LAB PROJECT

Computer Structure.Game:Blocky

Juan Perea Campos

José Antonio Martínez López

Francisco López Navarro

Rubén Pérez Rubio

Pablo Moraga Navas

Group: G1. Team: “Blocky”

19TH OF MAY OF 2014

User's guide:

Instructions to play:

- When we switch the NDS, in the tactile screen will appear a main screen with a message "PLAY GAME". To start the game we will just have to press on top of it.
- Once the game has started, the goal of it consists of arriving to the green block by the movement of the sprite. This movement will be done with the direction arrows from the pad.

- To reach up to the green zone we will have to keep crashing against the yellow blocks. Once a direction key has been pressed, the sprite will not stop moving until hitting against one of the yellow blocks. It will also be impossible to change direction while the sprite is moving.

- Once we have reached the green zone, you will have passed the level and a new screen with the option of selecting the next one will appear. For that, it will be necessary to press the message or button "NEXT".

- In the case we hit against the limits of the screen, you will lose automatically and a screen for retrying the level will appear. For this, we just have to press the message "RETRY".

- There will be a timer, which will count in seconds the progress of the game.

- The game counts with 3 levels in total and once we have finished all of them you will go back to the main screen after pressing a message that shows up greeting your achievement.

Theoretcal concepts

We have applied many different theoretical concepts while creating our game for the NDS. Some of them, the ones we have considered to be important, we detail them here briefly indicating as an example the code employed for it.

Input and Output devices (I/O)

- <u>Tactile screen</u>: we use it as an input (to touch in the screen) as well as an output to show images. For reading if the tactile screen is pressed or not, we use the function *touchRead(&touch)* and we establish the coordinates in which this function will wait the press or not of the screen called to the methods *touch.px* and *touch.py*.

- <u>Timer</u>: the timer is used to detect an interruption that the clock of the console generates every x time and calls a method(timeout) that increases a counter. This is the code used to call the increasing method mentioned before: *irqSet(IRQ_TIMER0, &timeout)*.

- Sound: after having transformed the sound file to the .raw format with a maximum of 256 kb, we load the file with *NF_LoadRawSound("sfx/crash", 0, 13025, 1)* and simply for reproducing it we use *NF_PlayRawSound(1, 127, 64, true, 0)* being the number 1 a sound identifier so we can implement several sounds. All of this is done after enabling the sound with *soundEnable() and loading the sound buffers with NF_InitRawSoundBuffers()*.

- PAD or buttons: we use tha PAD to control the movement of the sprite. We use the function *scanKeys()* to detect whenever any of the direction keys or buttons is pressed, and then we save it in a variable *keys = keysHeld()*. We control with an *if* if the variable *keys* is activated, or what is a synonym, if any key from the PAD has been pressed. If so, and if we detect that a key has been pressed in the direction we want (*if (keys **& KEY_UP**)*), we enter inside the *if* so it can execute the part of the corresponding code with the finality of the sprite moving in the specified direction in the second condition of the *if.*

  We control the buttons used to mute the music in the same way but controlling it with another variable different from the other one, called in this case *newpress.* We control with an *if* and if the conditions are fulfilled we call the functions to mute the music or to restart it and so it can be played again from the beginning, *soundPause(sound_id)* y *soundResume(sound_id)* respectively.

- Top Screen: we use it to show images as well as we do with the tactile screen. Some of the commands we use to treat with the background images are:

  - *NF_InitTiledBgBuffers()* – Initializes the buffers to store background images.
  - *NF_InitTiledBgSys(0)* – Initializes the tiled backgrounds for the top screen.
  - *NF_LoadTiledBg("bg/bg1", "bg1", 256, 256)* - Loads the background in the top screen.
  - *NF_CreateTiledBg(0, 3, "bg1")* – Creates the backgrounds for the tactile screen.