



Pokecity

Gestiona tu ciudad estando inactivo

Manual técnico

Laura Jiménez Villalba





Tabla de contenidos



Introducción	2
Análisis del título	2
Diseño del producto	3

Introducción

PókeCity es un videojuego de gestión inactiva donde el jugador maneja una ciudad habitada por Pokémon. El usuario, recién nombrado como alcalde de la ciudad deberá de tomar decisiones para que prospere. Decidirá los edificios construidos, los ciudadanos, sus oficios e incluso podrá permitir que se creen variadas familias. El principal objetivo es ir consiguiendo dinero para que con el paso del tiempo virtual vayas ayudando a que los ciudadanos Pokémon consigan trabajos que les gusten, aumentes la amistad que tienes con ellos dándoles regalos y diseñes a tu gusto la urbanización.

Análisis del título

Dado a que el proyecto es un juego, más que explorar si resuelve alguna necesidad, serán detallados puntos a favor de porqué es una experiencia interesante y diferente a otros títulos de la saga Pokémon.

- Los videojuegos de Pokémon siempre han tenido momentos más relajantes y otros más frenéticos, siempre buscando conectar y que se representen a las criaturas como compañeros más que simplemente herramientas de combate. Hay juegos más viejos y discontinuados como “My Pokémon Ranch” donde los usuarios se dedicaban a crecer una granja solo por el afán del coleccionismo, valor muy presente en la saga. Existen también otros spin-offs más relajantes como “Pokémon Snap”, pero ninguno presenta una jugabilidad “Away from Keyboard” o “Idle” (inactivo).
- Además del factor Idle ya mencionado, PókeCity tiene presentes valores importantes de la saga ya mencionados como el compañerismo comunitario y el coleccionismo agregados a tomas novedosas que los propios creadores originales no han tenido el valor de explorar al menos recientemente y no en el enfoque que he decidido darle. Respeta cánones de la propia franquicia, ya que los Pokémon también pueden trabajar (cosa que aparece en los dibujos animados y en los juegos), además del hecho de sentir que pertenecen a una familia o incluso manada.
- Pokémon es una IP muy variada con fans más o menos “hardcore” pero es claro que al público más casual le parecerá una propuesta divertida y tranquila y a los fans más acérrimos les parecerá una bocanada de aire fresco de la que disfrutar de vez en cuando.
- Tomo en cuenta y hago presencia de elementos de juegos anteriores que los que ya eran fans pueden reconocer (por ejemplo, un jugador tendrá más interés en conseguir tener en la ciudad sus Pokémon favoritos, aunque no se diferencien mucho de cualquier otro viéndolo técnicamente) como los edificios o los Pokémon e incluso introducirlos a nuevos jugadores.
- Muchos jugadores hacen retos en los juegos originales para darle una dificultad extra, en PókeCity podría ocurrir lo mismo, haciendo retos como

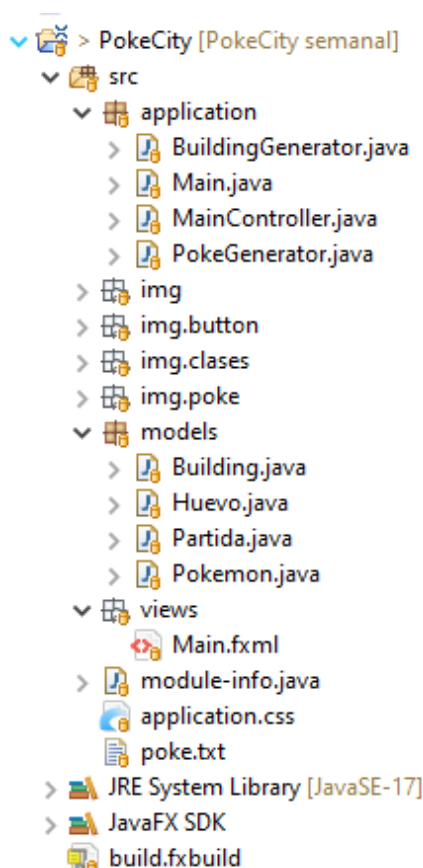
tener el mayor número de Pokémon Legendarios o limitarse el uso del dinero, regalos o huevos, o incluso crearse o ajustar sus propias narrativas, ya que ellos tienen el control realmente sobre elementos como los casamientos y las familias.

En resumen, su toque simple pero efectivo más la jugabilidad dan una experiencia relajante que pueden disfrutar tanto personas ajenas a la saga como los más experimentados.

Diseño del producto

Se utiliza Java y JavaFX para darle un aspecto moderno y sencillo. Con Eclipse IDE para facilitar todas las tareas. Necesita 2 archivos específicos en la misma carpeta que el .jar.

- El archivo names.txt es modificable para que los jugadores puedan añadir cualquier nombre para que salga al azar en los Pokémon. De forma predeterminada el .jar ya viene con uno. Para que funcione, los nombres deben estar colocados 1 en cada línea y no ser demasiado largo.
- El archivo savefile.ser no viene con el .jar ya que se crea al cerrar el juego, como forma de guardar partida. Es un archivo binario personalizado que guarda serializada la clase Partida (véase en el siguiente punto). Si se borra, el juego empezará de cero. También se puede eliminar desde el propio juego. Uno puede tener varias partidas si quiere, pero debe mantener el archivo que quiera cargar con exactamente ese nombre.



La arquitectura de las clases usadas en el programa es la siguiente:

El paquete application contiene las clases principales. Main es la que carga el archivo Main.fxml con su controlador. MainController es el que maneja toda la aplicación y contiene la mayoría de métodos. BuildingGenerator y PokeGenerator son las otras dos clases que manejan la creación aleatoria de edificios y Pokémon respectivamente. He usado 4 modelos que son los objetos personalizados. Building y Pokémon son los objetos que van cambiando con la partida y que son parte de esta. Huevo se encuentra dentro de Building hasta que se convierte en Pokémon. Partida es el principal archivo serializable que es el guardado en el archivo savefile.ser. Todas las pantallas se manejan desde una sola vista. El archivo poke.txt contiene todos los nombres de las especies Pokémon que PokeGenerator se dedica a ajustarlos con su imagen adecuada.