закључак 12](#_TOC_250001)
7. [Референце 12](#_TOC_250000)

# Увод

Желели смо да направимо видео игру која има за циљ да помогне играчу у вежбању прецизности.

Главне две опције које смо имали су те да ли игру направити у **2D** ili **3D** окружењу. 2D окружење би било једноставније приказати и створити али са обриром на то да је главно циљано тржиште које би играло ову игру то које жели да побољша своју прецизност због неких других видео игара које игра које су све у 3D-у ми смо одлучили за опцију која би верније то окружење приказала.

## 1.1. Основни подаци

Наслов игре: **Precision Trainer (Тренер Прецизности)**

Платформа: **Microsoft Windows**

Жанр: **First Person Shooter**, **Simulation**, **Action**, **3D**

Циљна група: **FPS Гејмери**

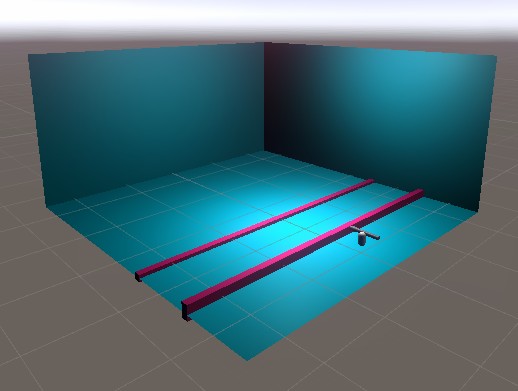
Главни циљ је побољшање и тестирање прецизности играча које се извршава прикупљањем података о перформансама играча у току игре и на крају приказује играчу:

* колико је мета погођено
* колико је мета омашено
* који је проценат прецизности
* колико је секунди протекло од почетка игре

# Циљеви, правила и структура игре

У игри постоји само једна просторија у којој се налази играч у којој се све одвија.

Прост приказ мапе игре у “**Scene**” панелу у Unity едитору пројекта:



Мапа је подељена на **3 дела**.

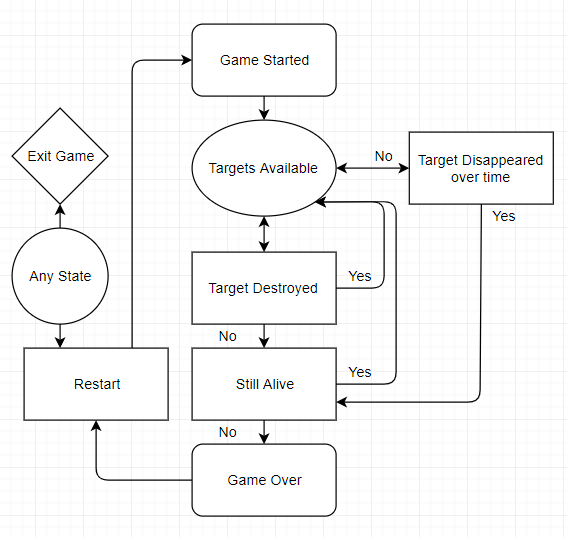
У **првом** делу десно се налази играч и он се може кретати само хоризонтално свуда све до прве љубичасте ограде.

**Други** део, тј. део између две ограде је празан и у њему се ништа не дешава током игре, намењен је као лимитиран простор у коме мете не треба да се стварају.

И **трећи** део скроз лево на слици у којем се стварају мете које играч треба да упуца тј. кликне. Те мете се стварају на било којим **X, Y, и Z** координатама тог простора унутар собе.

# Ток игре (Gameplay), дизајн и механика игре

Дијаграм тока игре:



Игра садржи **2** мода играња: **“Incremental” и “Instant”.**

У **“Incremental”** моду стварају се мете тј. сфере у игри које играч треба да уништи у зависности са временом. Временом се убрзава стварање мета докле год играч може да их испрати и уништи кликом левог тастера миша.

У другом моду игре **“Instant”** се стварају 3 објекта и након што играч уништи било који одмах се ствара нови докле год је број њих у игри мањи од боја 3.

Овим модом сви могу да иду својом максималном брзином пуцања од самог почетка без чекања да се игра убрза.

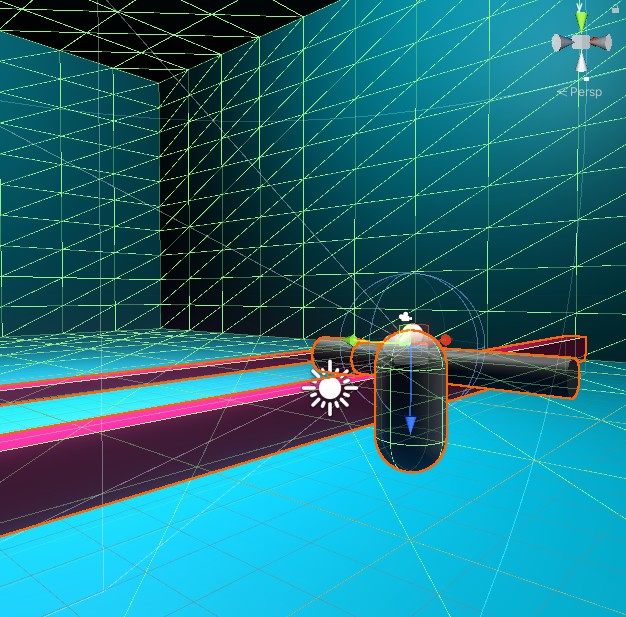
Играч има **3 живота**. Како би убрзали начин на који играчи могу да упале, ресетују и играју игру како би могли да рецимо вежбају своје циљање између играња других видео игара контроле игре се извршавају преко пречица.

Одмах након покретања игре креће стварање мета за играча.

Играч у било ком тренутку кликом на “**R**” може **рестартовати** игру и почети од почетка, у било ком тренутку двокликом на “**ESC**” може **угасити** игру и у било ком тренутку кликом на “**Q**” мења мод игре са “**Incremental”** na “**Instant”** i obrnuto.

Једини објекат који се помера у игри је сам играч и сви елементи који су њему додељени као сто су **камера** и **пушка** играча.

На слици испод су приказани сви колајдери мапе у видео игри:



Поред ових постоји и колајдер у облику **сфере** који стоји на свакој **мети**. Сви ови колајдери се користе за детекцију метака при пуцању.

Детекција колизије се извршава путем **Raycast** методе, тј. из центра **ViewPort**-**a** (видљивог дела игре путем камере) се шаље ласер на сцену игре.

Уколико тај ласер дође у судар са колајдером сфере онда се мета уништава и играчу се повећава број погођених мета. У другом случају ће ласер доћи у судар са зидовима или две постављене ограде на сцени при чему ће се само памтити то да је играч пукао али не и погодио.

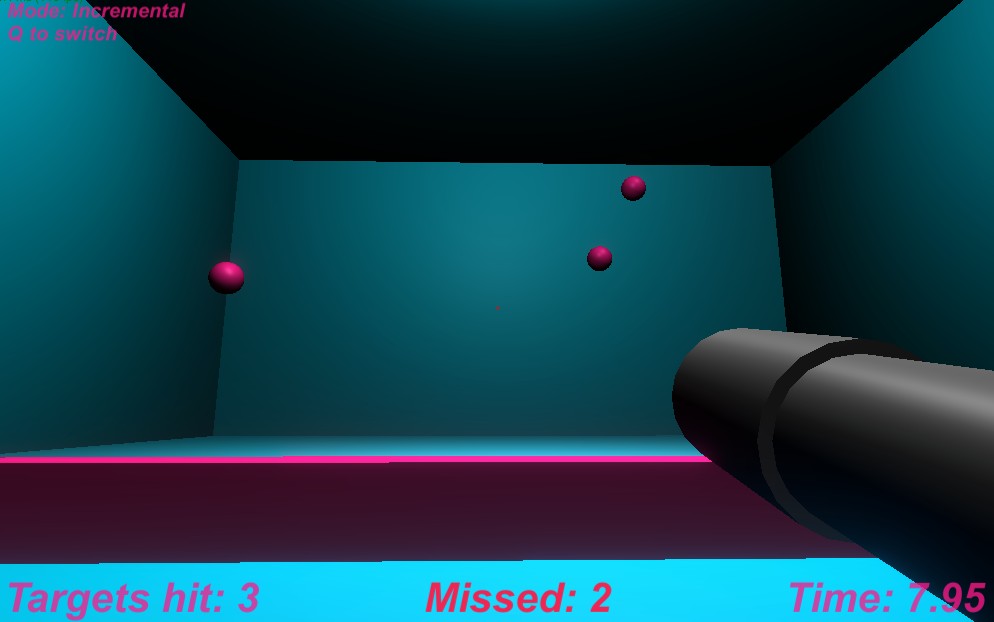
## Gameplay контроле

|  |  |  |
| --- | --- | --- |
| Акција | Тастер | Миш |
| Кретање лево | A |  |
| Кретање десно | D |  |
| Кретање напред | W |  |
| Кретање назад | S |  |
| Ротација играча |  | Померање миша |
| Пуцање |  | Леви тастер |
| Промена мода игре | Q |  |
| Рестартовање игре | R |  |
| Гашење игре | ESC \* 2 |  |

## Кориснички интерфејс

Као и изнад напоменуто коришћене су пречице за навигацију како играч не би морао да се фокусира на кликтање и гледање било чега осим мета за циљање па је зато коришћен је минималан дизајн при креацији интерфејса.

Ово је пример изгледа интерфејса при игрању игре:



А ово је пример интерфејса на крају игре:



Горе приказани елементи су сви текстуалног облика без позадинског панела који се преко “**GameManager”** скрипте освежавају у сваком фрејму игре.

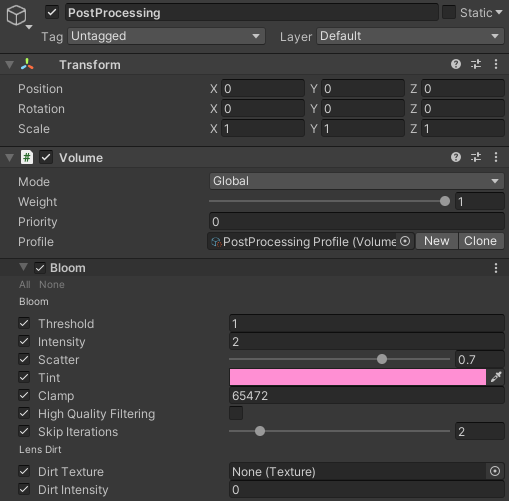
## Камера и осветљење

За осветљење је коришћено **дирекционо** светло да би подједнако могло да се осветли све по жељи а затим и једно **тачкасто** светло на плафону собе.

Друга битна ствар око осветљења и приказа сцене преко камере је то што је коришћен **Post Processing** који је недавно при почетку 2020. године додат у **URP** (**Universal Render Pipeline**) framework-у за рендеровање објеката на сцени.

URP је креиран и разликује се од **Built-in** почетног рендера по томе што нуди много већи број начина за едитовање и оптимизацију свега што видимо на сцени, већином око осветљења.

На слици испод се налази пример искоришћеног **Bloom** ефекта који је додат као део **Volume** компоненте празном game објекту на сцени:



# Играч

На главном играчу се налази скрипта “**PlayerMovement**” преко које се креће. Са обзиром да нема потребе за скакањем у игри у едитору Unity-ја што се тиче померања по мапи можемо мењати и видети само његову брзину.

Главна камера се налази на играчу како би пратила његове помераје и могла из те перспективе и да пуца на мете.

На главној камери играча од опција које утичу нама директно на gameplay се налази само опција за мењање сензитивности миша.

Играч има 3 живота, тј. могућности да у “**Incremental**” моду не успе на време да уништи мету пре него што се игра заврши.

# Додатни ефекти у игри и звук

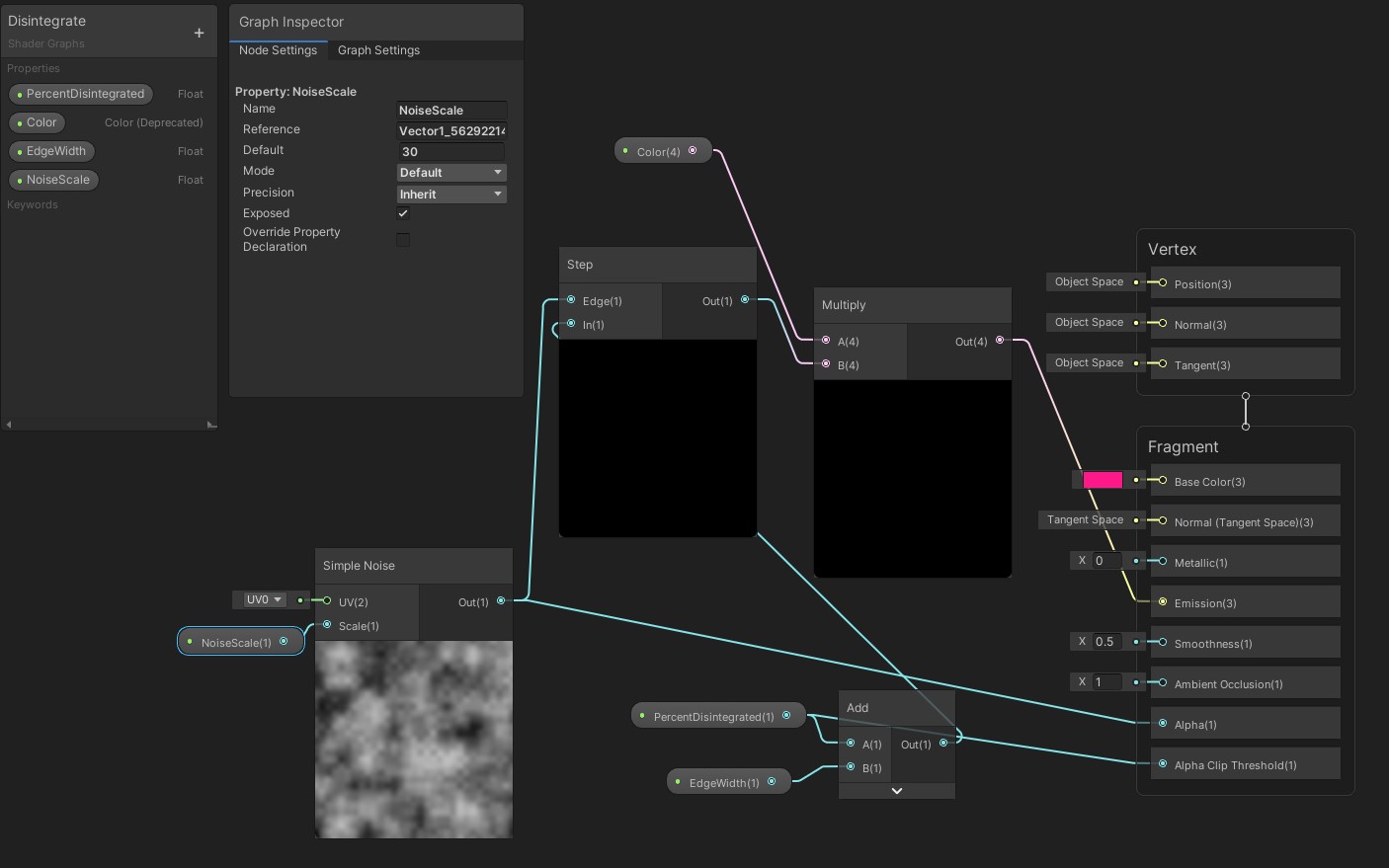
Још једна од битних ствари коју смо искористили у игри је Shader Graph функционалност **URP-а**. У њему конектовањем елемената визуелно можемо скриптовати и креирати **shader-е** без мануелног познавања и кодовања синтаксе за њихово прављење у Unity C#.

## URP Shader graph

Овако успорено изгледа стварање нових мета тј. сфера у видео игри.



Иглед **node** система повезивања који је коришћен како би дошли до приказаног ефекта изнад који је постављен на сваку од мета у игри.



Уместо знања синтаксе за рецимо креацију насумичних вредности ми десним кликом на празну површину можемо креирати **Simple Noise** који је овде искоришћен како би стварање сваке лопте било мало другачије.

## Звук

Постоје две групе звукова у игри, звук за пуцање и позадински звук.

Звук за пуцање је преузет са online сајта за бесплатне asset-е за видео игре

- **opengameart.org**

Преузето је 5 различитих звукова ласера од којих се један насумично селектује при **пуцању** и пушта играчу.

Затим је ту и константно понављајућа тиха **позадинска** музика која је аутоматски генерисана online на сајту: **tones.wolfram.com**

# Закључак

Овом релативно простом симулацијом циљања и пуцања смо прешли основе кретања, погледа и пуцања са камере играча, инстанцирање мета, контрола стања игре и друге основе али смо такође обухватили и новију технологију која је додата у **URP** рендеру Unity-ја преко коришћења **Shader Graph-a** и **Bloom** ефекта **Post Processing-а** додатим у игру.

Сви ови делови заједно на крају чине просту али корисну видео игру која због својих скалабилних начина играња које се базирају на самој брзини пуцања играча као што је наш “**Instant**” мод игре омогућава да ова игра буде изазов играчима свих вештина и помогне им у ономе за шта је намењена, побољшању прецизности.

# Референце

Unity URP:

[https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@10.2/](https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal%4010.2/manual/universalrp-builtin-feature-comparison.html) [manual/universalrp-builtin-feature-comparison.html](https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal%4010.2/manual/universalrp-builtin-feature-comparison.html)

WolframTones Music: <https://tones.wolfram.com/generate> Free Game Art: <https://opengameart.org/>

Youtube Unity Tutorials: <https://www.youtube.com/c/Brackeys/videos> Graph Creator: <https://www.diagrameditor.com/>