

Sistema de control táctil para videojuego

del dinosaurio utilizando Arduino

Carpeta de campo

Integrantes : Alma Carena , Mateo Lugo , Facundo Noriega , Santino Trevisano y Dante Bassus

Institucion : Tecnica N°1 de Vicente López

Curso : Laboratorio de Programación 5°3 A-B

Profesores: Mansilla Muñoz York E. / Ganduglia Yamil

Fecha: 00/00/2025

Índice

Carpeta de campo

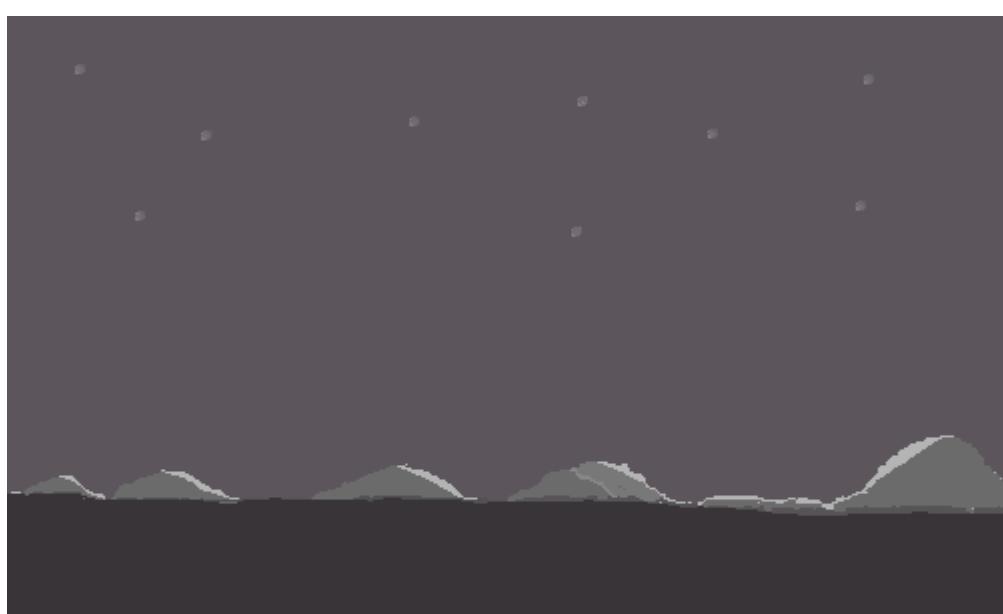
4/9 y 10/9

El primer día del proyecto pensamos el juego, los roles que íbamos a ocupar y tuvimos problemas con la organización del curso y de los profesores involucrados. También anotamos y creamos el repositorio y las bases del código

El segundo día del proyecto empezamos con las primeras líneas del código, el informe lo empezamos y aplicamos el formato APA, buscamos los materiales que íbamos a buscar y él definió el proyecto como una experiencia con un control hecho con un arduino.

17/09 y 18/09

Fuimos probando y haciendo los primeros frames junto al diseño de la experiencia y creamos la carpeta del campo. A continuación unas capturas, Estas capturas estan separadas dado al funcionamiento del codigo

Capturas:

Código:

```

import pygame
import random
import sys

pygame.init()
pygame.mixer.init()

ANCHO = 400
ALTO = 300
FPS = 60

ventana = pygame.display.set_mode((ANCHO, ALTO))
pygame.display.set_caption("Dinosaurio con 2 imágenes")

# Colores
BLANCO = (255, 255, 255)
NEGRO = (0, 0, 0) # Fondo negro

# Fuente
fuente = pygame.font.SysFont("Arial", 28)

# CARGA DE IMÁGENES
try:
    dino_img = pygame.image.load("imagenes/dino.PNG").convert_alpha()
    cactus_img = pygame.image.load("imagenes/cactus1.jpg").convert_alpha()
except pygame.error as e:
    print("Error al cargar imágenes:", e)
    sys.exit()

# Escalar imágenes
dino_img = pygame.transform.scale(dino_img, (60, 60))
cactus_img = pygame.transform.scale(cactus_img, (40, 60))

# Dinosaurio
dino_x = 50
dino_y = ALTO - 60 - dino_img.get_height()
dino_vel_y = 0
gravedad = 1.2
en_suelo = True

# Obstáculos
obstaculos = []
velocidad_obstaculos = 6
tiempo_ultimo_obstaculo = 0
tiempo_entre_obstaculos = 1500 # ms

# Puntuación
puntaje = 0
record = 0
juego_activo = True

# Reloj
clock = pygame.time.Clock()

# FUNCIONES
def mostrar_texto(texto, x, y, color=BLANCO, tamaño=28):
    fuente_local = pygame.font.SysFont("Arial", tamaño)
    render = fuente_local.render(texto, True, color)
    ventana.blit(render, (x, y))

def reiniciar_juego():
    global dino_y, dino_vel_y, en_suelo, obstaculos, puntaje, juego_activo
    dino_y = ALTO - 60 - dino_img.get_height()
    dino_vel_y = 0
    en_suelo = True
    obstaculos.clear()
    puntaje = 0
    juego_activo = True

def crear_obstaculo():
    rect = cactus_img.get_rect()
    rect.x = ANCHO + random.randint(0, 100)
    rect.y = ALTO - 60 - rect.height
    return rect

# BUCLE PRINCIPAL
while True:
    dt = clock.tick(FPS)

    ventana.fill(NEGRO) # Fondo negro

    # EVENTOS
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

        if juego_activo:
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_SPACE and en_suelo:
                    dino_vel_y = -18
                    en_suelo = False
                else:
                    if event.type == pygame.KEYDOWN and event.key ==
                        pygame.K_SPACE:

```

```

        reiniciar_juego()

if juego_activo:

    # MOVIMIENTO DEL DINOSAURIO
    dino_vel_y += gravedad
    dino_y += dino_vel_y

    if dino_y >= ALTO - 60 - dino_img.get_height():
        dino_y = ALTO - 60 - dino_img.get_height()
        en_suelo = True

    # GENERAR OBSTÁCULOS
    tiempo_actual = pygame.time.get_ticks()
    if tiempo_actual - tiempo_ultimo_obstaculo >
tiempo_entre_obstaculos:
        obstaculos.append(crear_obstaculo())
        tiempo_ultimo_obstaculo = tiempo_actual

    # Mover y dibujar obstáculos
    for obstaculo in obstaculos:
        obstaculo.x -= velocidad_obstaculos
        ventana.blit(cactus_img, obstaculo)

    # Eliminar obstáculos que salen
    obstaculos = [o for o in obstaculos if o.x + o.width > 0]

    # DETECCIÓN DE COLISIONES
    dino_rect = pygame.Rect(dino_x, dino_y, dino_img.get_width(),
dino_img.get_height())
    for obstaculo in obstaculos:
        if dino_rect.colliderect(obstaculo):
            juego_activo = False
            if puntaje > record:
                record = int(puntaje)

    # ACTUALIZAR PUNTUACIÓN
    puntaje += 0.1

    # DIBUJAR DINOSAURIO

```

```
ventana.blit(dino_img, (dino_x, dino_y))

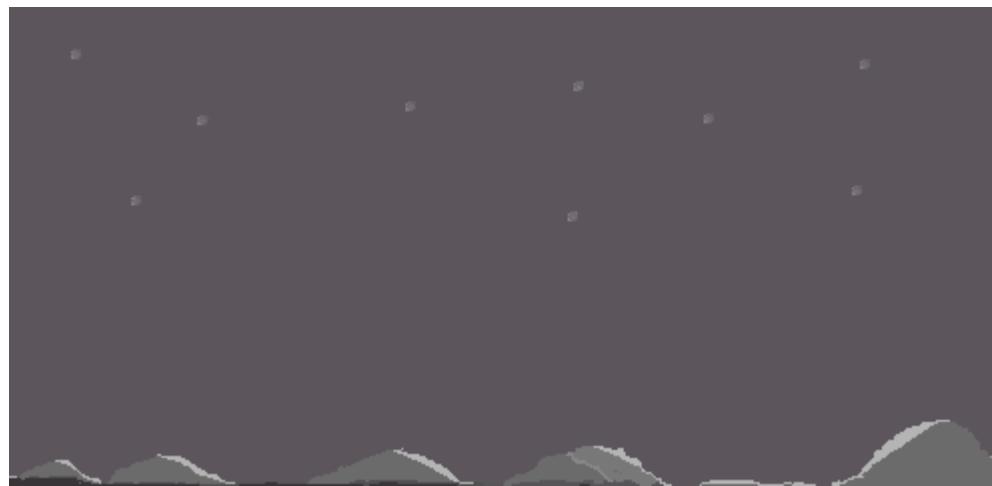
# HUD
mostrar_texto(f"Puntos: {int(puntaje)}", 10, 10)
mostrar_texto(f"Record: {record}", ANCHO - 150, 10)

else:
    # Pantalla de Game Over
    mostrar_texto("GAME OVER", ANCHO // 2 - 80, ALTO // 2 - 30,
BLANCO, 36) # Texto blanco
    mostrar_texto("Presiona ESPACIO para reiniciar", ANCHO // 2 -
160, ALTO // 2 + 20, BLANCO) # Texto blanco

pygame.display.flip()
```

23/09 (casa)

-Terminamos de diseñar algunos sprites y frames, preparamos la primera versión estable del juego v1.0.0 y creamos la presentación.



24/09 y 25/09

Introducción

Esta semana presentamos, vimos las presentaciones de nuestros compañeros y avanzamos un poco en todo.

Desarrollo

-Planteamos nuestras ideas como un ranking de puntos conectado a un base de datos, nuevas mecánicas como el pajarito y empezar ya con el Arduino y el manual de usuario

Problemáticas surgidas

-No sabemos cuando empezamos con el manual y el arduino.

01/10 y 02/10

Introducción

Está semana vimos las presentaciones de nuestros compañeros que quedaban y empezamos a hacer los cambios y prototipos, diseñamos nuevos sprites, implementamos mejoras y vimos tema BD con los puntajes.

Desarrollo

-Creamos nuevos sprites para el próximo jefe(no tiene nombre todavía), cambiamos el readme, actualizamos el README, buscamos formas para resolver el tema de los commits, implementamos los sprites del jefe cuando perdes y empezamos a ver tema BD y su implementación. Arreglamos el tema de la generación del obstáculo inesquivable de los cactus y el pájaro, creamos el nuevo mundo.

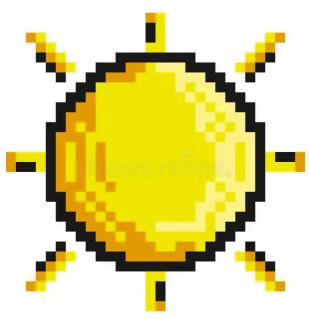
Problemáticas surgidas

-No pudimos pensar bien los sprites, estamos probando tema bases de datos pero el internet del colegio se cae cada 2x3, hay problemas temas commits ya que todos trabajamos y comítmamos con cosas que cometió el compañero y queda mal. Tema generación cactus y del ave, si se generan al mismo tiempo se crea un obstáculo imposible de esquivar.

Movimos el tema de la BD a lo ultimo ya que hay que testear el juego con el XAMPP abierto, prendido y con la BD puesta, lo que requiere mas organización y recursos.

Capturas:





Código(prueba):

Api_client.py

```

import requests
import json

# URL donde está corriendo tu servidor. Si es en el mismo PC, esta es la correcta.
SERVER_URL = "http://192.168.1.50:5000"

def cargar_puntajes_online():
    """Obtiene los mejores puntajes desde el servidor."""
    try:
        response = requests.get(f"{SERVER_URL}/get_scores")
        response.raise_for_status()
        return response.json()
    except requests.exceptions.RequestException as e:
        print(f"Error al conectar con el servidor: {e}")
        return []

def registrar_puntaje_online(nombre, puntaje):
    """Envía un nuevo puntaje al servidor."""
    payload = {"player_name": nombre, "score": int(puntaje)}
    headers = {"Content-Type": "application/json"}
    try:
        response = requests.post(f"{SERVER_URL}/add_score",
data=json.dumps(payload), headers=headers)
        response.raise_for_status()
        print("Puntaje enviado exitosamente.")
        return True
    except requests.exceptions.RequestException as e:
        print(f"Error al enviar el puntaje: {e}")
        return False

def obtener_record_online():
    """Obtiene el puntaje más alto de todos para mostrarlo como récord."""
    puntajes = cargar_puntajes_online()

```

```

if puntajes:
    return puntajes[0].get('max_score', 0)
return 0

```

server.py

```

from flask import Flask, request, jsonify
from flask_cors import CORS
import mysql.connector

# --- CONFIGURACIÓN ---
# REEMPLAZA ESTO CON TUS CREDENCIALES DE MYSQL
DB_CONFIG = {
    'host': 'localhost',
    'user': 'root',
    'password': 'RedesInformaticas',
    'database': 'Dinosaurio'
}

app = Flask(__name__)
CORS(app) # Permite que tu juego se conecte al servidor

def get_db_connection():
    """Crea y devuelve una conexión a la base de datos."""
    try:
        conn = mysql.connector.connect(**DB_CONFIG)
        return conn
    except mysql.connector.Error as err:
        print(f"Error de conexión a la base de datos: {err}")
        return None

@app.route('/get_scores', methods=['GET'])
def get_scores():
    """Devuelve los 10 mejores puntajes únicos por jugador."""
    conn = get_db_connection()
    if not conn:
        return jsonify({"error": "No se pudo conectar a la base de datos"}), 500

    cursor = conn.cursor(dictionary=True)
    query = """
        SELECT player_name, MAX(score) as max_score
    """

```

```

        FROM high_scores
        GROUP BY player_name
        ORDER BY max_score DESC
        LIMIT 10;
    """
    cursor.execute(query)
    scores = cursor.fetchall()
    cursor.close()
    conn.close()

    return jsonify(scores)

@app.route('/add_score', methods=['POST'])
def add_score():
    """Añade un nuevo puntaje a la base de datos."""
    data = request.json
    player_name = data.get('player_name')
    score = data.get('score')

    if not player_name or score is None:
        return jsonify({"error": "Faltan datos"}), 400

    conn = get_db_connection()
    if not conn:
        return jsonify({"error": "No se pudo conectar a la base de
datos"}), 500

    cursor = conn.cursor()
    query = "INSERT INTO high_scores (player_name, score) VALUES (%s,
%s)"
    cursor.execute(query, (player_name, score))
    conn.commit()
    cursor.close()
    conn.close()

    return jsonify({"success": "Puntaje agregado"}), 201

if __name__ == '__main__':
    # Inicia el servidor. Debe estar corriendo para que el juego
funcione.
    app.run(host='0.0.0.0', port=5000)

```

08/10 y 09/10

Introducción

En estos días vimos el tema del desarrollo de los sprites del fondo, de los otros dos personajes y nuevo menú con el pájaro arreglado.

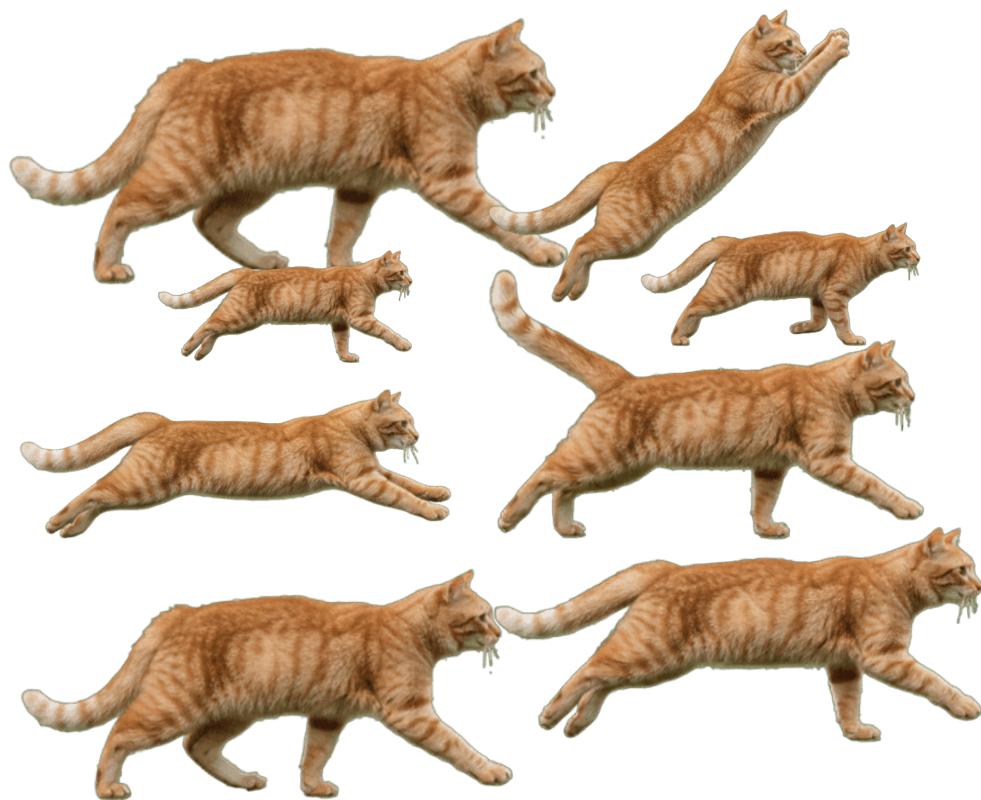
Desarrollo

Hicimos los sprites de los otros dos personajes, la rana y el gato, pusimos en el menú una sección para elegir personaje, resolvimos bugs, errores de los sprites y la calidad, escala y la posición del gato. Los marcos de diálogo lo seguimos desarrollando. Además como estamos trabajando en múltiples ramas git a la vez tuvimos que implementar todos los cambios y unificarlos para poder seguir trabajando y mejor.

Problemáticas surgidas

Tema de que la animación del gato se reproduce mal, problemas de código por la implementación masiva de nuestros diferentes códigos, tema

Capturas:



15/10 y 16/10

Introducción

En estos días expusimos los proyecto todo los grupos, hicimos el segundo avance, pensamos las cinco preguntas para los demás grupos

Grupo 1:

¿Capacidad máxima de tamaño?

¿Tipos de archivos que se puedan mandar?

¿A qué velocidad se mandan? ¿De qué depende?

¿Hay un tutorial de ejemplo de la app?

¿Tiene capacidad máxima el “lobby”?

Grupo 2:

¿El usuario puede compartir la cuenta?

¿Qué opciones te dan al crear tu propia página?

¿La estética de la página/mi sección es modificable?

¿Hay una cantidad límite de páginas que se pueden hacer por usuario?

¿Tiene modo oscuro?

¿Hay un registro de los cambios de los productos?

Grupo 4:

¿Es legal?

¿Puede cargarse con otros tipos de municipios?

¿Rango máximo de alcance?

¿Se puede modificar el alcance?

¿Qué sensores van a usar?

Grupo 5:

¿Se puede vincular a otros dispositivos ?

¿Por qué sacaron el componente ?

¿De qué materiales piensan hacer el Hardware ?

¿Para qué piensan usarla ?

¿Dónde se van a guardar los registros?

Grupo 6:

¿El fondo se puede cambiar?

¿Tienen pensado en agregar más niveles?

¿El usuario puede cambiar de personaje?

¿Se podrán desbloquear nuevos objetos?

¿Tendrá un final boss?

Grupo 7:

¿Va a desarrollarlo como un MOD del Friday funk?

¿Va a tener otras modificaciones extras como modificación de menú, interfaz, etc?

¿Va a contar con dificultades extras como algunos MODs del juego cuentan?

¿Va contar con un ranking con BD el juego?

¿Futuras mecánicas?

Desarrollo

No desarrollamos nada el primer día, vimos las presentaciones, planteamos las preguntas y agarramos ideas de los compañeros que presentaron, como lo del .exe.

Problemáticas surgidas

En la última versión no anda bien los sonidos del juego debido a problemas técnicos, su funcionamiento hace que los sonidos se ejecuten de fondo de manera indefinida pero con el volumen al 0%, cuando el jugador aprieta la tecla ESPACIO se sube este volumen al 100% pero en nuestros MP3 los sonidos tienen pequeños espacios de 1s entre el bucle, por lo que si el jugador presiona la dicha tecla puede ser que el sonido se reproduzca entró esos intervalos sin sonido.

Capturas:



22/10 y 23/10

Introducción

En estos días estuvimos exponiendo y aprendiendo sobre cómo mejorar nuestras presentaciones, aceptamos nuevas sugerencias las cuales fueran estas:

¿El juego es infinito?

¿El juego se vuelve de noche después de ciertos puntos?

¿Se puede esquivar el portal?

Cambiar la letras

¿Por qué hay un solo jefe final?

¿Cómo funciona la generación de obstáculos?

¿Tienen pensado hacer un ranking de puntajes?

¿Planean implementarlo para celular?

¿Cómo vas a derrotar el jefe?

¿Es compatible con otros sistemas operativos?

¿Los sprites están definidos? hitbox

¿Cuál va a ser el jefe final?

¿Tiene modo daltonico?

¿Se puede pausar el juego mientras jugas?

Desarrollo

Implementamos nuevas mecánicas como el DASH, el PLANEO y AGACHARSE. hicimos el ejecutable(.exe), presentamos la última versión del código, pusimos otros sprites de efecto de dash y comenzamos a programar el Arduino y el Ranking de BD.

Problemáticas

Tuvimos problemas de actualización y adaptación a la hora de implementar el Arduino y la BD, problemas otra vez con los sprites, etc.

29/10 y 30/10

Introducción

Desarrollamos un sketch para el arduino, sin embargo este no conecta con nuestro código anterior de python. Entonces estamos investigando diversas librerías para la conexión entre ambos.

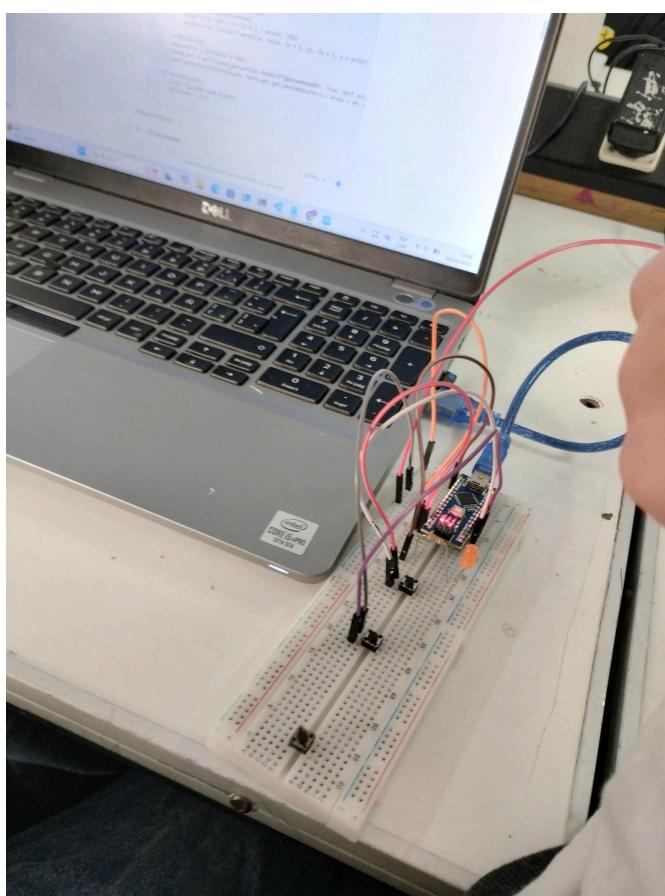
Desarrollamos e hicimos un diagrama de Gantt, seguimos trabajando en el Arduino. Actualizamos los informes, manual de usuario, etc.

Desarrollo

Diagrama de Gantt: [Link](#)

Problemáticas

Tuvimos problemas con la implementación del Arduino, generaba irregularidades de rendimiento, lag, etc.



05/11 y 06/11

Introducción

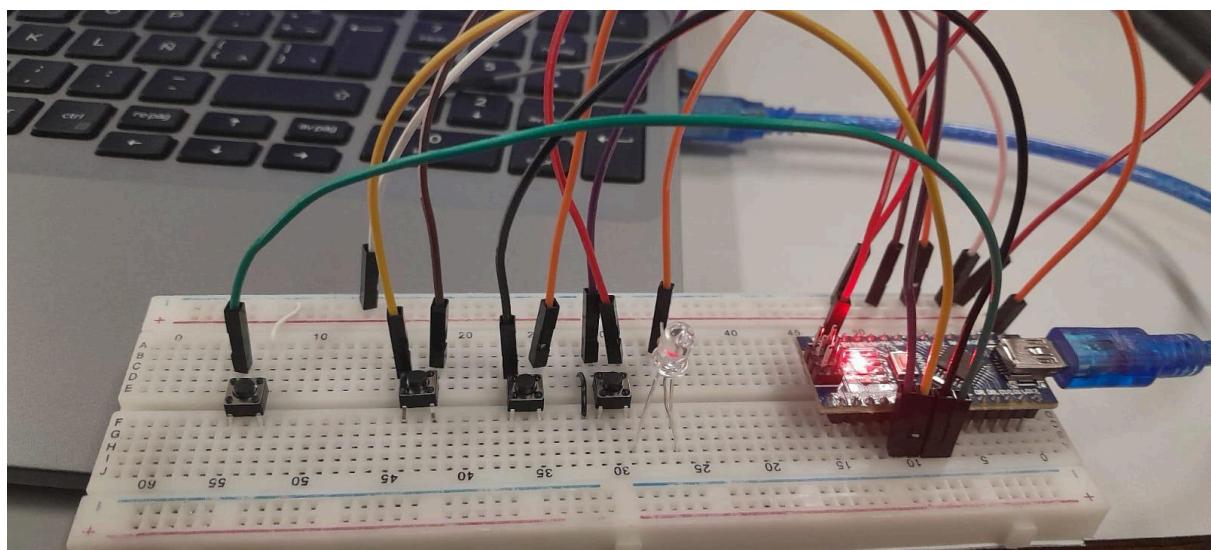
Esta semana estuvimos avanzando en el manual, puliendo detalles finales del juego y terminando de desarrollar el Arduino., tuvimos problemáticas pero las fuimos resolviendo,

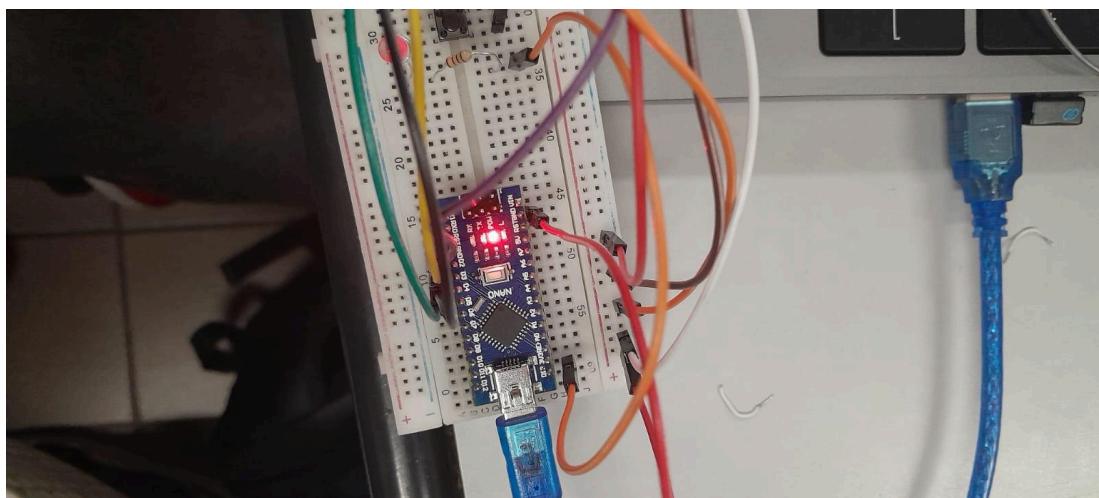
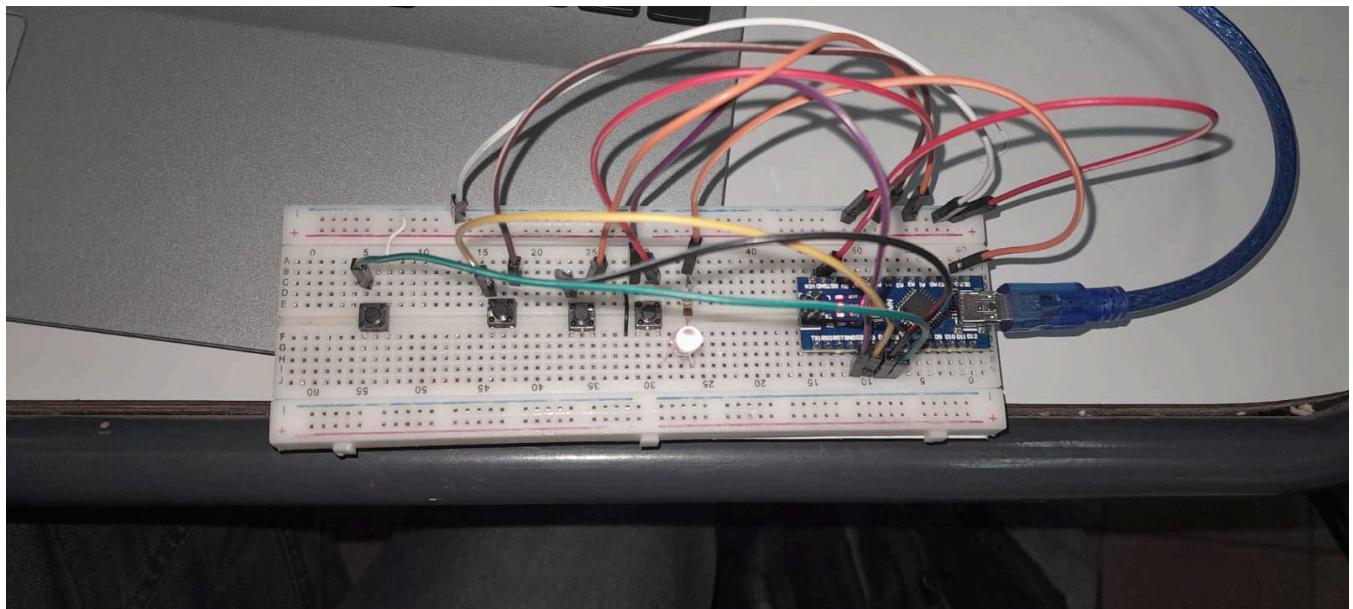
Desarrollo

Arreglamos y seguimos armando el tema del Arduino, pudimos arreglarlo e implementamos la descarga del instalador “Setup” del dino en el readme, asi que terminaremos de hacer el paso a paso del instalador. top

Problemáticas

Tuvimos problemas ya que el Arduino no respondía a una combinación de botones, respondiendo de un botón a la vez, el protoboard no transmite bien la señal y retraso de actualizaciones.





12/11 y 13/11

Introducción

Terminamos los manuales. AMBOS, el hardware del AQUILES, el código y todo en general incluido la presentación para la expotecnica.

Desarrollo

Arreglamos errores del instalador, pintamos y armamos el AQUILES con los nuevos touchs y terminamos ambos manuales los cuales están en el git.

Problemáticas

Tuvimos problemas por que no sabíamos que había que hacer el manual del programador, la mala comunicación entre los profesores y entre otras cosas habían determinado que no se hacia pero bueno lo tuvimos que hacer.

