TEC | Tecnológico de Costa Rica

Ingeniería en Computación
Programación Orientada a Objetos - Unearth
Profesora: Adriana Álvarez

1. Motivación

"Hace mucho tiempo, su civilización fue un imperio de maravillas que se extendió por todo el mundo, una joya de invención y sabiduría que todos buscaban.

Entonces algo sucedió, y todo se arruinó.

Ahora usted y su pequeña tribu están tratando de recuperar lo que se perdió, reconstruir lo que se rompió y recordar de dónde vienen."- Unearth.

2. El problema

Hoy en día existen muchos juegos de mesa entretenidos y de diferentes temáticas, tanto son de dados, fichas, cartas, de roles, de mesa. En fin, existe una increíble cantidad de juegos que permiten pasar el rato con amigos o familiares.

Por desgracia, muchos de estos juegos son muy difíciles de conseguir, son costosos y en la mayoría de los casos estos no se encuentran en una forma computarizada, dando así un mayor alcance al mercado de personas que son más amantes de la tecnología que al juego tradicional.

Por lo tanto, en esto consiste su proyecto. Usted y su equipo deberá elaborar una versión programada del juego Unearth.

"Unearth" es un juego de mesa, de colocación de dados y juego de colección. Cada jugador lidera su propia tribu, representada por cinco dados. Los jugadores se turnan para lanzar y colocar dados en un intento de reclamar ruinas.

Los lanzamientos altos ayudan a los jugadores a reclamar ruinas, mientras que los bajos ayudan a recolectar piedras. Esto abre dos caminos hacia la victoria: reclamar conjuntos de ruinas o usar piedras para construir maravillas. Las cartas ayudan a afectar tus lanzamientos de dados o dados en el juego y las maravillas pueden otorgar habilidades que impactan en el desarrollo del juego.

3. Arquitectura

Durante la revisión se deberá justificar las decisiones hechas sobre el diseño de la aplicación. Adicional a esto se espera una implementación elegante, con un código fuente que siga las normas básicas establecidas en el libro Clean Code (Martin, 2008).

4. Requerimientos

El juego debe funcionar en su totalidad, por lo tanto, se les entrega (en formato pdf), el manual del juego, y también un video que muestra a detalle cada uno de sus componentes, en que consiste, como se juega y como se llega a ganar.

Requiere utilizar todas las técnicas vistas en cases sobre programación orientada a objetos.

5. Otras consideraciones

1. La aplicación debe ser gráfica.

2. Para el diseño de las cartas puede utilizar imágenes de internet que sean de libre uso.
3. El programa debe ser escrito en su totalidad en el lenguaje programación de JAVA.
4. Se evaluará el diseño y la funcionalidad (+1 punto extra en la nota).
5. La aplicación debe tener un logo creado por el equipo. No puede ser un logo copiado de internet o de otras personas y debe ser original. Puede solicitar ayuda del estudiante de diseño.
6. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo. Se recomienda Gitlab.
7. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
8. La entrega oficial se debe hacer en un zip en el Tec Digital antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Reglas del Juego.
9. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
10. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.

6. Documentación

La siguiente documentación debe ser entregada:

1. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación. Si utiliza un wiki para la documentación, esta debe accederse desde la aplicación.
2. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.
3. El código debe estar documentado internamente según Clean Code.
4. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.

7. Apoyos

1. Manual del juego: se encuentra en la carpeta de proyectos del TD.
2. Video explicativo del juego Unearth: https://www.youtube.com/watch?v=eCK7IY5ebL4