Programming Project Report

LEMMINGS

By Carlos Iborra and Rafael Contasti
Group 89 – Computer Science

Abstract

Our program is made thinking into mirroring one of the most popular videogame classics: Lemmings.
For this project, we focused on a user-friendly appearance, easy controls, and a simple and organized code, which helped us to change and improve our own game through its development.
We also saved all the versions and steps in the lemming's development, that way we could look back into previous programs if we had any doubt and to ameliorate our code.

Table of contents:

1. Classes design.

We have created 9 different classes to divide the program into the most important parts and to have it well organized for us to edit it later on. This part was one of the most difficult challenges as it was the basis of the program and in some way the skeleton of it, so, we needed to choose wisely each class and the number of classes we wanted to create.
In order to achieve that organization, we have made use of a bunch of different object-oriented programming techniques.

- **Platforms.**
  One of the pillars of this game was one of the first things to be created and it is thought to be randomly generated with different lengths and different x and y positions, which makes this game a little bit more complicated and original.

- **Gates.**
  These are the entry and exit doors, they are also randomly generated above one of the seven platforms.
  From the entry gate, the lemmings are summoned and the goal is to make the lemmings reach the exit gate by using the game tools.

- **Lemmings.**
  They are the basis of this game, they are on the whole, the character of the game and the object manipulated with tools so that they can get to the exit gate and be saved. They are generated in a random number between 10 and 20 every game always facing right first. Lemming's movement is limited to walking right and left and falling while not using tools.

- **Cursor.**
  It is an element of the game that you can move along the board and use as a guide to place the tools, it is moved with the arrow keys.

- **Umbrella.**
  It is a tool used by lemmings to fall and descend from one platform to another without dying. It is placed with the "W" key and grabbed by a lemming when it reaches the umbrella's position.

- **BlockerSign.**
  Placed with the "E" key, it is a sign to represent the position of a blocker until a lemming reaches it and becomes a blocker itself. Once this happens it is removed.
  A blocker acts as a collider and when the other lemmings collide with it, they change their direction.

- **Stairs.**

  It is the tool used by lemmings to climb and go up to other platforms above the one they are. We can place ladders in both rights of the program, as well as executing the main pyxel module for the game and drawing all the components. It is run automatically by pressing the enter key after the main welcoming screen.

- **Game.**

  It is the most important class. The game class contains most of the logic and functions of the program, as well as executing the main pyxel module for the game and drawing all the components. It is run automatically by pressing the enter key after the main welcoming screen.

- **PreGameHUD.**

  A class designed to run a welcoming message on the screen before the game starts and show the title of the game. It is closed after pressing the enter key and it automatically runs the Game class to start the game.

2. Versions.

An important part of our project is the different versions we used in it; we have undergone over 7 versions that differ a lot one from the other. It was very useful for us to save every change we made into different programs in case we got an error and we needed to fix it or to go back in our previous day's code to start again from a well-built solid base.

The versions are divided into 5 different big versions and some other smaller changes, the big versions (which are placed as the first number after the V) refer to the different Sprints written in the "Final Project: Lemmings "file uploaded to the Aula Global. For each big version, we focused on the corresponding Sprint, for example, in v.1.0. we focused on "Sprint 1: Objects and graphical interface" and so on. By making the project in that way, we made sure that we were following the right path.

- **v.1.0.**
  In this first version, the game was very basic, we used different sprites that were not as good as the new ones, and the program just consisted of three fixed platforms and a lemming that walks, falls, and bounds, we also created the top part of the game which showed the level in which you are, the lemmings alive, the saved lemmings, the dead lemmings, the number of ladders placed, the number of umbrellas placed, and the number of blockers used, but it was just the text and did not contain any information. In this version, the game was in a 15x15 grid.

- **v.2.0.**
  In this second version, the gate sprites were created and placed randomly, in this version we added the seven platforms instead of the three platforms used in the first version, and we also randomized the position where the platforms were placed. In this new version, the grid was changed from 15x15 to 16x16 and therefore all the movements and physics were changed.

- **v.3.0.**
  In this third version, we changed the sprites we did in .png to completely new and better-quality sprites in .pyxres. In that way, we could create more sprites and to make some of them moving sprites, which added a better look to the game. We also made sprites for the cursor and we eliminated the lines that appeared showing the grid lines in the screen. Finally, we added a start introduction to the game and we made the cursor work and move around the screen.

- **v.4.0.**
  In this version, we added the umbrella class and its corresponding sprites as well as the effects it has in the game physics when a lemming goes through it.

- **v.4.1.**

  In this version, we changed the Game Class which was made of lists into a Game Class that contained objects instead of lists. This helped us to organize better the code and to make it a more efficient code.

- **v.4.2.**

  In this version, we added the ladders and the ladder classes as well as their corresponding sprites and effects to the game.

- **v.5.0.**

  In this fifth version, we separated all classes into different files, except for the tool that they are all in the same file. Next, we added background music, which had been created by ourselves, we fixed minor errors and bugs and we corrected some of the comments we had done along with the program.

3. Performed work.

We handled to perform all the steps following all the rules step by step that were asked in the project. We wanted to keep the code simple and legible in order to be very easy to follow the path of the program for us to be easy to orientate and to be possible to change between versions without any struggle.

First, it appears the introduction with the title and our names, and then, when the ENTER is pressed, the program is started and the platforms randomized, level 1, the lemmings start summoning until all the lemming numbers are spawned, meanwhile, you are able to move the cursor with the keyboard arrows, and whenever you are in the desirable place, you can press the letter "W" to place an umbrella, "A" to place a left-oriented ladder, "S" to place a right-oriented ladder and "E" to transform a usual lemming into a blocker(all the lemmings that collide with it, change its direction) and whenever you want to close the program, you just need to press the "Q" letter.
After the first level, if you have passed it, a second level is going to be randomly created and so on until level 10. On the other side, if you don't manage to pass a level, you will start again that level until you pass it (this is due to the complex levels sometimes created as they are completely random).

If you are lucky enough to pass all the levels, the final part is showed, showing you that you have completed the game and the numbers of tries it took you to make it.

While on the game, at the top of the game screen, it is showing the level in which you are, the lemmings alive, the saved lemmings, the dead lemmings, the number of ladders placed, the number of umbrellas placed, and the number of blockers used.

It goes without saying that the keys chosen for the tools and quit have being pondered in order to be easy to remember and to associate each letter to the corresponding tool, and not only that but also have been chosen because they are some of the most used letters in gaming, so, gamers are going to feel completely comfortable with our key distribution.

If you play the game, you are going to notice instantly the time and the effort we have put into every sprite, we have animated the lemming sprite, the umbrella, the lemming with the umbrella, the cursor (in order to see it better while playing) and even the dead lemming. Moreover, you will notice that the sprites are original and it is our remastered version of the first lemming game.
Also, if you open the pixel editor you will observe that there are more sprites than the ones used in the program, and this is because we have the intention to upgrade the game and add more things and objects in the future.

To finish, another noticeable feature is the background music we have added in the game, which we have created ourselves as well as the already-mentioned sprites.

4. Conclusion.

During this project, we both have learned a lot about programming in python, especially the creation of classes and objects and how to make games, and the steps to follow.

To be honest, this hasn't been easy, we both live very far from each other and as programming through the internet is quite different and complicated if you need to use pyxel, we needed to meet and program together, so we did that and we met to program this exercise every weekend until we finished.

But the most difficult part that really delayed us through this project was when we needed to make and decide which were going to be our classes, what to do first and all the little bugs and problem everybody encounters through a programming process, some of the bugs even took us weeks to solve because the hard it was to encounter the problem in such a long program.

For us, the most important part was creating different versions for each time we programmed something new, that way we made sure everything worked fine, to correct the newest code, and to help us to develop the incoming ones, and to make sure we were going the right way.
Also, a page that was very helpful was the one created in the official pypi.org, where they explained in detail how pyxel worked and some program examples they did with python and pyxel.

This gave us other perspectives when programming and to make us more acquainted with the real programming process and the different problems we are going to meet and how to solve them.

5. Personal comments.

As I mentioned before, we found very useful the pypi.org page of pyxel 1.4.3 which was by far the best, with a lot of explanations and the most detailed we have seen among all the other ones. Also, it contained some game examples which were pretty useful.

Another page we found was the blog of the creator of pyxel, named “Kitao’s Blog” where he talked about how pyxel worked and made an introduction to it. But if you don’t know Japanese, you must translate the webpage as it is completely in Japanese.

A thing we noticed was that it existed another module of python oriented to games creation different from Pyxel, which is called Pygame, and we though it could be better, because, for this module, there existed way more pages talking about it and a lot of tutorials and information that could have been very useful to know, therefore we though that it could be interesting to look into this module for the next years, as sometimes pyxel became slow compared with other modules or languages, so maybe Pygame could be better for programming a video-game.

To summarize, this helped us to get used to programming, creating programs and to solve common mistakes and problems when programming, which is certainly very useful for our future as programmers, and if you were wondering, of course, it was a great adventure and we have enjoyed every part of the creation of this game.