



Subida de nota

Tiraje de dados

DnD

Manuel Robles Barrales



Índice

Introducción	3
Arquitectura del software	3
Diseño de la aplicación	3
Implementación	3
Pruebas	3
Gestión del proyecto	3
Conclusiones	3



Introducción

Este proyecto consta de la creación de una aplicación para tirar dados de rol, desde el D4 hasta el D100 (es el dado decimal, aunque yo lo he implementado como un dado más, porque para mí tiene más sentido). Su función principal consta de elegir el dado que quieres tirar y dentro de su propia interfaz, podrás elegir si tirar el dado, guardar el resultado tras guardar el dado (Si no has tirado el dado, no podrás guardar el resultado) o volver atrás para elegir otro dado. Los objetivos principales son: Implementar las distintas clases en su botón concreto, guardar los resultados y tener una interfaz gráfica intuitiva y fácil de usar, además de ser temática

Arquitectura del software

Aquí muestro el diagrama de clases, el cual es un poco simple, pero es más que suficiente para detallar concretamente el funcionamiento del software:

CREAR UML PARA LAS CLASES DEL PROGRAMA

Aquí muestro los requisitos funcionales y no funcionales de la aplicación

Requisitos Funcionales			
Número	REQUERIMIENTO	DESCRIPCIÓN	PRIORIDAD
RF1	Caras	Variable principal en la que se basa esta aplicación, define el número de caras que tienen los dados	Alta
RF2	Tirar	Método personalizado principal de la aplicación, permite realizar la acción de tirar los dados para obtener los resultados del número de que haya salido	Alta
RF3	Guardar resultados	Método personalizado que permite guardar los resultados al tirar los dados, no es tan importante como la de tirar, pero sí permite tener un recuento de los resultados obtenidos	Alta
RF4	Muestra de resultados	Método personalizado que permite mostrar en la interfaz los resultados de las tiradas de cada dado individual	Alta
RF5	Elección de dados	Permitir elegir al usuario el dado concreto que quiera tirar	Medio
RF6	Interfaz intuitiva	Se busca una interfaz intuitiva y amigable siguiendo con la temática medieval de un juego de rol típico de DnD, cuenta con sus botones e imágenes para identificar qué dado es el que se ha clicado para lanzar	Alta
RF7	Rapidez de respuesta	Se busca rapidez de respuesta en la aplicación para que se ejecuten rápido las funciones deseadas sin tener el denominado efecto de "Input.Lag", así mismo como la carga propia de la aplicación al ser lanzada o abierta	Medio

NO Requisitos Funcionales			
Número	REQUERIMIENTO	DESCRIPCIÓN	PRIORIDAD
RNF1	Compatibilidad	Se busca compatibilidad con todos los sistemas compatibles con Java	Alta
RNF2	Portabilidad	Se busca la portabilidad, tanto para sistemas operativos Windows, basados en Linux, Mac, IOS y Android	Alta
RNF3	Estable	La aplicación debe ser estable y por lo tanto, no se cierre abruptamente	Alta
RNF4	Escalabilidad	Esta aplicación está pensada para ser escalable y con buena base de mantenimiento, ya que se hace uso de herencia para crear los dados y con posibilidad de modificar el comportamiento de la clase padre para crear más métodos personalizados o dados según se requiera	Medio
RNF5	Fiabilidad	Esta aplicación es fiable, ya que es capaz de manejar excepciones producidas en tiempo de ejecución gracias a los bloques Try-catch	Alta



Diseño de la aplicación

La mecánica principal y única es la de elegir y tirar los dados, ya que es la función principal del programa y de la que la función de guardar los resultados depende. Cada vez que elijamos un dado y lo tiremos, habrá una animación que simulará el tirar el dado, que durará en torno a 4 o 5 segundos antes de mostrar el resultado.

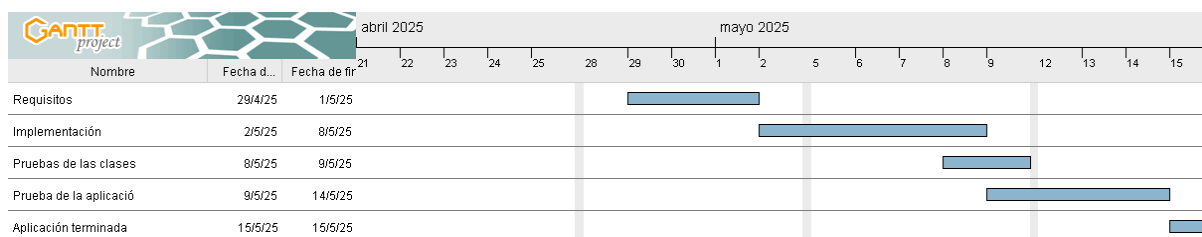
Implementación

Para la implementación de esta aplicación he usado el lenguaje Java, haciendo uso de la herencia de clases y de JFrame de la utilidad Java Swing, adicionalmente he añadido las bibliotecas java.io para poder hacer uso del GIF y de la música de taberna que tiene de fondo la aplicación al iniciarse. La implementación de cada objeto de cada clase, ha sido en sus JPanel correspondientes con los botones de las acciones correspondientes, adicionalmente, todos los botones de "Tirar dado" contienen la implementación para que aparezca el GIF durante los 4 o 5 segundos propuestos

Pruebas

Tras la implementación completa de la aplicación, hice las pruebas pertinentes para verificar que todo funcionase como debería funcionar, haciendo pruebas dado por dado y tratando de encontrar bugs o excepciones que pudieran ocurrir en el tiempo de ejecución del programa

Gestión del proyecto



Conclusiones

Mis conclusiones al hacer este proyecto han sido las siguientes:

Mi primera conclusión fue que sé más de lo que me pensaba y que no me ha costado mucho crear la mayoría del código

La segunda conclusión fue que al añadir una interfaz gráfica con diferentes “pantallas” es algo nuevo y por lo tanto, complicado al no saber muy bien como implementarlo, pero fué una buena forma de aprender y crecer como programador

La tercera conclusión fue que al crear la aplicación y pensarla, podría ser una aplicación totalmente usable para la gente que juegue juegos de rol o DnD en su defecto y no tenga dados o se le olviden, como tampoco es difícil de usar y tiene música de fondo, puede crear un ambiente más que óptimo para una partida seria o una partida entre amigos

La última conclusión que saqué es que, he aprendido bastante y que hacer proyectos por tu propia cuenta y cosas pensadas por tí mismo pueden ser tanto didácticas como entretenidas, ya que al hacerla, aunque me encontrara con dificultades, quería seguir para terminarla y verla funcionar, ya que me pareció desde un primer momento una muy buena idea, además de últimas tuve que hacer unos cambios inesperados porque no funcionaba como yo quería que funcionase, logré cambiarlo y no cambié nada de los archivos ya creados como los UML y etc por el hecho de que, creo que esto es el día a día de lo que me encontraré en un entorno laboral si ejerzo de programador en algún momento.

