Comparison of Software Application Development Procedures in C++ and C# Programming Languages

Maša Knežević*, Eleonora Brtka * and Igor Vecštejn*

* Technical Faculty ‘Mihajlo Pupin’ Zrenjanin, Serbia

littlepanda.mk@gmail.com, eleonorabrtka@gmail.com, igor.vecstejn@gmail.com

Abstract - The topic covered in this paper aims to present the creative application and development of a software application in the C # programming language. The motive that is crucial for processing this topic is that the process of programming and development of software applications need to be presented in an interesting and tempting way, in order to motivate other students to use and apply programming languages and programming in general. In order to make learning a new programming language more interesting, it is necessary to emphasize the practical application of each step. By emphasizing loops and logical procedures as well as their application, we can more easily understand and accept.

I. Introduction

The motivation is the primary fact in this paper. When we got creative interpretation and using of some programming languages the students are more interested in that topic and they have better concetration on the class. Making some games or other project can show them how programming is easy and how it can be usefull. Programming like mathematical example can be understood by some students, but not to all of them.

In this day and age we are expected to follow the new daily technological advances. Everything on ours phones, computers or some other gadgets are interactive, funciolan and full of colours.

Children of this days in the first 3 years of their lives learned how to use a mobile phone. Our mobile phones are interactive and full of colours. We can present our class of programming in the same way.

In this paper is that one way how easy and fast students can understand and use procedures in C++ and C# programming languages. The game „Pac-Man” is made in Unity with C# programming language. Unity allowed us to make animation of our actors, control audio sound and to see visual use of some procedures.

II. Previous Work

The key to object-oriented programming is that the data contains not only the properties of the entities but also the operations on them. When the work with one class is finished, only then does the work with another class begin, and only when everything is finished does the programmer return to the initial problem. The characteristic of object-oriented programming is expressed precisely by the brevity of the main program. The goal is to create classes that will not be rewritten, but used over and over again. [1]

An object represents the uniqueness of the properties by which it stands out from other objects and by which it is recognizable, it is called identity. If we enter in a library, we will realize that the library is made up of shelves and books. Books can be classified into categories: classics, tragedies, novels, poetry, prose… Each book is characterized by a certain author, publisher, genre, date of publication, circulation, main characters, number of pages, short description of the work but also criticism. In this case, the book represents an object, and all these characteristics or attributes represent what makes it unique.[5]

C ++ is a higher level programming language. It arose in response to the need to create objects. It provides an easy and safe way to manage the fields of an object, more precisely attributes. Inheritance solves the problems of concretization and specificity of each algorithmic problem. C ++ is an upgrade of the C programming language. The difference between the C ++ and C programming languages is the way how data is input and output and the ability to manipulate objects through the introduction of classes.

The C # programming language is a language that allows you to write console and Windows applications, and even Web applications. It is one of the younger programming languages. It was created in 2002 and is an integral part of MS .NET Framework 1.0. [2]

The C++ programming language is a low level programming language that adds object-oriented features to its base language C whereas C# is a high level language. The C++ compiles down to machine code whereas C# compiles down to Common Language Runtime, which is interpreted by JIT in ASP.NET. The C++ programming language is an object-oriented language while C# is considered a component-oriented programming language. In C++ is nessesery to manage memory manually whereas C# runs in a virtual machine, which performs memory management automatically. In C++ development should follow any specific architecture and must be portable whereas C# development should be simple, modern, general-purpose, object-oriented programming language. [3]

III. SOFTWARE APPLICATION DEVELOPMENT

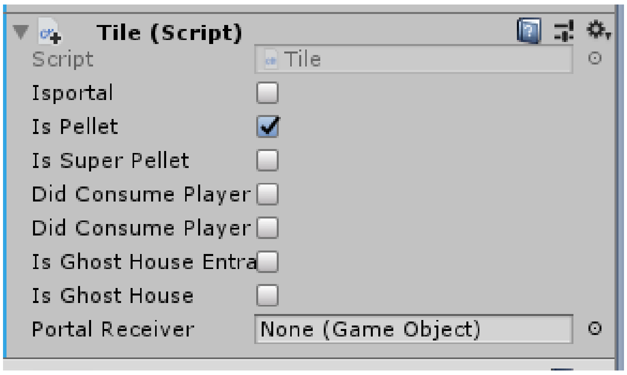
PacMan is an arcade game developed by Namco. The game was first released by Midway in Japan on May 22, 1980. The original name of the game was Puck-Man, but that name was changed due to the European standard. The game can be played by a maximum of two players, alternately. Movement was enabled then through the joystick, and now also through the computer keyboard in four directions (up, down, left and right).

The main character is called PacMan, he is yellow. He has three lives in the game, his goal is to collect all the so-called pills that are on the screen. The pills are arranged around a maze and PacMan moves through that maze. The obstacle that poses a problem for passing the game are the ghosts chasing PacMan. The ghosts are called: Blinky, Pinky, Inky and Clyde.

The begin of creating our software aplication is a map. Map is place in game where our actor Pac-Man is moving all around. In this part of the paper is important to know one field of math. Pac-Man way is based on vecotrs. The points should be distributed the same in relation to the X or Y axis. It is important that they stand in the appropriate plane and that the vectors match adequately. In case there is a deviation, our hero will not be able to move later. The movement of the heroes is based on vectors, more precisely on the directions of the vectors. And that is made possible thanks to those points that we will place over the entire map of our game.

On the map we distinguish: nodes, portals, pills and super pills. A knot is any pill that has a turn, we look at it as a crossroads. Portals are those pills that do not bring points, but therefore allow you to move from one end of the map to the other end of the map. Pills are all pills that are visible on the map and assign a certain number of points. Super pills are those pills that change the state of our heroes and they also change roles. By consuming the super pill, the points are doubled, PacMan has the ability to attack ghosts, PacMan moves faster, but that is why ghosts move slower.

A Tile script is added to each pill. The Tile script allows object identification. More precisely, it allows us to assign any pill, whatever role we want. *(Figure 1)*



Tile (Script)

Script: Tile

Isportal □

Is Pellet ☑

Is Super Pellet □

Did Consume Player □

Did Consume Player □

Is Ghost House Entra □

Is Ghost House □

Portal Receiver: None (Game Object)

Figure 1- Using of Tile script

If we mark that the so-called pill is a node or intersection, it is necessary to join the Node script. Using the Node script, we indicate which nodes are adjacent to a given node and how many adjacent nodes there are. Each pill has its own unique number. When we draw nodes with given unique numbers, we get a grid of motion. *(Figure 2)*

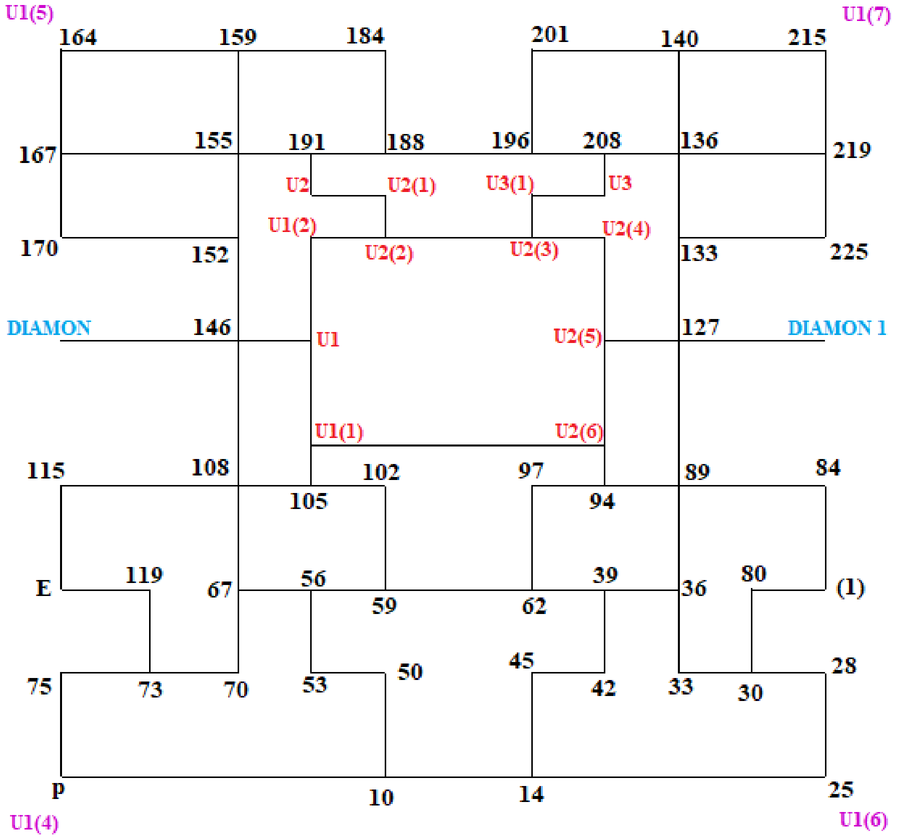
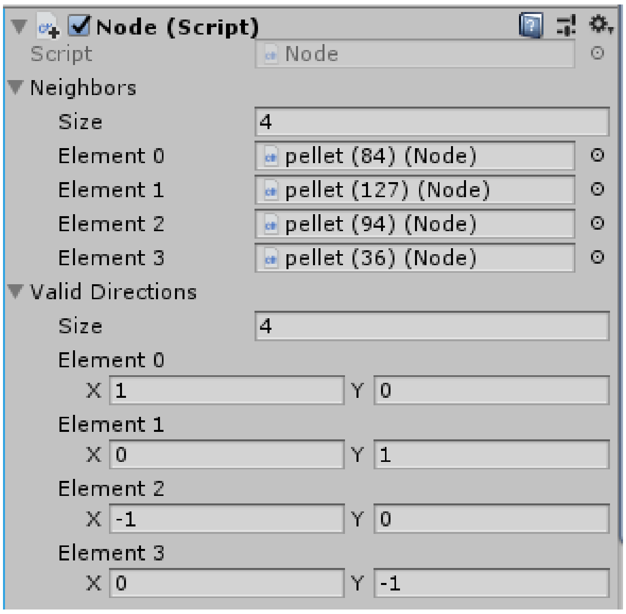


Figure 2- The Grid of map's nodes

Each node needs to define its size (number of neighbors), name the neighbors and define the direction of movement towards those neighbors. The definition of the direction of movement of the neighbors is determined on the basis of the X and Y axes. If the movement is in the direction of the X axis or the right movement is defined on the number 1, if the movement is opposite to the X axis or the left movement is defined with the number -1. If the movement is defined upwards in the direction of movement of the Y axis, it is defined by the number 1, and if the movement is performed downwards opposite to the direction of movement of the Y axis, it is defined by the number -1.For example, if we consider pill 89. Pill 89 has 4 adjacent nodes. Adjacent nodes are 127, 84, 94 and 36. Node 127 determines the upward movement and the value is along the Y axis 1, and along the X axis 0. Node 84 determines the movement to the right and it indicates movement along the X axis 1, and along the Y axis 0. The downward movement is defined by the adjacent node 36 and its value along the Y axis is -1, and along the X axis 0. Node 94 denotes the movement to the left and its value along the Y axis is -1, and along the X axis 0. *(Figure 3)*


☑ Node (Script)

Script Node

Neighbors

Size 4

Element 0 pellet (84) (Node)

Element 1 pellet (127) (Node)

Element 2 pellet (94) (Node)

Element 3 pellet (36) (Node)

Valid Directions

Size 4

Element 0

X 1 Y 0

Element 1

X 0 Y 1

Element 2

X -1 Y 0

Element 3

X 0 Y -1

Figure 3- Using of Node script

The super pill has the task of making the game more fun. There are only four super pills on the whole map. Super Pills are defined using the Tile script where the Super Pellet option is checked. By consuming super pills the spirits transition into their Frightened mode. This means that they change their animation into a timid and blue ghost and their speed of movement slows down. PacMan accelerates its movement and has the ability to eat the ghosts. When PacMan consumes ghost, extra points are earned and each subsequent point gets its own specific bonus. After consuming ghosts, the ghost passes from its timid state to Consumed mode, more precisely to a state where only the eyes appear and at high speed they go to their initial position where they return to their normal shape with normal speed of movement. *(Code 1)*

```
        if (tile.isSuperPellet) {
                          GameObject[] ghosts
                       =
GameObject.FindGameObjectsWithTag("Ghost");
                          foreach (GameObject
            go in ghosts)
                                {

go.GetComponent<Ghosts>().StartFrightenedMod
                e();
                                }
                              }
```

Code 1 – The part of code when Pac-Man consumed a super pill and ghosts change into frightened mode

IV. Conclusion

This task software application has to motivate the reader to perfect his programming and apply it effectively. Shaping our acquired knowledge into something applicable and creative can contribute to beginners in object-oriented programming to develop their logic and look at things in perspective. So that each of their newly created software applications has the opportunity to be further developed and perfected.

The PacMan software application can be refined and improved with a few tweaks. First of all, in order to attract new users of the application, it is necessary to add a new scene in which each user would enter their data and record their own result. This data would be stored in a specific database and could be shared on social networks.

When the PacMan software application gets its final look, it is necessary to surprise users with some thematic look. If the holidays are followed in the place and surroundings where this software application is represented, it is possible to create a menu and heroes in that spirit. In that way, we get the possibility of constant renewal.

Programming as a wide range can play a key role in the development of our logic, but also the imagination itself. Through the creation of such a project, in addition to the development of the mentioned items, we can conclude that the application of mathematics itself is crucial in the process of programming and creating some new ideas.

Acknowledgment

Ministry of Science and Technological Development, Republic of Serbia financially support this research, under the project number TR32044 ”The development of software tools for business process analysis and improvement”.

References

[1] Laslo Kraus, ‘‘Programski jezik C++: sa rešenim zadacima’’, Akademska misao, Beograd 2000.

[2] J. Price, ‘‘Visual C#.Net’’, Kompjuter biblioteka, Beograd 2003.

[3] H. Ferrone ‘‘Learning C# by Developing Games with Unity 2019’’, Packt, Unitet States 2019.

[4] A. Tanenbaum, „Arhitektura i organizacija računara’’, Mikroknjiga, Beograd 2017.

[5] S.Foote, „Naučite programiranje’’, Računarski fakultet i CET, Beograd 2015.