

Reporte de Diseño: Annoying Bird (Clon de Flappy Bird).

Carlos González
caal.0522@gmail.com

10 de junio de 2014

1. Introducción

A continuación se muestra un reporte de diseño de un clon de Flappy Bird, el cual lleva por nombre Annoying Bird, realizado en lenguaje de programación python, usando pygame como motor de juego.

2. Diseñadores

El juego fue diseñado y programado por¹:

- Roberth Castaño García
- Carlos Álvaro González Echeverry

3. Género

El juego se puede encajar simplemente en el género de juegos casuales y juegos Sidescroller.

4. Plataforma

El juego actualmente está limitado a computadores personales, aunque puede correr en computadores con sistema operativo Windows, Mac, o Linux. A continuación se muestran los requerimientos mínimos de Hardware:

- Sistema Operativo Windows, Mac, o Linux
- Instalación de intérprete de Python v2.7 y Pygame 0.9
- Memoria RAM 128MB

¹Cabe resaltar que no se tuvo mas que los sprites de Flappy Bird para el desarrollo del juego

- Espacio en disco 15MB
- Procesador de al menos 1GHz
- Periféricos:
 - Teclado, Mouse (No soporta Gamepad)

5. Versiones

El software fue creado en su totalidad desde el inicio del proyecto usando el manejador de versiones Git, y alojado en la plataforma GitHub, todo el código de cada versión se puede encontrar en: <https://github.com/caal-15/Flappy-Python>.

La versión actual del software es 0.5.

6. Sinopsis de jugabilidad y contenido

Annoying Bird es un sidescroller sencillo en 2D (cuyo objetivo en parte en este caso es frustrar al usuario) en el cual la meta es guiar un pájaro (el Annoying Bird) de manera que pueda pasar entre una oleada de pares de túneles (uno arriba y uno abajo por cada par), y en general, el objetivo es pasar por entre la mayor cantidad de pares de túneles posibles, en este caso se han hecho algunos ajustes para que sea un poco mas difícil que el original, y así hacerlo incluso mas frustrante.

El juego, además de la escenografía y sprites del original, contiene una pista arreglada, grabada, y mezclada por el diseñador Carlos González.

7. Características principales

- Sidescroller 2D
- Estilo retro con “píxeles grandes” estilo juego SNES
- Arreglo musical original
- 60 cuadros por segundo
- Animación por sprites
- Sistema sencillo de scoring

8. Licencia

Actualmente el software se encuentra regido por licencia GPL v3², lo que significa que es de código abierto y puede ser clonado por quien lo desee desde la plataforma GitHub

9. Mecánicas

9.1. Modo de juego

Inicialmente se muestra un menú estático, para empezar, el jugador debe apretar la tecla “Espacio”, esto hará que el juego inicie y que la música empiece a sonar, cada vez que el jugador aprete la tecla espacio, el Annoying Bird subirá una distancia vertical y luego empezará a caer, la idea es transportar el Annoying Bird a través del nivel usando estos “Saltos”, pero sin que choque con ninguno de los túneles o el piso mostrado en pantalla, si lo anterior ocurre, el usuario será llevado de nuevo al menú principal y la música parará, esto significa que la música se reiniciará cada que el usuario pierda, generando aún mas frustración en el mismo, además con el fin de cumplir este objetivo (el de frustrar al jugador), no se ha implementado una función para pausar el juego. Para simplemente escuchar la música, el usuario puede presionar S en el menú, para salir del juego, el usuario simplemente debe hacer click en el botón cerrar.

9.2. Elementos interactuantes

- Annoying Bird (pájaro), aparece en la parte central de la pantalla y solamente se mueve verticalmente a comando del usuario y efecto de la gravedad implementada.
- Túneles, Aparecen por pares frente al Annoying Bird, uno arriba y uno abajo y se acercan constantemente al mismo (a velocidad constante), cada par está separado una distancia constante vertical, y su punto de aparición vertical es random (dentro de un rango), los pares de túneles están separados entre si una distancia constante.
- El suelo, está constantemente bajo el Annoying Bird, moviendose de derecha a izquierda a la misma velocidad que los túneles para dar la ilusión de movimiento.

9.3. Cámara

La cámara, como en todo sidescroller, es una cámara fija en tercera persona.

9.4. Controles

- En el menú principal:

²Aquí se puede encontrar la información sobre la licencia GPL v3 <http://www.gnu.org/copyleft/gpl.html>

- ESPACIO para iniciar el juego, S para escuchar el soundtrack del juego.
- Durante el juego:
 - ESPACIO para hacer que el Annoying bird “Aletee” Ganando altura.

9.5. Puntaje

El juego lleva un sistema de puntaje sencillo mostrado en la parte superior de la pantalla, por cada par de tuneles que el Annoying Bird logre evitar exitosamente, el puntaje subirá en 1.

9.6. Estado del juego y Menús

El juego sólo tiene un menú principal sencillo, al cual regresa cada que el usuario pierde, y por supuesto aparece al comienzo del juego, en la sección anterior se describe lo que se puede hacer en este menú.

10. Tecnología

10.1. Software y Plataformas de Desarrollo

El software se desarrolló usando las siguientes herramientas de software:

- Python v2.7 (Intérprete)
- Pygame v1.9 (Motor de juegos)
- Editor de texto Gedit v3.10 personalizado
- Desarrollado usando SO Linux Debian v7.3
- Git v1.7 (Manejador de versiones)
- Plataforma online GitHub
- Clase Spritesheet tomada de <http://www.pygame.org/wiki/Spritesheet?parent=CookBook>
- Herramienta de Diagramación en Creately (<http://createlly.com/>)

10.2. Estructura del software

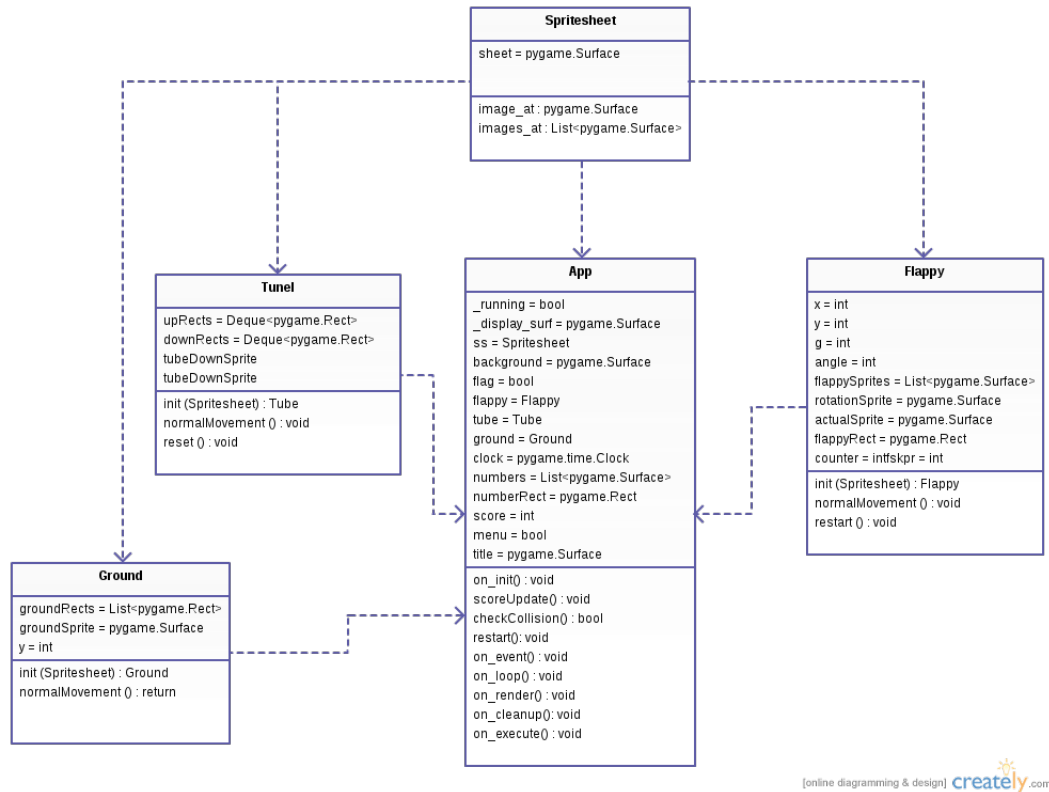


Figura 1: Diagrama de Clases

10.2.1. Breve descripción de las clases y métodos

- **Clase App:** La aplicación en sí, encargada del manejo centralizado de todo el juego, al ser la clase principal tiene gran cantidad de atributos, a continuación se explica brevemente cada uno:
 - `_running`: booleano encargado de saber si la aplicación está corriendo o no
 - `_display_surf`: la superficie raíz donde se mostrará el juego.
 - `ss`: imagen donde se encuentran todos los sprites a ser usados por el juego (spritesheet)
 - `background`: imagen del background general del juego
 - `flag`: bandera encargada de saber cuando se presionó la tecla y liberó espacio (debido a que no hay manejo nativo del evento Tap)
 - `flappy`: instancia de la clase Flappy usada para controlar el Annoying Bird

- tube: instancia de la clase Tube usada para controlar los tubos que van apareciendo.
- ground: instancia de la clase Ground que controla el movimiento del suelo
- clock: reloj general de la aplicación tomado de `pygame.time`
- numbers: lista conteniendo los sprites de los números
- numberRect: rectángulo para el primer dígito del score (el otro sprite se pintará en la cercanía de este)
- score = entero que contiene el puntaje actual del jugador
- menu = bandera booleana que determina si se está en el menú principal o no
- title = imagen que tiene el menú principal

Además de ello al ser la clase principal contiene la mayoría de métodos, está diseñada para que cada función que deba cumplir la clase esté dentro de un bloque de código digerible, por este motivo la lógica de renderización, la de inicialización, el manejo de eventos, la lógica del juego, y la acción de limpieza y ejecución están en métodos diferentes, además de estos métodos comunes en los `sidescrollers`, se tiene un método de detección de colisiones, uno de reseteo, y uno para actualizar el score. A continuación se muestra una breve descripción de cada uno de los métodos:

- `on_init()`: método encargado de inicializar todos los atributos y otros componentes necesarios para el funcionamiento de la clase, además de setear aspectos como el tamaño de la ventana principal
- `scoreUpdate()`: método que verifica si el Annoying Bird ha cruzado un túnel, y actualiza el score de acuerdo a ello
- `checkCollision()`: método que chequea si ha habido alguna colisión entre el Annoying Bird y algún objeto del juego, retorna un booleano como respuesta
- `restart()`: llama a los métodos de reposicionamiento de las clases Flappy y Tube, además de ellos pone la bandera `menu` en verdadero y para la música
- `on_event()`: Maneja toda la lógica de lidiar con Eventos del teclado y pantalla
- `on_loop()`: básicamente contiene toda la lógica del juego, llama a los movimientos normales de Flappy, Tube y Ground luego de llamar al método `checkCollision`, y llamar a `reset` si es necesario, también llama a `scoreUpdate`
- `on_render()`: realiza todas las funciones de rendering con la ventana principal usando los rectángulos³ y sprites
- `on_cleanup()`: se encarga de limpiar todo cuando la aplicación se deje de ejecutar

³El manejo de posición en `pygame` no se hace directamente con los sprites, sino con rectángulos que representan la posición de los mismos, y luego se usan como base para pintarlos en pantalla, el método `checkCollision` también chequea, en realidad colisiones entre rectángulos

- `on_execute()`: encargado de llamar los otros métodos `on_` (`loop`, `render`, `init`, `event`) durante el transcurso de la ejecución del programa, hasta que `on_event` envía una señal indicando el fin de la ejecución, en cuyo caso se llama a `on_cleanup`
- **Clase Flappy:** Clase encargada de mantener los atributos propios del Annoying Bird, contiene atributos que indican la posición del Annoying, un atributo que simboliza la gravedad (mas exactamente la velocidad de caída), una Lista de sprites, un ángulo de rotación para cuando está cayendo, temporales para el sprite actual y el rotado, el rectángulo que representará la posición del Annoying, y dos contadores encargados de llevar la animación (cambiar de sprite). A continuación se describen brevemente los métodos de la clase, cabe anotar que el método `normalMovement` también se encuentra en las clases `Tube` y `Ground` y por consecuencia solo será explicado en esta sección, ocurre lo mismo con el método `restart` de la clase `Tube`.
 - `init(SpriteSheet)` recibe una hola de sprites para tomar los sprites del Annoying Bird, inicializa los otros atributos.
 - `normalMovement`: realiza el movimiento normal del Annoying Bird (usando la gravedad y la posición en y) durante el juego (es decir reposiciona el rectángulo del Annoying para cada frame).
 - `restart()`: reposiciona el Annoying al centro de la pantalla.
 - **Clase Tube:** Creada con el mismo propósito que la clase `Flappy`, solo que con respecto a los tubos que debe esquivar el jugador, cabe decir que esta clase contiene realmente en todo momento 8 rectángulos en todo momento representando las posiciones de 4 pares de tubos, es decir en todo momento existen 4 pares de tubos en movimiento, pero jamás más de 4, con el objetivo de economizar recursos, para esto se hace un `shift` (hacia la izquierda) en la lista de rectángulos cada que un par de tubos salen de pantalla por la izquierda, por ese motivo se utiliza la estructura de `Dato Deque` en vez de una lista, ya que esta tiene funciones integradas de `shifting`.
 - **Clase Ground:** Clase creada con el propósito de llevar el movimiento normal del terreno, funciona de manera muy similar a las clases anteriores.

10.3. Hardware

Ningún hardware especializado se utilizó para realizar la aplicación, y tampoco se requiere hardware particularmente potente para correrla (se puede ver en la sección plataforma), cabe aclarar que la aplicación de momento corre en CPU, y no aprovecha los recursos de GPU que puedan estar disponibles, sin embargo debido a su bajo consumo de recursos pareciera innecesario hacer este tipo de optimizaciones.