

## ПРИМЕНА ПРОГРАМА STENCIL У ПРЕДМЕТУ РАЧУНАРСКА ГРАФИКА

Перица Штрбац<sup>1</sup> Павле Штрбац<sup>2</sup> Вукман Кораћ<sup>3</sup> Милош Пејановић<sup>4</sup> Вера Петровић<sup>5</sup>

**Резиме:** У оквиру предмета Рачунарска графика на ВИШЕР учи се педесетак алгоритама који решавају проблеме из ове области. Проблем је што студенти пре тога нису имали предмет Увод у програмирање. Идеја је да се студенти који не знају да програмирају уведу у програмирање 2Д видео-игре, да искористе своја знања из програма Gimp и Inkscape за креирање потребних графичких ресурса те да примењују алгоритме из рачунарске графике. У раду је дато решење где се користи програм Stencil који омогућује креирање 2Д видео-игара са визуелним програмирањем. У оквиру програма Stencil бирају се управљачки програмски блокови који имају форму “слагалице” тако да се одговарајући блокови уклапају у једну већу целину која одређује програмски ток у видео-игри. Приказано је коришћење програма Stencil на примеру развоја конкретне 2Д видео игре “Авиони”. Циљ рада је да се студенти који нису слушали предмет Увод у програмирање уведу у програмирање конкретне 2Д видео-игре коришћењем визуелног програмирања помоћу “пузле” програмских блокова програма Stencil.

**Кључне речи:** рачунарска графика, програмирање, видео-игра, Stencil, 2D

## USAGE OF STENCIL SOFTWARE IN THE COMPUTER GRAPHICS COURSE

**Abstract:** The course Computer Graphics at VIŠER includes about fifty algorithms that solve problems on this topic. The problem is that students did not have the Introduction to programming course before. The main idea is to introduce student who don't know programming into 2D video game programming, to use his Gimp and Inkscape skills to create the necessary graphics resources, and to apply computer graphics algorithms. This work shows a solution using Stencil software that enables creation of 2D video games with visual programming. Stencil has logical blocks that are in the form of a "puzzle" so that the corresponding blocks fit into one larger unit that determines the program flow of the video game. The use of Stencil is shown through development of a concrete 2D video game refers to as "Avioni". The aim of this work is to introduce students who have not listen Introduction to Programming course to programming of a specific 2D video game by using visual programming of the Stencil program.

**Key words:** computer graphics, programming, video game, Stencil, 2D

### 1. УВОД

Предмет Рачунарска графика, који се слуша у другом семестру основних студија Високе школе електротехнике и рачунарства у Београду, обухвата алгоритме из: растерске графике, 2Д и 3Д трансформација, попуњавања, скривања, морфинга, фрактала, одсецања, кривих, сплајнова те *OpenGL* технологије (Ferguson 2014). Сви примери се дају кодирани у *C++* коришћењем *OpenGL* технологије. У оквиру наведеног предмета уче се апликације *Gimp* и *Inkscape*, тако да су студенти способни да сами креирају потребне графичке ресурсе.

Основни проблем је што студенти студијског програма Нове рачунарске технологије у истом семестру слушају први програмерски предмет Увод у програмирање, док студенти студијског програма Аудио и видео технологије не

<sup>1</sup> професор, Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, pericas@viser.edu.rs;

<sup>2</sup> студент, Факултет информатике и рачунарства, Данијелова 32, Београд, e-mail:pavle.strbac.16@singimail.rs;

<sup>3</sup> асистент, Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, vkorac@viser.edu.rs;

<sup>4</sup> предавач, Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, pejanovicm@viser.edu.rs;

<sup>5</sup> професор, Висока школа електротехнике и рачунарства, Војводе Степе 283, Београд, verap@viser.edu.rs;

слушају програмерски предмет. Ако би се избегло програмирање, предмет би изгубио смисао, а ако се форсира програмирање предмет би био непримерено тежак.

Уважавајући напред наведене проблеме, поред доста програма за креирање видео-игара (Гемовић 2019) одабран је програм *Stencyl* који омогућује креирање 2Д видео-игара коришћењем програмских логичких блокова који се слажу у целину. Дат је пример употребе програмских логичких блокова на примеру игре „Авиони“.

## 2. ПРОГРАМ STENCYL

За креирање 2Д видео-игре у програму *Stencyl* користи се креирање: ресурса, сцене, актера, понашања те управљање догађајима (Borkwood 2013). У ову сврху се користе логички блокови који се као слагалица слажу и одређују програмски ток игре. На примеру игре „Авиони“ даје се приказ коришћења логичких блокова који одређују програмски ток игре.

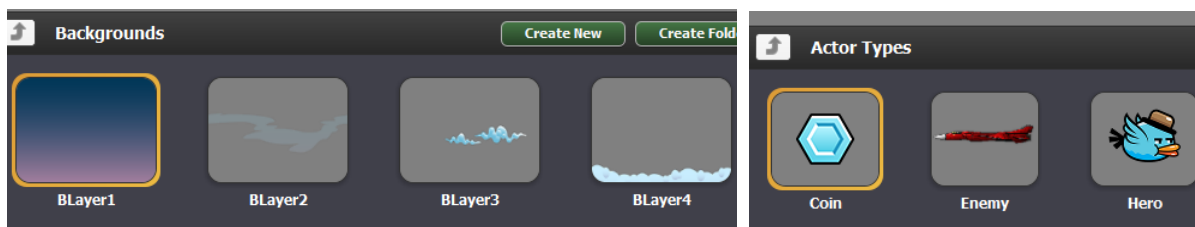
### 2.1. Креирање пројекта

На сајту: <http://www.stencyl.com>, креира се налог и преузима најновија верзија програма за оперативне системе: Windows, OS X и Linux. Налог омогућује преузимање готових ресурса који се нуде бесплатно или се плаћају на *Stencyl* продавници.

Игра која је припремљена за овај рад је 2Д видео-игра типа *Endless Horizontal Scrolling Runner*. На основном екрану апликације креира се нови пројекат резолуције 640x480 пиксела и назива „Авиони“.

### 2.2. Креирање графичких ресурса

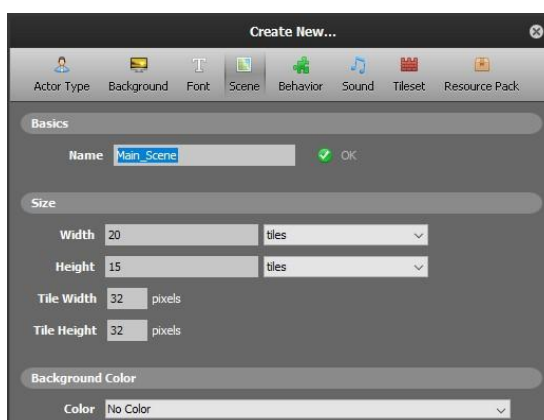
Ресурси у овом раду су или добијени инсталацијом програма *Stencyl* или су преузети са *Stencyl* продавнице. Графички ресурси су: птичица, авион, новчић и 4 слике за позадину. Позадинске слике се увозе у поље *Resource>Background* где се користе као четири слоја која дају ефекат дубине (*BLayer1*, *BLayer2*, *BLayer3* и *BLayer4*) (слика 1). У поље *Actor Types* се уносе актери игре: новчић (*Coin*) са анимацијом од 6 фрејмова, птичица (*Hero*) са анимацијом од 4 фрејма и авион (*Enemy*) (слика 1). Птичици и авиону се искључује гравитација и ротација (таб *Physics*). Студенти сами креирају и увозе графичке ресурсе. За креирање графичких ресурса студенти користе програме *Gimp* (Hoofst 2018) и *Inkscape* (Strode, Baechler 2019), који се изучавају на предмету Рачунарска графика.



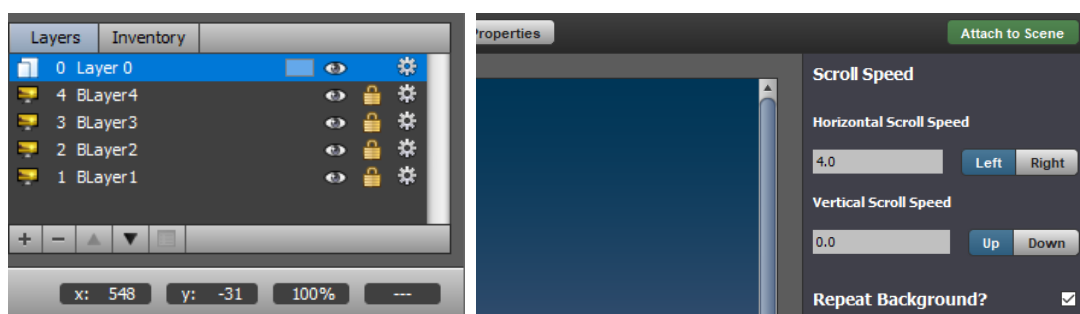
Слика 1 – Сlike 4 позадинска слоја и актери игре

На сцену, чије су вредности параметара дате на слици 2, поставља се птичица на нулти слој, а онда се убацују позадински слојеви. Нулти слој мора бисти стављен испред осталих слојева да би се видели актери (слика 3.а). За слојеве се поставља

брзина померања (*Horizontal Scroll Speed*) унутар подешавања увезених позадина те поставља опција понављања позадине (слика 3.б).



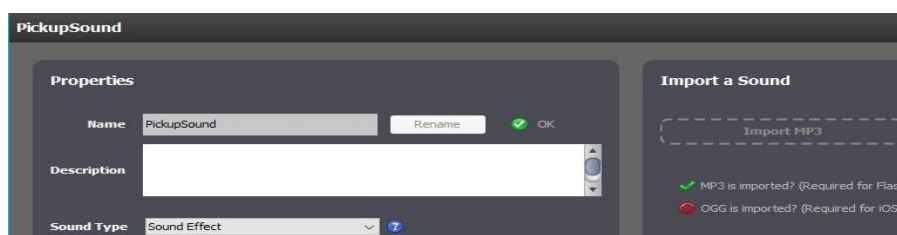
Слика 2 – Параметри сцене



Слика 3 – а) Слојеви сцене, б) Брзине скрловања и понављање позадине

### 2.3. Креирање аудио ресурса

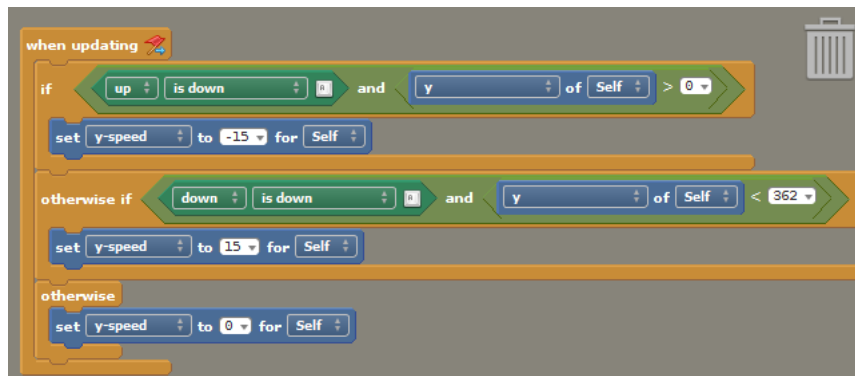
*Stencyl* барата са *MP3* и *OGG* аудио форматима. У раду је коришћен *MP3* аудио формат. Аудио ресурсе је потребно креирати у оквиру опције *Resources>Sounds*. Могуће је креирање два типа звука: музика (*Music*) која се користи као позадинска музика и звучни ефекти (*Sound Effects*). У примеру се користе: *MainSong* – позадинска музика, *PickupSound* – звучни ефекат купљења новчића и *ExplodeSound* – звучни ефекат судара птичице са авионом. На слици 4 дат је пример креирања аудио компоненте *PickupSound* где се увезеном аудио фајлу додељује назив и тип звука. Студенти користе готове аудио компоненте или их модификују коришћењем одабраног програма *Audacity*.



Слика 4 – Креирање PickupSound аудио компоненте

## 2.4. Креирање програмске логике

Студенти за креирање логике користе визуелно слагање логичких блокова у оквиру програма *Stencyl*. За лет птичице креира се понашање *PlayerControll* где се додаје догађај *+Add Event>Basic>When Updating* који управља кретањем птичице горе-доле (слика 5).



Слика 5 – Логика управљања птичицом

За сцену (*Scene*) се креира понашање *SceneManager* коме се додају догађаји. Догађај *Keyboard*, креиран са *+Add Event>Input>Universal Keyboard*, се односи на почетак игре и активира се притиском на тастер *Enter* (слика 6). Догађај користи скривени атрибут *Playing* типа *Boolean*. Ако игра није почела и притиснут је *Enter*, онда ће се атрибут *Playing* поставити на *true* и активираће се догађај *SpawnEnemies* који је додељен понашању *SceneManager*.



Слика 6 – Логички блок за тастер ентер

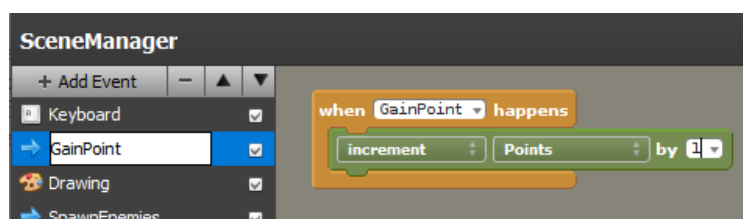
*SpawnEnemies* је догађај који се креира са *+Add Event>Advanced>Custom Events >Custom Event* и који генерише авионе на случајној позицији и са случајном брзином кретања (слика 7). *Random integer between* представља генератор случајних бројева који насумично бира вредности за висину на којој лете авиони (0 - врх сцене, 352 - дно сцене). Блок *do after seconds* позива креирање авиона сваких 0.5 секунди (слика 7).



Слика 7 – Догађај SpawnEnemies

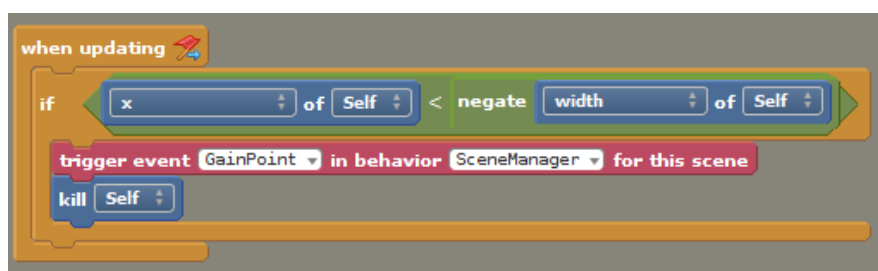
У догађају *SpawnEnemies* проверава се да ли је скривени атрибут *TempNum* типа *Number* достигао број 5 (броји од 0 до 5), чиме ће се након сваког 5. креираног авиона креирати један новчић на крајњој левој позицији сцене, на случајно одабраној висини и са брзином -24 по *x* оси.

Број поена једнак је броју авиона изашлих са сцене. Унутар *SceneManager* додаје се догађај +Add Event>Advanced>Custom Event који ће се назвати *GainPoint*. За овај догађај је потребан скривени атрибут *Points* типа *Number*. Блок кода који увећава број поена *Points* за 1 дат је на слици 8.



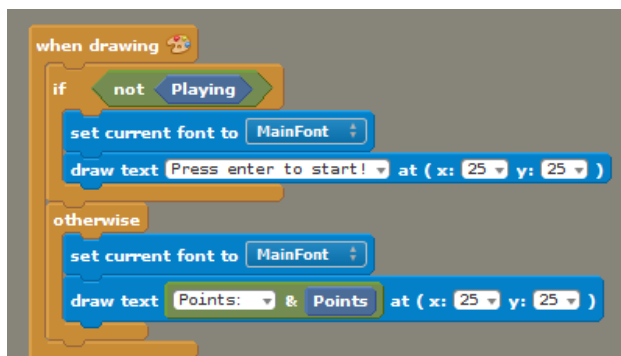
Слика 8 – Догађај GainPoint

Авиону се додаје понашање *EnemyBehavior* ((+Add Event>Basic>When updating) које повећава број поена и уништава објекат авион када исти изађе са сцене (проверава се *x* координата) (слика 9).



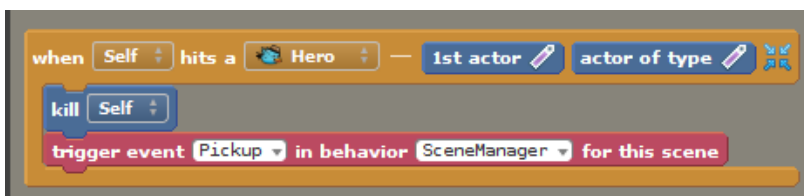
Слика 9 – Позив догађаја *GainPoint* пре уклањања непријатеља

У *SceneManager* се додаје догађај исцртавања *Drawing* (+Add Event>Input>Basic>When Drawing ) који на сцени црта текст „Press enter to start!“ када игра није стартована, односно, црта број поена на сцени ако је игра у току (слика 10).



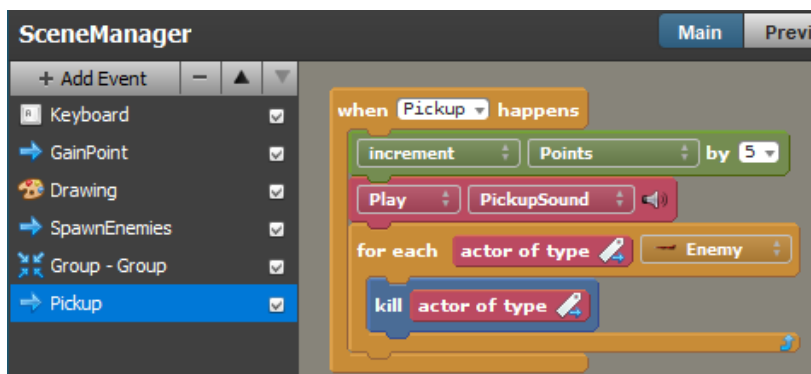
Слика 10 – Догађај исписивања поена

Новчићу је придружено понашање *PickupBehavior*. Унутар овог понашања додаје се догађај +Add Event>Colisions>Actor of Type, који ће уништити новчић након што га птичица додирне при чему ће се позвати догађај *Pickup* у понашању *SceneManager* (слика 10).



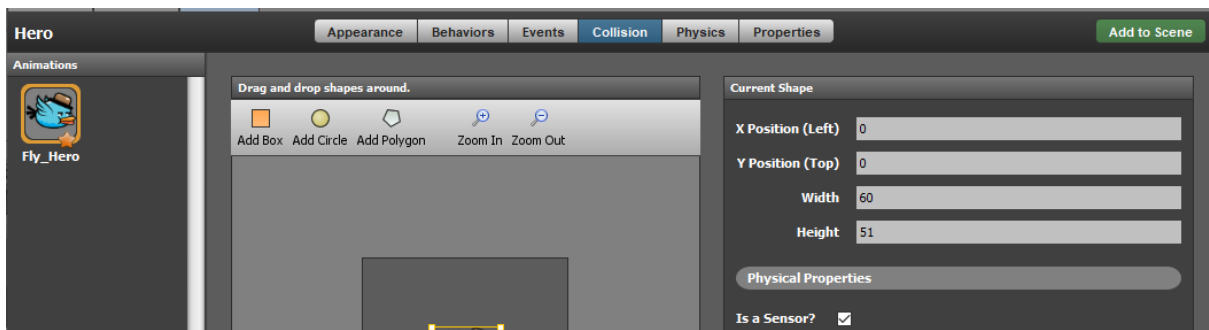
Слика 10 – Догађај уништавања новчића

Унутар понашања *SceneManager* додаје се догађај *Pickup* који при купљењу новчића повећава број поена за 5 и уништава све авионе на сцени (слика 11).



Слика 11 – Уништавање авиона на сцени и повећање броја поена при купљењу новчића

У подешавању *Settings>Group* постављају се колизије група (*Coin*, *Actors*, *Players*), а онда се групе додељују актерима у табу *Properties*. За птичицу се поставља својство сензор у оквиру таба *Collision* (слика 12).



Слика 12 – Постављање сензора за птичицу

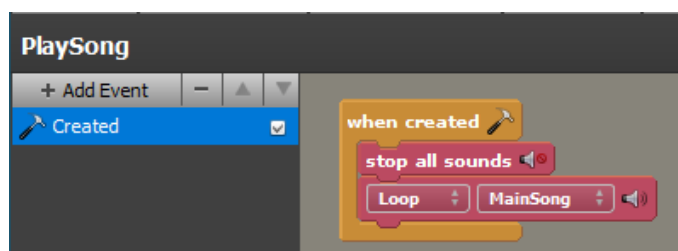
У понашању *SceneManager* додаје се догађај (+*Add Event>Collisions>Group&Group*) који при судару птичице и авиона: активира звучни ефекат *ExplodeSound*, уништава сударене објекте и рестартује игру (слика 13).



Слика 13 – Код за колизију и рестартовање игре

## 2.5. Програмска логика за аудио компоненте

Аудио компонента типа *Music* намењена је као позадинска музика. За укључење аудио компоненте у игру потребно је креирати понашање *PlaySong* за главну сцену. На слици 14 дат је логички код који при креирању игре, зауставља све звукове и пушта у непрекидној петљи позадинску музику (слика 14).



Слика 14 – Понашање *PlaySong*

За звучни ефекат *PickupSound* додат је логички блок који пушта аудио запис унутар догађаја *Pickup* из *SceneManager* што је приказано раније на слици 11.



Слика 15 – Активирање звучног ефекта *ExplodeSound*

Звучни ефекат *ExplodeSound* додат је логички блок за догађај Group – Group који је додат понашању *SceneMenager*. У логичком блоку се испитује да ли је дошло до колизије птичице и авиона те ако јесте, онда се активира звучни ефекат *ExplodeSound*, униште се учесници колизије и поново иде на почетак игре када се очекује притисак на тастер *Enter*.

### 3. ЗАКЉУЧАК

У оквиру предмета Рачунарска графика, који се слуша у другом семестру на ВИШЕР, за примену стечених знања и креирање 2Д игара одабран је програм *Stencyl*. *Stencyl* омогућује креирање програмске управљачке логике коришћењем слагалица кода. Овим је обезбеђено да студенти који тек тада слушају Увод у програмирање могу да креирају програмске блокове које ће обезбедити логику игре. Са друге стране, студенти креирају потребне графичке ресурсе коришћењем програма *Gimp* и *Inkscape*, који се уче у оквиру предмета, а онда те ресурсе користе у програму *Stencyl*. У раду је дат пример коришћења слагања логичких блокова програма *Stencyl* за креирање 2Д видео игре "Авиони" са тежиштем на коришћење: понашања, догађаја, атрибута, група, физике и колизије. Програм *Stencyl* је прошао двогодишњу пробу у оквиру предмета Рачунарска графика на ВИШЕР и показао се као подесан за креирање видео-игре и развијање програмерске логике за оне који нису програмирали или су почетници.

### 4. ЛИТЕРАТУРА

- [1] Borkwood, I. (2013). *Learning Stencyl 3.x Game Development Beginner's Guide*. Birmingham. Packt publishing.
- [2] Гемовић, Б. (2019). *Примена анимација за израду 3д игара*. 5. међународно саветовање управљање знањем и информатика, стр. 106-114.
- [3] Ferguson, S. (2014). *Practical Algorithms for 3D Computer Graphics*. Boca Raton, London, New York. CRC Press.
- [4] Hooft, B. (2018). *The Ultimate GIMP 2.10 Guide: Learn Professional photo editing Paperback*. Independently published.
- [5] Strode, M., Baechler, O. (2019). *Sams Teach Yourself Inkscape, Gimp and Blender in 24 Hours*. Indianapolis. Sams Publishing.