PROGRAMMING DONKEY KONG REPORT

ANDER LÓPEZ GALARRAGA & PAULA SERNA

Groups 96/97

**GUIDE:**

1.CLASS DESIGN

2.MOST RELEVANT FIELDS AND METHODS

3.WORK PERFORMED

4.CONCLUSION

5.PERSONAL COMMENTS AND ERRORS

1. **CLASS DESIGN.**

We have created 7 classes to achieve a better project organisation for every part of the game. The most important classes are:

**GAME:**

The most important part of the program. It has 2 parts de update one where we include all the movement because it update every frame. And the draw one where it print on the screen what the different elements do in the update part.

**MARIO:**

We included in here the position of the object Mario, in order to make him move depending if we press the right or the left arrow. As well, to control the jump of Mario we also include a parameter that lets us know if Mario is jumping or not. This will be used to jump over the barrels. Also, we used 2 other properties called MoveRight and MoveLeft to make better the appearance of Mario moving. Also 2 properties that controls the times that Mario dies.

**BARRELS:**

It follows the same structure of Mario but in this case it doesn’t follow the directions if we click the arrows, it follows a guided way. Because of that we implement the moveRight, … It will be used in order to make the move of the barrels and also to delete if the barrels get to the end of the last platform.

**CONSTANTS:**

To make the code nicer, we stabilised different parameters that will never change and will be used so many times

The rest of the classes follows the same structure as the previous ones, principally, to storage the different positions.

1. **MOST RELEVANT FIELDS AND METHODS.**

The most relevant methods are related to the movement of the sprites. Related to Mario, is the one who has more conditions and restrictions because the player can do so many things. With the 2 arrows he can move left or right, with the up arrow he can go up through the ladders and with the space he can jump. For that we create:

* *move()* : that controls basically the arrows movement, if go up, down, left or right by adding or subtracting to positions x and y of the object of Mario
* *jump ()*: that controls where Mario is jumping to. If jumps to the right, we must add 15 to posx and if jumps to the left we need to subtract it for example.
* *desjump ()*: for bring back to the floor Mario. This is so important because we need to play with the counter frame in order to see and jump properly because if we don’t do this then it will go so fast that wouldn’t look like a jump.
* *Descend ()*: Mario can fall from the platforms if he arrives to the end of it so this method let us to make him descend in every platform.
* *Remove ()*: If the position of Mario is so close to a barrel, he loses a live and starts again so this function return Mario to the start.
* Movement of the barrels: we use 3 different methods to control the move of the barrels, first if goes to the right or left, then if it arrives to the end of a platform is necessary that it fall down and also if the barrel is over a ladder, it will have a 25 % to go down thought the ladder.

**3. WORK PERFORMED**

The most important thing we did was calling the objects, refering to their clases (objects for Mario, the platforms, the ladders, the barrels… each one with its class), and gibing them an initial position. Some of them , such as the platforms or ladders are static.

For a better organisation, we put all our objects’ positions and values in a ‘constants’ file, such as the height and weight of the board andthe static objects’ positions(x and y).

The essential functions in our program are the update and draw ones. Instead of creation one of them for each class, we created two global ones in the main program (calling the clases, obviously) , and run everything inside them.

* -In **update**:
* First of all, we assigned to some keys of the keyboard some ‘functions’ refering to our classes. The “right”, “left”, “up” , “down”, “space” and the “Q” one to go out of the game. We allowed Mario to move over the platforms refering to the constants file where all our positions are. Mario is able to move freely over them and also to descend from them where there is an empty space, taking into account the distance between each pair of platforms, as this value changes ; and with a specific function we created in Marios’ class.
* After this, we allowed Mario’s object to go up and down the ladders establishing a range, as it is very difficult to put the exact position of the ladders.
* For Mario’s jump, we introduced a global counter which was related with the number of frames that were going to pass in each jump. One jump while pressing the key “right” and another one with the “left” one.
* Then, came one of the most difficult parts, creating the barrels. We created another variable called “counter” and we stablished that in a specfied interval of time, the barrel object were going to be created and added to a list until 10 barrels were active at the same moment. Then, to move them, we just refered to the platforms’ positions and calling the functions ‘moveRight’, ‘moveLeft’ and ‘down’, and when they reach the initial point of Mario, they dissapear.
* After lots of efforts, we managed to kill Mario when he was touched by a barrel, taking into account their hights and widths, this calculations were very important.
* -In **draw**:
* Here, we draw all the sprites. The most difficult part was to anímate some of them, such us Mario, because we had to create more variables such us “Saltando”, “Subiendo”, etc, so that we played with its images depending on what he was doing , using pyxel editor.
* We also managed to put that the player won if he reached the princess.

**4.CONCLUSION:**

During this project we have learnt a lot about the different ways to solve a problem. It lets us develop our capacities of thinking about the ways that we can address some problems. We used so many functions that we learnt in class principally the if, for and while loops.

It wasn’t is to achieve all the goals of this project, there were things that are pendent to do but it was because we didn’t know how to solve it even though we know the way.

We work as a team, trying to solve those problems, sometimes someone of us knows how to solve it and explains it to the other one and all the way around. This helps us a lot because for sure some problems wouldn’t be able to go through if we divided the work.

**5.PERSONALS COMMENTS**

First, it was so helpful the classes where teachers solve our questions and as an experienced people, gives us another way to see the problem and also to solve it. Also it was helpful because it was so easy to be confused and lost and if we didn’t have a good start we would be able to finish it.

For other years I think that maybe 2 more weeks preparing and learning more about pyxel would be better because we spent to many weeks learning how to program in python but in a few weeks we were given so much information about pyxel and I think that we weren’t enough prepare to do this project perfectly. And in our case, we are frustrating because there are some problems and we know how to solve it but we don’t have the knowledge to go through it, we just have the idea.

**-NON SATISFACTORY POINTS:**

However, we tried to perform some things but couldn’t, unfortunately. Some mistakes this project contains are: we did not manage to animate donkey Kong, even thogh we created an extra variable when the barrils were added to the list so that Donkey could change its image when barrels were being created….;we wanted to eliminate a life from the board when Mario died; we also tried to sum 100 points when Mario jumped over a barril, but this was very difficult because of the distances between the objects; we couldn’t eliminate all the barrels when the player wins, etc.

We know it is not perfect, but we did our best.