ESNE University School of Design, Innovation and Technology

Práctica Final. Android Studio

Desarrollo para Dispositivos Móviles

Profesor: Ángel Rodríguez Ballesteros

Mariana Moreira

25/01/2020

Intro

Esta clase presenta el título del juego con un efecto de *fade in* y *fade out*. Para crear el efecto, se utiliza un temporizador y la opacidad se cambia según el estado de la escena.

Menu

Incluye todas las opciones del menú principal (Jugar, Ayuda y Créditos). Cada botón redirige al usuario a la escena especificada.

- configure_options () -> dibuja los atlas *slices* en las posiciones establecidas.
- option_at () -> comprueba qué opción se está seleccionando y devuelve su índice.

Cuando se hace clic en un botón, se llama a la siguiente línea y redirige al usuario a la escena solicitada.

- director.run_scene (shared_ptr< Scene > (new Game_Scene));

Help & Credits

Help: Muestra las instrucciones del juego.

Credits: Muestra la información sobre el desarrollador del juego.

Se dibuja la textura de fondo con la información específica de cada escena y también un botón para volver al menú principal.

Game

Donde se ejecuta el juego. El HUD muestra la cantidad de vidas, el puntaje y el botón de pausa. El juego consiste en hacer clic en la comida que aparece en la pantalla para evitar que se caigan.

- handle () -> comprueba las coordenadas donde se tocó la pantalla y utiliza la función de la clase Food contains _ point () para comprobar si se tocó un elemento de comida. Si es así, aplica una fuerza y suma puntos. También hace una verificación similar, pero para el botón de pausa. Si se toca este botón, el usuario se redirige a la escena de pausa.
- load_textures () -> carga todas las texturas y crea los *sprites* para los elementos del juego.
- render () -> dibuja todas las texturas, *sprites* y elementos de texto, como las vidas, la puntuación y el temporizador del juego.
- get_ready () - muestra un mensaje para que el jugador pueda prepararse para comenzar a jugar. Aplica un temporizador para evitar que el juego comience antes de tiempo.
- run_simulation () -> donde ocurre la acción del juego. Después de la generación de los primeros elementos, se llama a esta función. Calcula los límites y retrasos para generar nuevos panqueques y fresas. Si un elemento de comida cae por debajo del límite inferior de la pantalla, se elimina una vida. Si no hay más vidas, se termina el juego y se redirige al usuario a la escena de *gameover*.
- create_sprites () -> llama a la función de crear panqueques y establece el límite inicial de cuántos panqueques deben generarse para que aparezca un elemento de bonificación.
- create_pancakes () -> crea un panqueque. se establece la posición inicial, siendo *x* aleatoria e *y* debajo del límite inferior de la pantalla. También se agrega un impulso para que la comida se arroje en la pantalla del juego desde abajo. Después el nuevo elemento se agrega a la matriz “food”. Esta función tiene un contador que se usará para calcular la generación de artículos de bonificación y establece el retraso de tiempo entre la generación de un nuevo panqueque.
- create_strawberries () -> similar a los panqueques, pero crea una fresa. Este es el elemento de bonificación. También establece la posición, agrega un impulso e inserta en la matriz “food”. Aquí se recalcula el límite de panqueque que se establece cuando se genera este elemento.

Al igual que en el menú, la siguiente línea de código se usa para cambiar a la escena de *gameover*, pero esta vez se pasan la puntuación y del temporizador del juego como parámetros.

- director.run_scene (shared_ptr< Scene > (new Gameover_Scene(score_counter, game_timer)));

Food

Esta clase crea un elemento de comida a partir del atlas *sprite*. Este elemento se agrega a una matriz "food" en la escena del juego cuando se llama las funciones de creación.

- render () -> aquí se crea la animación de la comida. Si la velocidad en *y* es mayor que 0, entonces muestra los *sprites* hacia arriba. Si no, muestra los *sprites* que caen. Las animaciones se realizan solo en el eje *y*.
- contains_point () -> comprueba si se ha hecho clic en un elemento. Calcula la distancia entre el punto tocado en la pantalla y el radio del *collider* del elemento. Si la distancia es menor que el radio, entonces es cierto.
- apply_force () -> aplica una fuerza a un elemento en el que el usuario ha hecho clic.
- apply_impulso () -> aplica el impulso inicial.
- También tiene funciones para obtener y establecer valores.

Pancake & Strawberry

Ambas clases crean los alimentos individuales (panqueques y fresas). Heredan la clase de comida.

- get_points () -> obtiene los puntos asignados a cada elemento.

Pause

El menú de pausa aparece cuando el usuario hace clic en el botón de pausa en la escena del juego. Tiene una opción para volver a jugar y otra para volver al menú principal.
Esta clase aplica las mismas funciones que el menú.

Gameover

Esta pantalla aparece cuando el usuario pierde toda su vida. Presenta un mensaje de *gameover*, así como las opciones para volver a jugar o volver al menú principal. Para los botones esta clase aplica las mismas funciones que el menú.

También muestra el puntaje final y el tiempo de juego. Para eso recibe en su constructor los parámetros puntaje y tiempo enviados en la escena de juego.

- Gameover_Scene (int score, float time).