

СИМУЛАЦИЈА КРЕТАЊА ФУДБАЛСКИХ ИГРАЧА СА АСПЕКТА ДЕФАНЗИВНИХ ТАКТИКА**SIMULATION OF FOOTBALL PLAYER MOVEMENTS FROM AN DEFENSIVE STANDPOINT**

Никола Скробоња, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Виртуални свет постаје све више део реалног света и живота, користи се у едукацији, спорту и забави. У овом раду је описан процес креирања симулације кретања играча на фудбалском терену. Описан је читав процес креирања овакве симулацију у Unity-ју и приказане програмске технике коришћене за имплементацију кретања.*

Кључне речи: *Симулација, Развој игара, Фудбалске тактике*

Abstract – *The virtual world is becoming more and more part of the real world and peoples lives, it's used in education, sport and entertainment. In this paper is described the process of creating a simulation of player movements on the field. The whole process of creating this kind of simulation in Unity is described, as well as the programming techniques used to implement this.*

Keywords: *Simulation, game development, Football tactics*

1. УВОД

Фудбал представља најпопуларнију игру на свету где се из дана у дан рађају нове тактике и нови начини играња игре како би се дошло до победе и титула.

Основе фудбала се уче на фудбалском терену са тренером. Међутим, често је потребно доста времена како би играч стекао осећај за преглед игре и сходно томе и кретао се на одговарајући начин према жељама тренера. Играчи се можда могу и не слагати са идејама које тренер има услед неразумевања идеје коју тренер има.

Рачунарске симулације су уопштено значајне јер омогућују поглед на ситуације које се могу десити у реалном животу и припремити човека на такве ситуације.

Симулација кретања фудбалских играча са аспекта дефанзиве има управо тај значај где тренер има прилику да објасни играчу како треба да се креће у одређеној ситуацији и која је сврха таквог кретања. Играч са друге стране се на овај начин припрема за такву ситуацију која је врло вероватна да се деси у реалности на утакмици.

НАПОМЕНА:

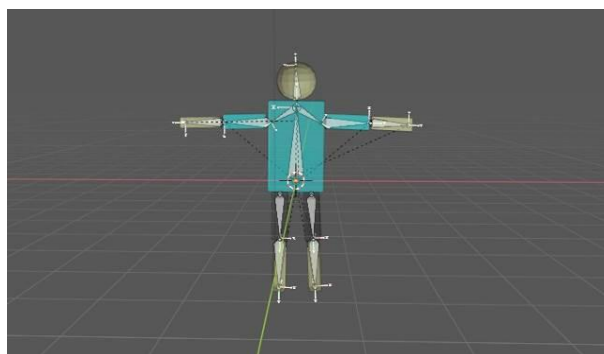
Овај рад проистекао је из мастер рада чији ментор је био проф. др Драган Иветић.

2. МОДЕЛОВАЊЕ

Први корак у креирању овакве апликације јесте креирање модела који су део сцене, као и анимирања покретних објеката сцене.

Најпре је креиран модел играча. Циљ овог модела је да се прикажу покрети потребни за кретање играча на терену, стога је креиран најједноставнији облик модела који ово подржава.

За моделовање играча и остатка сцене је коришћен Блендер, програм за 3Д моделовање.



Слика 1: Приказ модела са скелетом

Уз меш модел који је направљен од скалираних коцки и сфере, направљен је и скелет којим су креиране анимације играча. Креиране анимације су мировање (*Idle*), трчање и додавање лопте.

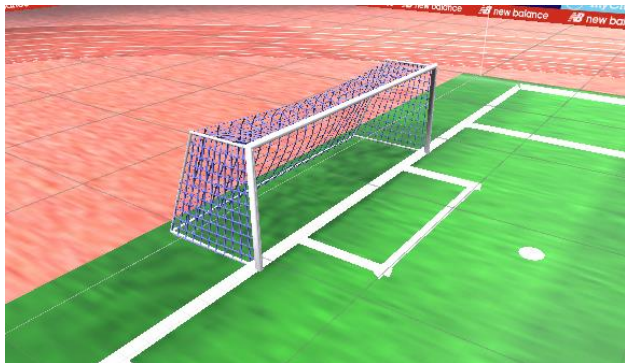


Слика 2: Приказ анимације

Скелет модела представља хијерархијски модел са кореном у кичми на основу којег су анимације креиране помоћу технике *Keyframing* у оквиру Блендера. Овај модел је заједно са анимацијама експортиран у *.fbx* фајл који је даље импортован у *Unity Game Enging*.

Остатак сцене представљају статични модели. То су модели гола, терена и окружења које представљају рекламе и трибине.

Пречке и стативе модела гола су моделоване помоћу модела цилиндара на који су залепљене одговарајуће текстуре. За мрежу је коришћен *Wireframe Modifier* (Блендер операција) уз примену физике платна како би се допринело реализму.



Слика 3: Приказ модела гола

Терен и подлога око терена су креирани помоћу *Terrain* објекта у оквиру Јунити погона који представља погодан начин за креирање одговарајућих подлога. С обзиром да је подлога у ДТГ апликацији једноставна и без неправилности, односно неравнина, могуће је било врло лако креирати подлогу и у Блендеру па тај модел онда учитати. *Terrain* објекти су погодни за неправилне терене и нуде алате за подешавање таквих подлога.

Линије на терену су креиране помоћу *LineRenderer* компоненте Јунити погона. Други начин за исцртавање линија по терену би био помоћу утврђених техника текстурисања, али с обзиром да Јунити нуди компоненту која олакшава овакво исцртавање, та компонента је и употребљена.

Око терена је креирана стаза за трчање која је моделована у Блендеру користећи *Circle* компоненту. Две ове компоненте креирају прстен чија раван је текстурирана тартаном са исцртаним линијама што је класична представа тркачке стазе.

Рекламе око терена су креиране помоћу равни на које су налепљене (текстурисањем) слике неких насумично одабраних реклама. Публика око терена је креирана помоћу текстура.



Слика 4: Приказ сцене са остатком модела

3. АНИМАЦИЈА МОДЕЛА ИГРАЧА

Основни вид кретања фудбалских играча је трчање. Свакако да постоје и други облици кретања у фудбалу, али за потребе ове апликације довољне су анимације мировања, трчања и додавања лопте. Анимације се раде помоћу *Keyframing* технике.

Идеја *Keyframing*-а је да се за одређене фрејмове (дискретне временске тренутке) креирају и сачувају позиције и ротације скелета (чиме управљамо и меш моделом). Линеарном интерполацијом трансформација између кључних фрејмова (*Key Frames*, енгл.) добија се привид покрета карактера.

У стању мировања, играч ради полускип, односно клати се лево-десно уз прављење ситних корака у месту. Оваква анимација не захтева велики број позиција и ротација скелета јер је анимација поприлично једноставна. Конкретно, у овој анимацији има три фрејма (*Frame*¹, енгл.) између којих се врши интерполација. *Keyframe*-ови су први, двадесети и четрдесети фрејм, где су први и четрдесети фрејм исти како би добили цикличну, односно понављајућу анимацију.



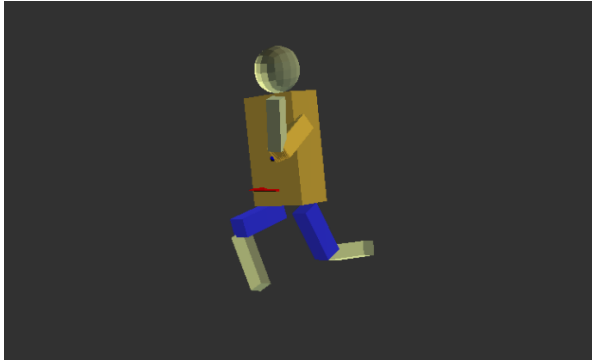
Слика 5: Први фрејм у анимацији мировања

Први фрејм је добијен применом одговарајућих трансформација над скелетом. Други фрејм је добијен инвертовањем позиција и ротација примењених у првом фрејму. Операција инвертовања је могућа јер су испоштоване конвенције задавања назива костију скелета и извршена је преоријентација локалних координатних система костију према глобалном координатном систему како би остварили одговарајуће понашање. За последњи фрејм анимације су копиране трансформације примењене у првом фрејму.



Слика 6: Петнаести фрејм

Трчање представља комплекснију анимацију те је потребан већи број кључних фрејмова како би се остварио задовољавајући резултат. Због тога су за анимацију трчања коришћене референтне позиције класичне анимације трчања. Креирано је дванаест кључних фрејмова који су допринели креирању анимације трчања играча. Како је потребно да ово буде циклична анимација, први и последњи фрејм су исти, слично као и код анимације мировања.



Слика 7: Приказ једног фрејма анимације трчања

За офанзивне играче у ДТГ апликацији је креирана анимација додавања лопте која је направљена по узору на додавања лопте у реалном свету од стране фудбалских играча и по узору на то како је урађено у неким од познатих игрица (*Fifa*, *PES*).



Слика 8: Приказ једног фрејма анимације додавања

Сви *Keyframe*-ови су креирани применом трансформација над скелетом модела. Због тога треба истаћи важност креирања доброг модела скелета како би се олакшао процес анимације. За креирање комплекснијег модела било би потребно креирати и скелет са *IK* костима како би се анимирао правилан покрет човека.

4. СЦЕНЕ У ДТГ АПЛИКАЦИЈИ

ДТГ апликација има две сцене, симулациону у којој се одиграва симулација кретања и сцена која садржи графички интерфејс одакле корисник покреће симулацију и подешава формацију и боју дресова.

У симулационој сцени су уз наведене моделе, постављена рефлекторска осветљења са свих страна стадиона и подешене су сенке играча како би се додало на реализму сцене. Уз то је додат и звучни ефекат публике који је адекватно компресомн који доприноси амбијенту и реализму.

Графички кориснички интерфејс у ДТГ апликацији се састоји од три дела која заједно чине целину у оквиру које се управља понуђеним подешавањима. Сваки од делова представља панел (*Panel*¹, *енгл.*). *MainMenu* представља основни мени који се први појављује кориснику након покретања апликације. Састоји се из позадине (која се провлачи кроз читав графички интерфејс), лабеле којом је исписан назив апликације и четири дугмета.

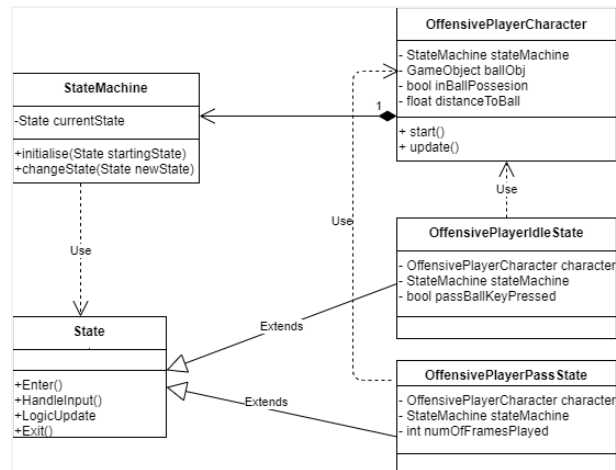
Притиском на дугме *Start* се покреће *gameplay* део апликације, односно започиње симулација под тренутним подешавањима, док се притиском на дугме *Quit* излази из апликације. Контроле *Kits* и *Formation* воде на *Kits Menu* и *Formation Menu*, респективно. У првом се може мењати боја гарнитуре дресова, док се у другом поставља формација играча.

5. ПРОГРАМСКЕ ТЕХНИКЕ (КОДИРАЊЕ КРЕТАЊА ИГРАЧА)

Кодирање кретања играча у фудбалској утакмици (уколико је циљ добра симулација) изискује уграђивање вештачке интелигенције која се све више користи у игрицама ради веће имерсивности и задржавања пажње корисника. Обично се уграђује више нивовкси *AI* у фудбалску симулацију јер играчи морају да сарађују између себе како би одбранили гол, нападали адекватно, испоштовали договоре везане за формацију и тактику.

Имплементација кретања играча остварује кроз коначне машине стања (*Finite State Machines*, *енгл.*) које су честа појава у свету развоја видео игара.

Играчи офанзивног тима, односно тима који има лопту у поседу, се не крећу по терену, већ се налазе на истом месту током читаве симулације и додају се између себе. Сваки играч заузима неки регион терена и према тим регионима се дефанзивни играчи оријентишу.



Слика 9: Class дијаграм машине стања

Притиском на тастер *P* се прелази у стање додавања (*Pass State*) где је најпре потребно одредити ком играчу да се дода лопта. Ово се остварује помоћу инстанце класе *GameManager* која се креира и иницијализује приликом покретања апликације и представља објекат који садржи податке који су потребни кроз скоро читаву апликацију. Она садржи

листе офанзивних (и дефанзивних) играча. Помоћу те листе се може пронаћи играч коме се додаје лопта. Тренутна имплементација је таква да се овај играч насумично бира. Играч који додаје се ротира ка смеру додавања и истовремено се лопта поставља испред њега (дајући утисак да је играч поставља) тако што се одреди вектор (нормализован) који дефинише смер у коме иде додавање. Овај вектор се рачуна одузимањем позиције играча којем се додаје и позиције одакле се додаје лопта.

Затим се параметар *passing* аниматора поставља на *True* чиме се покреће *PassingAnimation*. У *Update* методи се врши провера колико фрејмова је исцртано од тренутка активације стања и уколико је тај број прешао број фрејмова који садржи анимација, стање се враћа на *IdleState*. У *Exit* методи се ажурира позиција лопте која има глатко кретање и ротацију око своје осе приликом померања што даје бољи доживљај кориснику.



Слика 10: Офанзивни играч у моменту додавања

Са аспекта имплементације кретања дефанзивних играча није zgodно радити са коначним машинама стања јер сваки играч има посебно понашање. То значи да сваком играчу треба обезбедити индивидуално понашање које је условљено формацијом, позицијом лопте и позицијом играча на терену. Стога се сваком играчу додељује скрипта приликом креирања сцене. Скрипта која је додељена играчу зависи од позиције на терену и формације која је на снази. У оквиру скрипте се налази логика која врши проверу где се лопта налази и управља померањем играча. Све скрипте наслеђују класу *PlayerMovement* у оквиру које се унутар *Update* методе позива метода *CalculateBestPosition* која као параметре прихвата позицију играча (физичку позицију на терену), позицију лопте и димензије терена.

Свака формација има одговарајућу групу скрипти које се додељују дефанзивним играчима. Кроз ову имплементацију, у неку руку је и остварена машина стања, али не кроз класичан начин имплементације, већ користећи погодности које *Unity* нуди.

У овом случају се то односи на могућност доделе скрипте играчком објекту (*GameObject*) у току извршавања апликације. И код дефанзивних играча се мора водити рачуна о анимацијама, односно у ком тренутку играч трчи, у ком мирује. Дефанзивни играчи су такође оријентисани према лопти. Уз помоћ методе *MoveTowards* се обезбеђује постепено кретање играча уз *Running* анимацију.

Услов за прекидање анимације трчања јесте да играч стигне до позиције која је израчуната са *CalculateBestPosition* методом која се састоји од неколико *if-else* конструкција којима се проверава где се лопта налази на терену.



Слика 11: Приказ коначне сцене

6. ЗАКЉУЧАК

Резултат рада је апликација која представља интерактивну симулацију кретања дефанзивних играча на основу параметара које је корисник дефинисао. Корисник може ући и у тренерски режим који му омогућује да се слободно креће по терену и посматра симулацију.

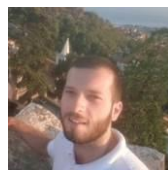
Главни проблем се јавља код додавања нових формација јер је потребно писати скрипте за сваку нову позицију која је везана за ту формацију, а не постоји ни у једној претходној. Ово се може решити уграђивањем неког алгоритма вештачке интелигенције.

У апликацију се може увести физика која би играчима и лопти дали масу и омогућили колизију, јачину додавања и шута. Кретања играча би додатно се морала кориговати са *Pathfinding* алгоритмом (нпр. *A* Search*) како би се избегла колизија између играча. Свакако би напредак био у смислу бољих модела играча и стадиона.

7. ЛИТЕРАТУРА

- [1] Mat Buckland, *AI Techniques for Game Programming*, 1st Edition, 2002.
- [2] Mat Buckland, *Programming Game AI By Example*, 1st Edition, 2004

Кратка биографија:



Никола Скробоња рођен је 1995. године у Новом Саду. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства – одбранио је 2021. године.