Un dibujo de un perro

Descripción generada automáticamente con confianza media

## ALGORITMOS (CC215)

**Enunciado del Trabajo Final**

**2024-02**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PROFESORES:** | |  | | --- | | CARDENAS MARIÑO FLOR CAGNIY | | MENDOZA PUERTA HENRY ANTONIO  MENDIOLAZA CORNEJO EDSON | | PEREZ ROJAS JOHANN CRISTIAN | | RAYMUNDO CHACALTANA LUIS ALBERTO | | ROJAS SIHUAY, DIEGO | | SALAS ARBAIZA CESAR ENRIQUE | | TACZA VALVERDE, IGNACIO RUBEN | | VALDIVIA HUMAREDA LUIS ALBERTO | |
| **Sección:** | TODAS |
| **Fecha de evaluación:** | Semana 15 |
| **Ciclo Académico:** | 2024-02 |

**1. Introducción**

El aprendizaje requiere un estímulo inicial que resulte interesante y nuevo. Precisamente el juego es la conducta que desarrolla el estudiante para aprender con el estímulo de la curiosidad. El juego es un invento poderoso de la naturaleza, combinación de curiosidad y placer, es el arma más poderosa de aprendizaje (Mora 2013). El Trabajo Final del curso de Algoritmos que se propone consiste en la construcción de un videojuego empleando Windows Form que permitirá utilizar lo aprendido en las sesiones de clases y a la vez será un medio para concientizar al estudiante sobre los problemas actuales de nuestra sociedad que generan impacto en la salud pública, seguridad, bienestar, así como factores globales, culturales, sociales, ambientales y económicos.

**2. Objetivo**

EI presente documento define al Trabajo Final (TF) y a Ia rúbrica que permite evaluar eI Iogro deI curso CC215 Algoritmos. EI objetivo deI Trabajo Final permitirá que Ios estudiantes apliquen Ios conocimientos adquiridos de Programación Orientada a Objetos, Conceptos de cIases, objetos, relaciones de agregación, y relaciones de herencia.

**3. Logro del curso**

Estructura mentalmente soluciones plasmando los principios fundamentales de la programación orientada a objetos.

Competencia: **Pensamiento crítico -** Nivel de logro: 2

Capacidad para explorar de manera exhaustiva problemas, ideas o eventos para formular conclusiones u opiniones sólidamente justificadas.

Competencia: **Trabajo en Equipos Multidisciplinarios -** Nivel de logro: 1

Funcionar eficazmente como miembro o líder de un equipo que participa en actividades apropiadas para la disciplina del programa.

Competencia: **Diseño y Desarrollo de una Solución -** Nivel de logro: 1

La capacidad de aplicar el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas con consideración de salud pública, seguridad y bienestar, así como factores globales, culturales, sociales, ambientales y económicos.

**4. Enunciado**

En este videojuego, el jugador tomará el rol de un "Guardián del Medio Ambiente" cuya misión es restaurar el equilibrio natural de un ecosistema que ha sido dañado por la contaminación y la deforestación. El jugador debe moverse por un mapa y recoger recursos como agua, plantas y residuos reciclables, con el objetivo de limpiar el ambiente y reforestar la zona. El éxito del jugador dependerá de su capacidad para gestionar estos recursos de manera efectiva y evitar que la contaminación siga expandiéndose.

El videojuego se desarrollará en un entorno en el que el jugador tendrá que tomar decisiones rápidas y estratégicas para restaurar el medio ambiente. A medida que avance, encontrará desafíos como evitar fuentes de contaminación, gestionar correctamente los recursos recolectados y enfrentar el crecimiento incontrolado de desechos.

Se debe tener en cuenta que los juegos son desarrollados en Windows Form y aplicando el paradigma de Programación Orientado a Objetos (POO) donde emplee polimorfismo, herencia, sobrecarga de métodos, se debe utilizar imágenes (Sprites) para el Trabajo Final y Windows Form.

**Ejemplo de videojuego usando Windows Form**



**5. Elaboración del videojuego**

En base a la información de la sección 4, el equipo de trabajo debe elegir un **objetivo** centrado en la recolección y gestión de recursos naturales, con el propósito de restaurar el equilibrio ambiental en un ecosistema afectado por la contaminación. La secuencia de inicio a fin deberá ser diseñada por cada equipo, considerando las dinámicas de recolección de recursos, manejo de la contaminación y reforestación. Para ello, el equipo de trabajo debe tener en cuenta un conjunto de requisitos establecidos, que guiarán el desarrollo del videojuego y asegurarán la correcta implementación de las mecánicas ambientales.

**6. Requisitos del videojuego**

A continuación, se mostrarán todos los requisitos necesarios para el desarrollo del videojuego.

**Misión**

* Restaurar el ecosistema recolectando y gestionando recursos naturales (agua, semillas, residuos reciclables) para reforestar el terreno afectado por la contaminación.
* El objetivo es reforestar al menos el 70% del mapa antes de que se acaben los recursos o el tiempo.
* El equipo de trabajo deberá diseñar el mapa y distribuir los recursos estratégicamente.

**Pantalla de Juego**

* El videojuego debe desarrollarse en Windows Form.
* El equipo debe representar los recursos, personajes y elementos del juego usando caracteres imágenes.
* El videojuego debe ser claro, intuitivo y amigable para el usuario, con una coherencia entre los símbolos elegidos y el tema ambiental seleccionado.
* El mapa estará representado como una cuadrícula (ejemplo: 20x20).

**Personaje principal**

* El "Guardián del Medio Ambiente" es el personaje principal, encargado de recolectar recursos y reforestar el terreno.
* Puede moverse por el mapa utilizando las teclas de dirección.
* Está representado por un sprite.
* Tendrá un número limitado de vidas o energía que puede perder si colisiona con áreas contaminadas o enemigos.

**Recursos**

Los recursos disponibles incluyen:

* Agua: Representada por un Sprite elegido por el grupo de usuarios, necesaria para reforestar.
* Semillas: Representadas por un Sprite elegido por el grupo de usuarios, usadas para plantar árboles.
* Residuos reciclables: Representados por un Sprite elegido por el grupo de usuarios, que el jugador debe recoger para limpiar el ambiente.
* El jugador debe gestionar estos recursos de forma efectiva para lograr el objetivo del juego.

**Aliados**

* No se requiere la presencia de aliados en este juego, pero el equipo de desarrollo puede optar por incluir personajes que ayuden al jugador proporcionándole recursos adicionales o información estratégica.

**Enemigos**

* Las fuentes de contaminación son los enemigos que se expanden a lo largo del mapa si no se controlan.
* Están representadas por un Sprite elegido por el grupo de usuarios, y al colisionar con ellas, el jugador perderá una vida o energía.
* El equipo puede elegir la cantidad de enemigos o fuentes de contaminación que se expanden en el mapa.
* El jugador debe eliminar la contaminación plantando árboles o utilizando residuos reciclables.

**Gestión de Archivos**

* Haciendo uso de las librerías de gestión de archivos de texto y/o binarios deberá guardar dos archivos
  + Archivo de texto, que contenga la configuración del juego que contenga los parámetros de configuración de este (por ejemplo: número de enemigos, numero de aliados, tiempo, etc.) con el siguiente nombre: configuracipon.txt
  + Archivo binario, que contenga información del ranking de usuarios que hay ido jugando el videojuego (por ejemplo: Nombre del jugador, puntaje, fecha, etc.) con el siguiente nombre: puntaje.bin
* Los archivos serán leído y escritos cada vez que el videojuego este ejecutándose, se deberá validar la existencia e integridad de estos, mostrando un mensaje de error de corresponder.
* La ubicación de los archivos dentro de la estructura del videojuego deberá ser dentro de la carpeta: datos/

**Dinámica del juego**

* El jugador debe recolectar suficientes recursos para poder plantar árboles y limpiar el ambiente antes de que el tiempo o sus vidas se agoten.
* Cuando el jugador logre reforestar el 70% del mapa, se mostrará una pantalla con una animación de "GANASTE".
* Si las vidas del jugador llegan a 0 o no consigue suficientes recursos para reforestar el terreno, se mostrará una pantalla con una animación de "PERDISTE".
* Mensaje de Perdiste (“Game Over”) cuando se cumpla la condición de derrota.

**Diagrama de clases UML**

* Desarrollar el diagrama de clases UML de todas las clases, incluyendo atributos, métodos y relaciones entre clases, que se utilizarán en el videojuego.

**7. Instrucciones para la entrega del Trabajo Final**

El Trabajo Final será entregado de forma individual en el Aula Virtual. Dicha entrega se realizará hasta **las 23:30 horas del domingo 17 de noviembre**. No se aceptará ningún Trabajo Final por correo electrónico.

El plazo es impostergable y por ningún motivo y/o circunstancia se recibirán trabajos fuera de la fecha y hora indicada con antelación. El Trabajo Final se realiza en equipo de trabajo (definido previamente con el docente de la sección), si un equipo de trabajo no presenta el Trabajo Final de acuerdo con el cronograma de entregas establecido por el docente a cargo de la sección, recibirá la nota de cero (00) correspondiente.

**8. De la calificación**

El docente a cargo de la sección tiene la autonomía para solicitar a cada estudiante que demuestre el conocimiento del código de programación que presenta. Asimismo, tiene la potestad de penalizar con puntos en contra si lo considera pertinente.

**9. Falta de probidad académica**

Estamos seguros de que cada estudiante del equipo de trabajo realizará el Trabajo Final, sin embargo, para evitar cualquier perspicacia, le recomendamos leer sus reglamentos de estudios y disciplina del estudiante. Las faltas contra la probidad académica, entendida éstas como la falta de honradez u honestidad académica que se manifiesta ante cualquier acción u omisión que podría permitir a un alumno salir beneficiado injustamente sobre los resultados de un trabajo académico y/o evaluación. Constituyen faltas contra la probidad académica las siguientes conductas:

**(i)** Intentar o realizar un plagio o copia total o parcial, al rendir una evaluación, ya sea de manera presencial o virtual, o al presentar un trabajo académico, usando cualquier medio, Internet, objeto o equipo para tal fin, en forma directa o valiéndose de terceros para ello.

**(ii)** Alterar, destruir o sustraer una evaluación o trabajo académico antes, durante o después de haberse rendido o entregado, aunque estuviese pendiente de calificación.

**(iii)** Presentar como propio el trabajo académico de otra persona o el desarrollado con otras personas o utilizarlo sin citar o reconocer la fuente original.

**(iv)** Falsear el trabajo intelectual, citando autores que no existen, refiriéndose a trabajos no realizados o tergiversando datos presentados como parte de un trabajo académico; o cometer cualquier otra acción similar que revele falta de honestidad.

**(v)** Presentar un mismo trabajo, en todo o parte, en la misma u otra asignatura, sin el consentimiento expreso de los profesores a cargo o director de Carrera.

**(vi)** Solicitar o recibir clases particulares, individuales o en grupo, remuneradas o no de docentes de la Universidad, sean o no profesores de las asignaturas en que el participante se encuentre matriculado, salvo autorización expresa de la Universidad.

**(vii)** La suplantación de identidad al momento de rendir cualquier evaluación o actividad académica, ya sea que se trate de una evaluación presencial o virtual.

**(viii)** El ofrecimiento en venta, la adquisición, acceso indebido y/o divulgación de los contenidos de cualquier evaluación académica previa a su aplicación.

**(ix)** Alterar, destruir o sustraer listas, registros de notas, calificaciones, certificados constancias y/o documentos académicos.

**(x)** Encargar u ofrecer el desarrollo de las evaluaciones online que aplica la UPC.

**(xi)** Mentir dolosamente sobre el grado, nivel académico y/o nivel de inglés que se ha logrado en la UPC.

**(xii)** Promover o divulgar información para acceder a beneficios y/o métodos que permitan generar ventaja en una evaluación, a través de cualquier medio.

**Instrucciones para la entrega del Trabajo Final terminado**

Enviar un archivo ZIP: TF\_CODIGO\_ALUMNO.ZIP

Este archivo debe contener la carpeta con:

* Código fuente de la aplicación
* Archivo de diagrama de clases en donde podrá usar: StarUML, LucidChart, Draw.io, etc.. en formato: PDF.
* Informe del proyecto según el formato indicado.
* Desarrollar un video donde explique el funcionamiento de la Aplicación Windows Form, deben de participar todos los integrantes del equipo de trabajo, el video debe durar máximo 5 minutos.
* Debe eliminar los archivos que no son necesarios, por ejemplo: temporal, debug, ocultos, etc.

**Evaluación del Trabajo Final**

**1. PRIMER HITO:**

Fecha: Semana 12 (Domingo 03 de noviembre hasta las 23:30 horas)

Objetivo:

* Entregar un documento en MS-WORD con la siguiente información
* Objetivo de las preguntas de Cultura General elegida por el equipo de trabajo.
* Diagrama de clases UML completo.
* Plan de actividades obligatorio (ver ANEXO 2).
* Recopilar y seleccionar todos los recursos de imágenes, *sprites*, backgrounds, room, textos, mensajes y demás recursos que utilizará en el desarrollo del videojuego.
* Secuencia del videojuego de acuerdo con el objetivo que pretende cumplir con el gameplay del juego Guardián del Medio Ambiente.
* Enviar el hito 1 al aula virtual.

**2. SEGUNDO HITO:**

Fecha: Semana 13 (Domingo 10 de noviembre hasta las 23:30 horas)

Objetivo:

* Haber terminado con el hito 1.
* El personaje(s) principal debe moverse haciendo uso del teclado o en forma automática sin salirse de los límites.
* Codificar la aparición de dos aliados como mínimo, uno por cada vez que presione una tecla.
* Codificar el funcionamiento de un enemigo.
* Enviar el hito 2 al aula virtual.

**3. TERCER HITO:**

Fecha: Semana 14 (Domingo 17 de noviembre hasta las 23:30 horas)

Objetivo:

* Terminar completamente el videojuego, eliminando cualquier defecto.
* El equipo de trabajo debe elaborar un video de 5 minutos donde muestre el funcionamiento del juego con las características solicitadas, todos los integrantes del equipo de trabajo deben participar en el video.
* Se debe entregar el Informe del Trabajo Final de manera individual, de acuerdo con el formato entregado por el docente.
* Para la exposición considerar:
* El equipo de trabajo contará con 10 minutos para presentar el videojuego más 5 minutos de preguntas por parte del docente.
* Cualquier retraso en la exposición será restado del tiempo de los estudiantes
* El docente es libre de cortar al estudiante si excede el tiempo destinado.
* El docente aplica lo estipulado en el apartado “Anexo 1”.
* Ingrese al aula virtual y suba su archivo.

Instrucciones para la entrega del HITO 3 de Trabajo Final terminado

* Enviar un archivo ZIP: TF\_CODIGO\_ALUMNO.ZIP
* Este archivo debe contener la carpeta con su proyecto de aplicación.
* Debe eliminar los archivos que no son necesarios.
* Se recomienda que las imágenes de los *sprites, audios, videos, archivos* no sean muy pesadas.

**ANEXO 1**

# Rúbrica de calificación

**Declaración:**

Durante la exposición, el grupo debe declarar si ha utilizado herramientas de IA y especificar en qué parte del trabajo las ha aplicado. Si el grupo usa alguna herramienta de IA y no lo declara, el docente tiene la potestad de penalizarlos si detecta su uso. Además, el docente puede solicitar a cada integrante que demuestre su conocimiento del código de programación presentado. Si algún estudiante no demuestra conocer su código, su calificación será penalizada.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Experto** | **Intermedio** | **Por mejorar** |
| **Exposición** | El estudiante expone con claridad y demuestra dominio del tema, respondiendo preguntas de manera eficaz. | El estudiante expone el tema, pero no de manera clara o completa. | El estudiante no demuestra dominio del tema y no expone con claridad. |
| 3 | 2 | 0 |
| **Diagrama de clases UML** | El diagrama de clases UML está completo, con todas las relaciones y clases implementadas correctamente. | El diagrama UML está incompleto o con errores en las relaciones entre clases. | No se presenta un diagrama de clases UML o está incorrecto. |
| 3 | 2 | 0 |
| **Programación Orientada a Objetos (POO)** | Utiliza correctamente clases, objetos, herencia y agregación, implementando buenas prácticas de POO. | Implementa POO pero con errores o de manera incompleta (falta herencia/agregación, etc.). | No utiliza POO correctamente, o implementa todo en un solo bloque sin objetos. |
| 4 | 2 | 0 |
| **Presentación del juego** | Se muestra una ventana de presentación con título y detalles del equipo. | Se muestra la ventana de presentación, pero con detalles incompletos o faltantes. | No se muestra una ventana de presentación del juego. |
| 3 | 2 | 0 |
| **Usabilidad de Software: Facilidad de jugar “Jugabilidad”.** | El personaje principal funciona correctamente y tiene todas sus funcionalidades implementadas (movimiento, acciones).  Los enemigos o fuentes de contaminación están implementados y funcionan de acuerdo con el diseño del juego. | El personaje principal funciona, pero con algunas limitaciones en sus funcionalidades. Los enemigos/fuentes de contaminación están implementados parcialmente o con errores. | No se implementó el personaje principal correctamente o no se puede controlar adecuadamente.  No se implementaron correctamente los enemigos o fuentes de contaminación, o no afectan al jugador como deberían. |
| 4 | 2 | 0 |
| **Ganador** | Se muestra la pantalla de "GANASTE" o "PERDISTE" cuando se cumplen las condiciones. | Se muestra el ganador, pero no según las condiciones establecidas. | No se muestra la condición de victoria o derrota del juego. |
| 1 | 0.5 | 0 |
| **Gestión de Archivos** | Se muestra la pantalla de "GANASTE" o "PERDISTE" cuando se cumplen las condiciones. | Se muestra el ganador, pero no según las condiciones establecidas. | No se muestra la condición de victoria o derrota del juego. |
| 2 | 1 | 0 |

**ANEXO 2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N°** | **Actividad** | **Responsable** | **Semana** | | | | | | **Estado** | **Observaciones** |
| **S10** | **S11** | **S12** | **S13** | **S14** | **S15** |
| 1 | Diseño del diagrama de clase | Ariana | X | X |  |  |  |  | Cumplido |  |
| 2 | Prototipo de la solución | Juan | X |  |  |  |  |  | En proceso |  |
| 3 | Definir interface del video juego | Pedro | X | X |  |  |  |  | Problemas | Definir resolución del formulario |
| 4 | Definir estructura de datos | Pedro | X |  |  |  |  |  | Problemas |  |
| 5 | Implementar herencia | Ariana |  | X |  |  |  |  | Cumplido |  |
| 6 | Clase controladora | Juan |  | X |  |  |  |  | Cumplido | No reconoce a la clase hijo |
| 7 | Elaborar el menú | Pedro |  |  | X |  |  |  | Problemas |  |
| 8 | Clases complementarias | Ariana |  |  | X |  |  |  | En proceso |  |
| 9 | Gestión archivos | Juan |  |  |  | x |  |  | Problemas | Problemas de integración, no actualiza |
| 10 | Integrar clases complementarias al proyecto | Juan |  |  |  |  |  |  | En proceso | No reconoce el método… |
| 11 | Pruebas integrales | Ariana |  |  | x | x |  | x | Problemas | Se tiene 3 errores no identificados |
| 12 | Redactar el informe | Juan |  |  |  | X |  | X | En proceso |  |
| 13 | Preparar la exposición | Pedro |  |  |  |  | X | X | En proceso |  |