

TECHNICAL MANUAL

Poñyu

Effective May 15, 2019

Sarah Huang: Programming, music, images

Ceagan Nolan: Programming, sprite work

**Table of Contents**

Overview………………………………………………..……………………………………2

Development Environment………………………………………………………..….2

Programming Language………………………………………………………………..2

Definitions of Variables………………………………………………………………...2

Descriptions of Functions/Classes…………………………………………….…..5

Maintaining the Program……………………………………………………………...6

**Overview**

A button is used to start the game. Poñyu’s game code has a constant while loop to check for key presses on the keyboard to signal Poñyu’s movement and update graphics when certain events happen like defeating an enemy or getting hit by one. Once the variable score reaches its max of 250 for stage one, the enemies disappear. Speech bubbles are then displayed on screen before the boss battle. Afterwards, the boss appears and will disappear after 100 hits.

**Development Environment and Programming Language**

Poñyu was developed in PyGame using PyCharm. We chose PyGame for its excellence in displaying graphics, overall flexibility, and ability to create abstract games. PyCharm was our chosen SDE because we have experience in using it.

**Definitions of Variables**

* Canvas\_width: The width of the game window
* Canvas\_height: The height of the game window
* Window: The surface of the game window and takes the parameters canvas\_width and canvas\_height
* Bg: Loads the background image
* Bg\_x: Sets the x-coordinate of the original background to zero
* Bg\_x2: Sets the x-coordinate of the second background to the width of the background to make a looping background
* Clock: Creates an object to help track time
* Slime\_attack\_sound : Loads the sound effect for when the player pushes spacebar
* Hit\_sound: Loads the sound effect for when an enemy is hit
* Craneo\_fuego\_defeated\_sound: Loads the sound effect for when an enemy is defeated
* Boss\_defeated\_sound: Loads the sound effect for when the boss is defeated
* Walk\_right: Stores index of images that animate walking right
* Walk\_left: Stores index of images that animate walking left
* Self.x: The object’s x coordinate position
* Self.y: The object’s y coordinate position
* Self.width: The object’s width
* Self.height: The object’s height
* Self.vel: The object’s speed
* Self.is\_jump: Whether or not the object is jumping
* Self.jump\_count: Helps determine how high Ponyu jumps
* Self.left: Whether or not the object is going left
* Self.right: Whether or not the object is going right
* Self.walk\_count: Helps determine which image in an index is displayed
* Self.idle: Stores whether or not the player is not moving **(NOT USED)**
* Self.standing: Whether or not the object is standing still
* Self.hitbox: Sets up the object’s hitbox
* Craneo\_fuego.x: The x-coordinate of the enemy
* Craneo\_fuego\_dos.x: The x-coordinate of the second enemy
* Font\_death: Setting up the font of the death screen
* Text\_death: Renders the text message
* Text\_back: Renders the text message a second time behind the original to create an artistic effect
* Slimeball: Loading the sprite of the slimeball projectile
* Slimeball\_reverse: Loading the sprite of the slimeball projectile going the other direction
* Self.facing: Whether or not Poñyu is facing left to determine the direction of the slimeball
* Self.end: The position where the enemy will stop moving at
* Self.path: The enemy will travel from a certain starting position to a certain ending position
* Self.health: Sets the numerical health value of the enemy
* Self.visible: Whether or not the enemy is still on screen
* Boss\_sprite: The index of boss sprites
* Self.random\_shot: Randomized position of the boss
* Self.timecount: Stores the time for how long the boss waits before striking again **(NOT USED)**
* Self.color: Stores color of a button
* Self.text: Stores text of the button
* Font\_button: Sets font of the button
* Text: Renders the object’s text and color
* Self.image: Stores the image of an object
* Cinnabon\_delights\_text: Renders the numerical score and the color
* Cinnabon\_delights\_text\_back: Renders a copy of the numerical score with a different color to create artistic effect
* Run: Stores whether the game is running or not
* Start: Stores whether the game is on the title screen or not
* Shootloop: Stores the number of slimeballs shot
* Cinnabon\_delights: Stores the numerical score
* Bullets: Stores an empty list
* Start\_button: Stores a created Button object
* Cinnabon\_delights\_font: Sets the font of the Cinnabon Delights score
* Ponyu: Stores a created Ponyu object
* Craneo\_fuego: Stores a created Craneo Fuego object
* Craneo\_fuego\_dos: Stores a created second Craneo Fuego object
* Boss: Stores a created Boss object
* Boss\_dialogue1: Loads the first speech bubble
* Boss\_dialogue2: Loads the second speech bubble
* Boss\_dialogue3: Loads the third speech bubble
* Show\_boss\_dialogue1: Stores a created Dialogue object with the first speech bubble
* Show\_boss\_dialogue2: Stores a created Dialogue object with the second speech bubble
* Show\_boss\_dialogue3: Stores a created Dialogue object with the third speech bubble
* Title\_screen: Loads the image for the title screen
* Continue\_program: Whether or not the first level is done  **(NOT USED)**
* End\_screen: Loads the image for the game end screen

**Descriptions of Functions and Classes**

* Player class: Establishes the position, size, movement status, and hitbox of the player’s sprite, Ponyu
* Projectile class: Establishes the position and the direction the slimeball is shooting
* Enemy class: Establish the position, size, travel path, speed, health, hitbox, and visibility of the enemies, Craneo Fuegos
* Boss class: Establish the position, size, travel path, speed, health, hitbox, and visibility of the boss, Cinnabon Incinerator
* Button class: Establishes the position, size, choice of text, and color of the start button
* Dialogue class: Establishes the position, visibility, and choice of image of the three speech bubbles
* Def \_\_init\_\_(self…): Stores the given parameters into different variables for all classes
* Def draw(self, window): Displays a hidden hitbox, health bar, and a sprite depending on the index for Ponyu, Projectile, Enemy, and Boss
  + For Button class, the draw function draws an ellipse and puts text on it
* Def isOver(self, mousePosition): Returns a boolean to determine whether the button will change color depending if the mouse if over it
* Def appear(self): Displays the speech bubble image for a fixed amount of time
* Def redrawGameWindow(): Displays the updated background, score and sprites, Ponyu and Craneo Fuegos, on screen for the first level
* Def redrawGameWindow\_end(): Displays the updated background, score, and sprites, Ponyu and Cinnabon Incinerator, on screen for the boss battle
* Def hit(self): Displays text when player touches an enemy (Ponyu) or decreases an enemy’s health when hit by slimeballs (Enemy, Boss)
* Def move(self): Determines the movement of the enemy from its starting point to its end point

**Maintaining the Program**

Do not delete any files, music, sprites, or pictures. Ask for user feedback and criticism in order to focus on what next to improve or fix in the game. The game is not perfect, so continuously look for bugs and errors that can be fixed. Clean up and organize code to make it easier to read. In addition, improve code by rewriting code to be more efficient.