

DEVELOPMENT OF FLIGHT SIMULATION EDUCATIONAL GAME

VALENTINA PAUNOVIC

Belgrade Metropolitan University, valentina.paunovic@metropolitan.ac.rs

TATJANA GVOZDENOVIC

Belgrade Metropolitan University, tatjana.gvozdencovic@metropolitan.ac.rs

RADOSLAV STOJIC

Belgrade Metropolitan University, radoslav.stojic@metropolitan.ac.rs

Abstract: *In this paper we are describing the development of educational simulation game with audio-visual environment for flight simulation for a light aircraft at given airport using 3D game technology. The developed serious game is designed for the students initial familiarization how to control airplane flight. The development project was organized by a spiral model in which the cycles represent the individual student projects as a part of the education process. The paper describes the co-operation between the designer and developer (students of BMU) while creating a functional project.*

Keywords: *e-Learning, Educational systems, Educational games, Flight simulation game*

1. INTRODUCTION

Modern learning involves adaptive personalized learning to be more appropriate to human nature in comparison to the most popular learning method – Reading, considering that studies show how readers remember twice as more if they listen, rather than read, this increases five-fold if observation is involved [1].

Given that the goal of every teacher and every scientific institution is to provide the best possible education, there has been an increase of simulator usage in teaching. Simulators are computer programs (such as games or animated flowcharts), or dedicated devices that model some aspect of real life situations (For example: flying an aircraft), and can be manipulated to observe the outcome of different assumptions or actions, without exposing the experimenter to any danger or risk. Studying is the most effective when a student is placed in a real laboratory environment by use of simulators that allow practical exercises. Because of this reason, there is an increased tendency towards multimedia lectures, and one of these types of lectures involves simulators working on principles of serious games. Students who study lectures based on serious games and simulation can gain faster and more efficient knowledge, than in a traditional way [1]- Analysis have shown that teachers and students using such courses have more motivation to work, have better learning outcomes in comparison to the classes that are not based on realistic simulations and games. We must take into account that not every type of course is suitable for this kind of approach, but students listening to courses that can implement this approach had a easier time mastering the subject and often asked for it's implementation in conducted Surveys. In order to create a

functional simulation game, it is mandatory to start with learning objectives and create a detailed plan of simulation objects, to create a pleasant graphical presentation of given object and its environment, to program movement and object physics in a way that best mimics their real world counterparts.

This paper will present the creation of a flight simulation game that that uses the keyboard (rather than a dedicated hardware on training flight simulators) using Unity 3D game engine.

The simulator can be useful to the BMU students, as well as to students of civil engineering and military aviation. Making a video game requires a combination of artistic and technical knowledge. The domain of so-called “serious games” is even more demanding, since it requires good comprehension of teaching goals, laws of physics and mechanics, along with the usual requirements of making a video game.

In order to cope with the complexity of creating a “serious game”, and it's multi-disciplinary requirements, the project described in this paper involved a students of Design, and a students of Software engineering.

This article will show experiences in using computer games in teaching and education in the BMU, where a doctoral student and graduate student work together using basic research techniques to solve the given problem. The contribution of this paper is showing different approaches to learning that have been shown as more useful for students in learning and employment.

2. USING THE SERIOUS GAMES IN TEACHING

Serious games and their potential as a teaching tool were first accounted for in a book „Serious Games“ by Clark C. Abt in the year 1975.[2] The autor of the book states: „We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement“. This educational potential shouldn't be focused only on video game design and IT, but also sports,military etc.

Considering the long history of focus towards creating a harmony between learning and having fun, the design of multimedia learning poses a question how can serious games be different from Edutainment (content designed to educate and entertain at the same time), entertainment education or e-learning. Michael and Chen [3] committed a analysis of using serious games in education, and reached a conclusion that this form of education far outshines the “traditional” approach. There is no more need to learn materials by rote.

Most designers and researchers agree that serious games are taking a step forward from Edutainment, in a sense that Edutainment games present a sub-category of serious games. This research classifies edutainment games as games belonging to a family of serious games, created for use in K-12 education. These games focus on transferring standard knowledge (like those in textbooks) using games as a method of motivating students. Seeing as e-Learning includes disciplines like Psychology, pedagogy, computer and information sciences, serious games can be used to enhance this type of learning, making it more adaptable and personalizable.

One of the biggest advantages in e-Learning is the flexibility in time and location of learning, and if we would use the broadest definition, we could say that E-learning falls under computer-based learning, and that would mark serious games as its sub-category.



Image 1: Relation between serious games and other related education concepts

The team project in this article istied to a serious game – a Airplane simulator.

The plan was to use a student project (commonly consisted of homework and individual and team projects) as a training to master the concept of creating a functional serious Game (in this case: an Airplane simulator).

In order to create a game, we first need to create design, and comprehend the behavior of a plane, and transfer that exact behavior that into code.

BMU has multiple programs that engage in making videogames and their design. Those would be: Interactive media at FDA (Faculty of Digital Arts), Computer Games at FIT (Faculty of Information Technologies) at bachelor level, Computer science at FIT at both master and doctoral course level [4].

The Interactive media course is focused on arts, and its basic goal is to teach students how to create animated 3d object in a virtual environment. These techniques are based on tools for drawing and modeling (Photoshop, Illustrator, Maya, 3D Studio Max) and basic uses of game engines (e.g. Game Maker Studio, Unity 3D)

The Computer game course on FIT is based on computer science and software engineering fields, with a focus on creating software and programing intelligent objects. Master and Doctoral studies put a focus on advanced 3d modeling and programing of complex object behaviors.

3. CREATING FLIGHT SIMULATOR AT BMU

The first flight simulator, not based on wind, was used in 1910 for training and could be considered a structure. This simulator consisted essentially of two barrel halves, one placed on a pedestal and the other which represented a swinging cockpit. The pilot sat in the upper half barrel, which was moved manually and then had to control various flight attitudes (orientation relative to the Earth's horizons).

The most commonly available computer based Aviation simulator (games) are:

- Microsoft Flight Simulator X
- Strike Fighters 2 and its expansions
- IL-2 Sturmovik
- Lock on Flaming Cliffs 2
- DCS Black Shark
- DCS Black Shark 2
- DCS A-10C Warthog
- Flaming Cliffs 3
- Rise of Flight
- IL-2 Cliffs of Dover
- Wings of Prey
- Falcon BMS
- Flight Pro Sim

Some of these simulators are used as training for pilots, and require use of a joystick and a wide range of additions that enhance the simulation of flying a real plane

In order to create a simulator, it is necessary to develop a detailed creating plan for simulation object, the object must be graphically represented (must draw object, or modeled it object...), after that we must programmed the movements, physic and environment for object... Scripts for this project are created in JavaScript and 3D Unity framework. 3D model is created in Maya.

In order to create a functional simulator, it is mandatory to create a detailed plan of simulation objects, to create a pleasant graphical presentation of given object and its

environment, to program movement and object physics in a way that best mimics their real world counterparts. The project was created in the Unity 3D framework environment, Models we're created using Maya.

Unity 3D (U3D) is a framework used in development of video games and other 2D and 3D content. It was created by Unity technologies in 2005. This framework allowed small teams of programmers to create games that would otherwise require a substantial number of people. Unity is a simple and effective environment for developing simulations that has support for multi-platform development. It is based on the C/C++ programming languages.

It is powered by a strong graphical engine and a suite of tools that allow intuitive and fast ways to create functional worlds for different platforms. At this time, unity can be used to develop software for iOS, Android, Windows, Blackberry 10, OS X, Linux, Browsers, Flash, PlayStation 3, Xbox 360, Windows Phone 8 and Wii U

Unity is a great productivity tool, and it can dramatically decrease time needed to fix programming mistakes, and by that decrease the costs associated with software development. It is used for organizing available resources, adding lighting and effect, physics and animation, testing and maintaining project, as well as publishing the end-version of our software.

Resources include: 3D models, Materials, textures, sounds, pictures, programming scripts and fonts. It supports a wide array of formats, making it's resource system robust and intelligent. All resources are hierarchically organized in scenes

Scripts are used for describing resource behavior – a scene is created by code and resources become interactable. We can tie multiple scripts to an object, which promotes reusability of code. Unity supports C#, Javascript programming languages, and they can be used together inside a same project.

For the purpose of this simulator, a thesis of our FDA student was used [10]. This thesis covered the subject of visually modelling an Airport and cockpit of a plane. Afterwards, a simulation was created in Unity 3D (airport, planes, moving elements, primary scripts)



Image 2: Primary version of the simulator

Airport of the town Vršac was used as the object of our model. This airport hosts the national pilot academy for civil aviation.

The plane used for training is the Light Cessna 172, hence this is the plane used for our 3D model. The FDU student created the planes primary movement without take-off [10].

A student of FIT made the scripts for the given model, hence allowing a realistic depiction of movement and take-off of the plan, by adding

- Control of basic maneuvers (roll, pitch, yaw, forward)
- Acceleration of planes movement forward by using a voice
- Control of the nose wheel of the plane by using pedals.
- A realistic depiction of the instrument board
- Scripts for animating the needles on the instrument board

4. CESSNA 172

The developed flight simulator Cessna 172 is made from the model of an airplane used for training on the Vršac airport and contains all the needed instruments required for pilot training. Cessna 172 skyhawk is a 4seat airplane with 1 motor and fixed wings, the model was developed by Cessna aircraft company. First flight occured in the year 1955. This model is one of the most produced airplanes in the world, counting 43.000 planes by the year 2015.



Image 3: Cessna 132 overview

What sets Cessna plane apart is a wide specter of modifications, powerfull mottors (from 134KW to 157KW), constant speed propellers even using car fuel. Besides that, it is possible to increase the reservoir capacity and luggage space.

The look of the cockpit is shown below



Image 4: Cessna 172 cockpit

5. GETTING TO KNOW THE FUNCTIONING OF THE AIRCRAFT FOR SCRIPTS WRITING

The plane has 3 wheels, one of which is used for steering, and two for holding the planes weight

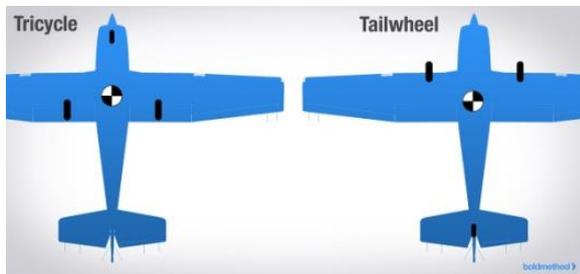


Image 5: Tricycle landing gear and tailwheel landing gear

With the so-called tricycle alignment, the steering wheel is housed in the front of the plane, rather than the back. Relations and scripts which will be shown can be used with both alignments, but the tricycle alignment will be shown.

On the ground, a pilot uses the pedals to move the nose wheel right and left in order to steer the plane in a desired direction.

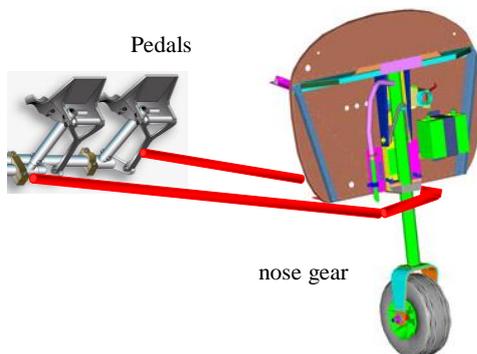


Image 6: Nasal wheel is controlled by pedals

It is important to note that the pedals also steer the rudder of the plane, which has a secondary role on the ground (since the plane is slower on ground than in flight, and hence, the aerodynamic forces are less intensive)

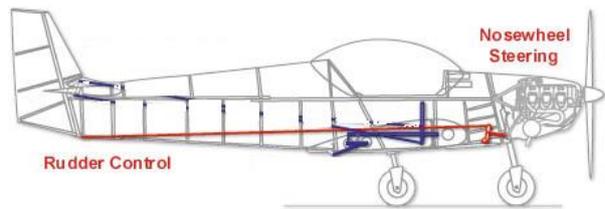


Image 7: Rudder is managed by rudder pedals

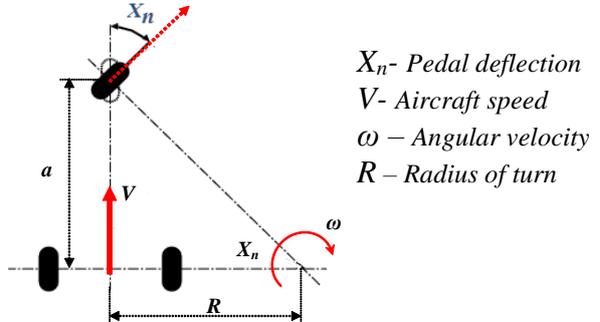


Image 8: Units of the taxiing (movement of aircraft on the ground)

Let us begin with the known relations (sizes are indicated in the figure)

$$V = R \omega$$

$$a = R \operatorname{tg} X_n \approx R X_n$$

thus affirms

$$\omega a = \omega R X_n = V X_n$$

If a is the distance between wheels, we can assume that:

The Angular velocity is proportional to the speed of the airplane V and pedal deflection (nasal wheel) X_n .

So we can state

$$\omega = r_{max} V X_n$$

where we have introduced the constant $r_{max} = 1/a$,

Now we can write the code to update the state of the plane after the pedals have been moved. Let's suppose that the plane is moving forward as well as to the side in order to reach Niš (or Novi Sad) from Belgrade. As with the car or other ground vehicles, the plane also adjusts height (in order to take off, land or move over an obstacle), but this was already done. Earlier scrips allow for the flight forward, as well as changes in altitude, but moving to the sides (the way the plane acts in these situations) is yet to be implemented.

The first idea is that we need to use the peddals to move the rudder, same as we would with a boat. But it's not that easy. This can be explained in the Wright brothers manner, if the plane flies straight, the wings are horizontal and generate the so-called lift force which keeps the plane in the air by balancing it's weight.

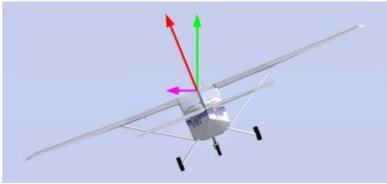


Image 9: Flights and forces in turn

If we have plane in turning whose radius is R and it moves with constant velocity V . By what degree should the plane be tilted (tilt angle is ϕ) in order to achieve a turnaround?

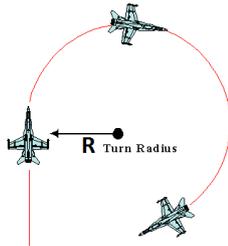


Image 10: Airplane in turn

The flight speed turning is proportional to the angle of inclination ϕ and inversely proportional to the speed of the plane V .

Another thing that needs to be noted is that in order to improve the plane we also need to make the needles of the control board operational.

First we need to make the needle (for example: cube), we make the plane behind her and apply the picture of the background(the scale) and implement the script for the needles movement.

The needle will now move over the point in her center, not over the point in the end(as do most needles), in order to fix this, we need to:

1. Make the needle.

We create the cube and scale it so it looks like a needle in U3D. If we want a more realistic look, we need to create a needle in Maya, export it in the fbx format and import it in U3D

2. Create a game object (GO) and name it „Indicator“

The point of this is that the needle(cube) can rotate over the point at the end, not in the middle (where the default coordinating point is set)

We set the cube in a new GO, and move it so the point on the end is located in the middle of the GO

3. Creating a test script

Using the arrows \leftarrow and \rightarrow we rotate the indicator (the needle) in one direction.

4. Creating the size scale

We find an appropriate image of the instrument board, we separate the picture into a file (taking care that the dimensions are appropriate for the size of the textures, 32x32,64x64 etc.) and import it into the U3D project

After this, we create a plane, and put it behind the needle, then we drag&drop the background image of the instrument.

6. CREATING SCRIPTS FOR AIRCRAFT SIMULATOR UPGRADE

The behavior of GameObjects is controlled by the Components that are attached to them. Although Unity's built-in Components can be very versatile, you will soon find you need to go beyond what they can provide to implement your own gameplay features. Unity allows you to create your own Components using scripts. These allow you to trigger game events, modify Component properties over time and respond to user input in any way you like. That means that simple definition for scripting in Unity is how a programmer defines the behavior for object in the game.

Unity supports two programming languages natively:

- C#
- UnityScript, a language designed specifically for use with Unity and modelled after JavaScript.

Application programming interface (API) is a set of subroutine definitions, protocols, and tools for building software and applications. A good API makes it easier to develop a program by providing all the building blocks, which are then put together by the programmer.

The scripting reference is organised according to the classes available to scripts which are described along with their methods, properties and any other information relevant to their use. API in Unity are grouped by namespaces they belong to, and can be selected from the sidebar to the left. For most users, the UnityEngine section will be the main port of call.

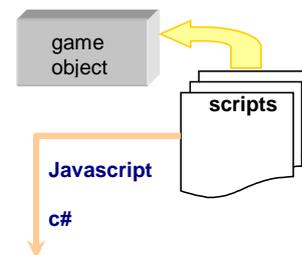


Image 11: Behaviors of objects are defined by the scripts



Image 12: Beta model

Before it came to work on the already modeled aircraft which was created for the purpose of graduate thesis,

primarily we created the airplane model in the order to better understand and create scripts which would operate the plane. It is necessary to define what exactly a plane should work and how that can be programmed. That is created by defining the beta model for testing scripts which can be seen in the following figure.

After creating script for beta model and after the testing, script code is transferred to existing model - primary model.

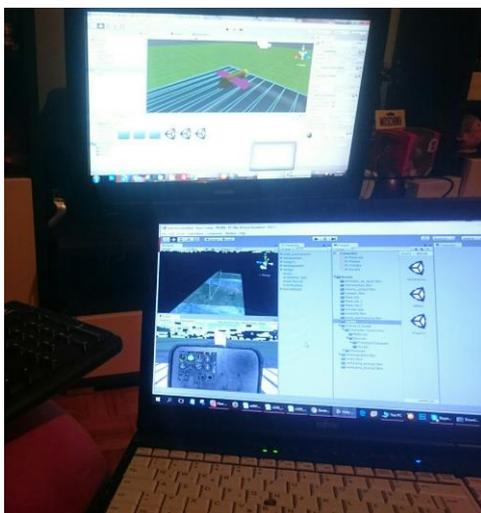


Image 13: Beta model preview and script transferring at the primary model

After creating the scripts and after using those scripts in the existing model, we have a plane with the ability to take off by using realistic physics (Image 14). The first part of image depicts the take-off and second the flight.

5. CONCLUSION

In this paper, we adopted approach that can be used in traditional and e-Learning, using serious games as a teaching tool, as well as a presentation of powerful teamwork between the FDA and FIT students at BMU.

The simulation game created for the purposes of this article demonstrates the increased interest for competition of specific project even when they exceed the bounds of the teaching plan.

The study shown provides encouraging results and a high interest of faculties students to work at teams and to create a final product at the end of completing the course, considering that the students successfully created the first version of the flight simulator which is planned to support VR technology and an ability to adjust flight controls for other types of aircrafts in the future, it can be said that this type of learning gives students stimulus for further work and therefore greater experience in working on real problems to graduates, helping in future employment.

From what we learned, we can say that serious games in cooperation with e-Learning present the future of learning and can be expected to become a standard in the advanced adaptive learning.



Image 14: Takeoff and flight

ACKNOWLEDGMENT

This work was supported by Ministry of Education, Science and Technology (Project III44006).

REFERENCES

- [1] Michael, David R., and Sandra L. Chen. Serious games: Games that educate, train, and inform. Muska&Lipman/Premier-Trade, 2005.
- [2] Abt, Clark C. Serious games. University Press of America, 1987.
- [3] Michael, David R., and Sandra L. Chen. Serious games: Games that educate, train, and inform. Muska&Lipman/Premier-Trade, 2005.
- [4] <http://www.metropolitan.edu.rs/osnovne-studije>, retrieved 2016
- [5] Breuer, Johannes S., and Gary Bente. "Why so serious? On the relation of serious games and learning." *Eludamos. Journal for Computer Game Culture* 4, no. 1 (2010): 7-24.
- [6] N.Ćeranić, M. Tanić, A. Đokić, I. Sabo, "On the role of audio effects in realism of virtual reality systems", *eLearning 5-th Conference of e-Learning 2015*
- [7] RadoslavStojic, Ivan Vujic, Milan Stojic, "Generic Model of Vehicle Dynamics for Low-Cost Simulators and Serious Games", *Journal of Emerging Trends in Computing and Information Sciences*, Vol. 4, No. 2, February2013
- [8] Timcenko O., Stojic R.: "On Problem Based Learning and Application to Computer Games Design Teaching", *Int. J. Of Emerging Technologies in Learning (iJET)*, Vol. 7, February, 2012, pp. 21-27
- [9] V.Paunovic, D.Domazet "Set of metadata established for application in learning material developed for BMU", *Conference of e-Learning 2013*
- [10] T.Gvozdencovic, Razvoj edukacionog vizuelnog okruženja za simulaciju leta, graduate thesis, BMU, 2016.