

Lección 97

Mejorando e iterando parte 2

Propósito

Los equipos han desarrollado en este punto un prototipo de aplicación que ha pasado por múltiples iteraciones y rondas de pruebas de usuario. Con la información y la orientación obtenidas de la última ronda de pruebas de usuario, cada alumno tiene la oportunidad de planificar e implementar mejoras en la aplicación del equipo. Según el tiempo que tengas disponible y el interés del alumno, puedes ejecutar el ciclo de prueba e iteración tantas veces como te parezca.

Si bien es tentador hacer de esta Lección una inmersión más profunda en la programación para todos los estudiantes, anima a cada alumno a encontrar los elementos del diseño de la aplicación que te interesen y concéntrense en eso para sus mejoras personales. Si bien la mejora puede significar agregar funcionalidad programática, también puede significar un diseño más llamativo, una copia de texto más informativa, una mejor uniformidad de la iconografía o cualquier cantidad de otras características no relacionadas con la programación.

En este punto del proceso de desarrollo, pueden optar por dedicar todo el tiempo que desees a las iteraciones y las pruebas. No hay más tiempo presupuestado para la prueba después de esta iteración, pero pueden utilizar las dos lecciones anteriores como una guía para el ciclo a través de iteraciones adicionales.

Secuencia para el aprendizaje

Conocimiento inicial (5 min)

Lección 1 (30 min)

Lección 2 (30-90 min)

Transferencia del conocimiento (15 min)

Objetivos

Los estudiantes serán capaces de:

Lección en línea

Recursos

¡Atención!

Por favor, haga una copia de cada documento que planea compartir con los estudiantes.

Para los estudiantes:

- [Análisis de prueba del usuario de la aplicación - Guía de actividades](#)

Vocabulario

- **Error:** Parte de un programa que no funciona correctamente.
- **Característica:** Un elemento funcional individual de software, como la capacidad de hacer algo nuevo, trabajar en múltiples plataformas o realizar de manera más eficiente.

- Analizar los comentarios de los usuarios de la Lección anterior y determina una lista de errores (fallas) que deben corregirse y características que podrían agregarse a la aplicación.
- Priorizar los errores y características de acuerdo con el impacto y la facilidad de implementación.

Preparación

- [Análisis de prueba del usuario de la aplicación: guía de actividades](#)
- Colocar notas adhesivas y papel para cada equipo.

Estrategia de aprendizaje

Conocimiento inicial (5 min)

Errores y características

Indicación: En función de las pruebas de usuario, ¿cuáles son algunos de los errores que descubriste en tus aplicaciones? Mantén un registro de estos en el tablero.

Discute: Es probable que encuentren muchos errores, pero ¿qué otros comentarios recibiste de los usuarios? Ayudará a la clase a identificar los comentarios que implican la necesidad de nuevas funcionalidades. Vamos a llamar a estas solicitudes funciones. Hoy los equipos trabajarán para identificar los errores y las características descubiertas a través de las pruebas, priorizarlas y planificar mejoras.

Lección 1 (30 min)

Distribuye: Papel, notas adhesivas y una copia del Análisis de pruebas del usuario de la aplicación: Guía de actividades para cada equipo.

Análisis de prueba del usuario de la aplicación

- 1. Interpretar los comentarios del usuario:** Los equipos comienzan completando una tabla en T que ayudará a conectar las observaciones de prueba de usuarios específicos con los posibles errores o características que faltan que revelan. Esta información será la base de una sesión de lluvia de ideas para crear una lista de errores encontrados en la aplicación existente y características que el equipo quisiera agregar a nuestra próxima iteración de la aplicación en la siguiente Lección.
- 2. Sesión de lluvia de ideas:** Una vez que los equipos han organizado todos sus comentarios en la tabla T, pueden pasar a la fase de lluvia de ideas. Completa una nota adhesiva por cada error o característica identificada. La parte superior de la nota adhesiva debe decir ERROR o FUNCIÓN. La mitad de la nota adhesiva debe ser una descripción de cuál es el error o las características, incluidos los pasos para reproducir el problema si es necesario. La parte inferior de la aplicación debe tener una estimación rápida del tiempo (en minutos) que tardará en solucionar este error o implementar esta característica. Deja que los estudiantes sepan que tendrán la oportunidad de refinar este cálculo más adelante.

Los errores a menudo son más fáciles de identificar para los estudiantes que los comentarios que las características. Aliente a los estudiantes a considerar realmente cómo los usuarios podrían haber supuesto que la aplicación debería funcionar, o los lugares donde necesitaron ayuda o dirección para navegar con éxito.

Es tentador centrarse únicamente en los errores que son la causa de, o se puede resolver con, el código. Recuerde a los estudiantes que hay muchos roles y conjuntos de habilidades en los equipos de desarrollo de software, y que no todos los errores y características son tratados por los programadores. El color inconsistente, el texto confuso y los diseños contraintuitivos son todos errores potenciales que es importante tratar. Esta es una oportunidad para que los estudiantes con otros conjuntos de habilidades brillen y hagan una fuerte contribución a sus aplicaciones, así que haga un esfuerzo extra para ayudar a resaltar a esos estudiantes.

en la Lección; todo lo que es necesario ahora es un cálculo aproximado.

3. **Análisis de fallas y funciones:** En una hoja de papel, dibuja los dos cuadros en la página dos de esta guía de actividades. Para cada una de las notas adhesivas generadas en la fase anterior, discute si es urgente o no, y si parece ser fácil o difícil de implementar. Con base en esa discusión, coloca el adhesivo en el cuadrante apropiado.

Discute: Ahora que tienen sus errores y características categorizados, cuál de las cuatro categorías debería ser la primera que aborda. ¿Cuál debería ser el último?

Lección 2 (30-90 min)

Reparar errores y agregar funciones

Niveles de Code Studio

Implementación de funciones y reparaciones de forma individual:

Debido a que App Lab es difícil para que equipos grandes de estudiantes trabajen de forma simultánea, los estudiantes trabajan en la implementación de funciones y soluciones individualmente o en pares. El beneficio de este enfoque es que permite que el equipo en su conjunto explore más características potenciales que si estuvieran trabajando en la misma aplicación, pero también significa que la aplicación de cada miembro del equipo será diferente de las otras a medida que se realicen más cambios.

Como Lección de extensión opcional, podría proporcionar un día de trabajo para que los estudiantes compartan sus cambios entre ellos e intenten integrar todas las funciones agregadas en una sola aplicación.

Mantener registro de los cambios:

Cuantas más funciones agreguen los estudiantes a su aplicación, más probabilidades tendrán de presentar nuevos errores imprevistos. Para garantizar que los estudiantes siempre guarden la copia de trabajo más reciente de su aplicación, deben hacer un remezcla antes de agregar nuevas funciones. Asegúrese de que los estudiantes dejen el remix solo y vuelvan a este nivel para hacer su programación; esto garantizará que cuando vayan a enviar su aplicación para su revisión, sea la versión más reciente

Esta Lección puede tomar tanto o tan poco tiempo como desee, siempre y cuando a cada alumno se le permita el tiempo para implementar al menos una corrección o característica. Incluso puede considerar volver a esta Lección más adelante en el año cuando tenga algo de tiempo libre; la distancia desde un problema puede agregar una perspectiva útil.

Transferencia del conocimiento (15 min)

Preguntar: Pedirá a los estudiantes que reflexionen sobre el desarrollo de las cinco prácticas de Descubrimientos CS (Resolución de problemas, Persistencia, Creatividad, Colaboración, Comunicación). Elegirá una de las siguientes indicaciones según lo considere apropiado:

- Elige una de las cinco prácticas en las que crees que demostraste crecimiento en esta Lección. Escribe algo que hiciste que ejemplifica esta práctica.
- Elige una práctica en la que pienses que puede seguir creciendo. ¿Qué te gustaría mejorar?
- Elige una práctica que pensaste que era especialmente importante para la Lección que completamos hoy. ¿Qué lo hizo tan importante?

Experiencias de aprendizaje de profundización

Use estos Contenidos para ampliar el aprendizaje de los estudiantes. Se pueden usar como Contenidos extras fuera del aula.

Fusionar nuevas características

En esta fase de iteración, cada estudiante ha estado trabajando en su propia copia de la aplicación del equipo. Si tienes tiempo disponible, considera la posibilidad de que los equipos vuelvan a reunirse para fusionar sus actualizaciones en una sola aplicación maestra. Esto puede ser bastante complicado, especialmente cuando los estudiantes pueden haber agregado códigos conflictivos o elementos de diseño, pero es un desafío excelente para desarrollar habilidades para resolver problemas y fortalecer la dinámica del equipo.

Seguimiento del progreso

Utilizando la tabla de tareas, comience a mover notas adhesivas desde el cuadrante Urgente / Fácil a la columna de tareas pendientes. Elija al menos dos tareas por miembro del grupo y escriba en la carpeta adhesiva a quién está asignada la tarea. A medida que trabajes en la mejora de la aplicación, moverás las notas adhesivas a través de las etapas de esta tabla.

Sugerencias de evaluación

Se sugiere los siguientes indicadores para evaluar formativamente los aprendizajes:

- Presentan un análisis de los resultados del trabajo
- Describe errores y mejoras
- Planifican la resolución de errores