

UNIVERSIDAD DE LOS ANDES
DEPTO. ING. ELÉCTRICA Y ELECTRÓNICA



Curso:	Sistemas Electrónicos Digitales - IELE2210		
Semestre:	2021-10		
Profesor:	Fredy Segura-Quijano	fsegura@uniandes.edu.co	
Asistentes:	Sebastian Sierra	s.sierra11@uniandes.edu.co	
	Jhoan Leon	je.leone@uniandes.edu.co	
	Ariana Paz	ja.paz@uniandes.edu.co	

PROBLEMA DE DISEÑO 1 (Grupos de 4) – (18 %)

PROYECTO: FROGGER

- Problema de diseño e implementación utilizando una solución tipo hardware.
-

1. MOTIVACIÓN PARA LA REALIZACIÓN DE PROYECTO

En el curso hasta este momento hemos identificado diferentes componentes básicos tanto combinacionales como secuenciales los cuales pueden hacer parte de un sistema más grande. Para dichos componentes hemos estudiado la forma de realizar una implementación hardware la cual se basa en la utilización de tecnologías como los dispositivos de lógica programable (p.e CPLDs, FPGAs). En este proyecto se propone la utilización de dichos componentes (de las bibliotecas) para dar solución a un problema con una complejidad mayor utilizando procesos estructurados de diseño. Se propone entonces identificar todas las especificaciones y restricciones del sistema a diseñar para abordar una metodología completa de diseño hasta llegar a una implementación física.

2. CONTEXTO DEL PROYECTO

“*Frogger*” es un videojuego publicado, originalmente como arcade, en 1981. La licencia para la distribución mundial fue de Sega/Gremlin y fue desarrollado por Konami. Frogger es un clásico de los videojuegos, que sigue siendo popular y del que pueden encontrarse muchas versiones en Internet.

El objetivo del juego es guiar una rana hasta su hogar. Para hacerlo, la rana debe evitar coches mientras cruza una carretera congestionada y luego cruzar un río lleno de riesgos. El jugador hábil obtendrá puntos adicionales a lo largo del camino.

Tiene un máximo de dos jugadores, que juegan alternadamente. Comienza con tres ranas. El jugador guía una rana que comienza en la parte inferior de la pantalla. Hay que cruzar una calle llena de vehículos, que en varias versiones incluyen coches, camiones, autobuses, taxis y/o motocicletas. Lo siguiente es un río lleno de troncos, lagartos y tortugas. En lo más alto de la pantalla están los cinco hogares de las ranas, las metas para cada rana. En cada nivel hay tiempo, así que el jugador debe llegar lo más rápido posible, antes de que el tiempo termine.

El único control que el jugador tiene son las direcciones de la palanca de joystick. Cada una hace que la rana vaya en esa dirección. Se debe guiar a la rana evitando ser atropellado en el camino, comido por los lagartos o ahogado en el río (la rana no sabe nadar). A mitad de camino, después de cruzar la calle, hay un lugar intermedio, donde a veces hay una serpiente, el jugador debe seguir para atravesar el río.

Brincando sobre los troncos o en los caparazones de las tortugas, el jugador debe guiar a la rana hasta los nidos vacíos. Se deben evitar lagartos, víboras y castores en el río pero puede comer insectos o acompañar una rana hembra para obtener un bonus. Cuando todos los nidos están llenos, el juego avanza al siguiente nivel, más difícil.



Figura 1: Imagen del juego Frogger.

Tomado de <http://www.paisdelosjuegos.es/juego/cruzar+la+calle/frogger.html>

3. DESCRIPCIÓN DEL PROBLEMA DE DISEÑO: ESPECIFICACIONES Y RESTRICCIONES

El objetivo de este problema de diseño; es diseñar e implementar un sistema digital basado en el funcionamiento básico del juego “*Frogger*”. El prototipo (Juego) a implementar usará de una matriz de LEDs de 8×8 donde los LEDs encendidos representarán un vehículo. Se debe implementar un juego con las siguientes características:

En la fila inferior de la matriz se encontrará un Frogger que será controlado y cuya posición puede ser modificada por el jugador. Así mismo, el escenario se desplazará a una razón de $\frac{1}{3}$ de segundo (debe variar para cada secuencia de vehículos).

El juego consta de mínimo 4 niveles y el usuario avanzará de nivel una vez lleve a 2 Froggers a su casa correspondiente.

En la fila superior al iniciar el juego todos los leds estarán prendidos excepto los correspondientes a las 2 casas de cada Frogger. En el momento que un Frogger llegue a una de sus casas, el led se debe encender y permanecer encendido indicando que ya hay un Frogger.

La cantidad de vehículos por fila del escenario dependerá del nivel al igual que el tamaño y la velocidad. Para el primer nivel existirán tres filas con vehículos de tamaño 2. La complejidad de los siguientes niveles será decidida por el grupo de trabajo siempre y cuando sea una complejidad que permita jugar y que sea mayor al nivel 1.

En el juego se utilizan 6 señales básicas de control que son, Reset, Start, Right, Left, Up, Down. Con la señal de Reset el juego volverá a un estado inicial a la espera de la señal de Start. Una vez detectada esta señal comienza el juego desde el primer nivel. Las señales de Right, Left, Up, Down controlan la

posición de Frogger.

El juego debe indicar al jugador cuando este pierde (Frogger es atropellado), cuando se cambia de nivel (cuando exitosamente los dos Froggers llegan a sus casas y cuando gana (si el usuario pasó exitosamente los 4 niveles). Se tendrá en cuenta la creatividad en la calificación.

4. CRONOGRAMA DE ENTREGAS

Conformación de grupos: El proyecto debe realizarse en grupos de **4 personas** (cada persona del grupo aporta 25 % del trabajo y un estudiante no puede aportar más de ese porcentaje). *No se aceptarán modificaciones en la conformación de los grupos durante el desarrollo del proyecto. Si alguna situación ocurre, los integrantes del grupo deben ser proactivos y proponer soluciones a dicha situación* Está permitido que los grupos de trabajo se formen independientes de la sección de laboratorio de cada estudiante. Los grupos deben inscribirse en la plataforma virtual en la sección de grupos a más tardar el **Viernes 9 de abril**. Este trabajo está pensado para ser desarrollado por 4 personas. Sea muy responsable en la elección de su grupo, establezca responsabilidades, cronograma de trabajo y sobretodo propóngase una meta y trabaje hasta conseguirla.

De acuerdo con el calendario académico de la universidad y teniendo en cuenta las experiencias previas del curso se estableció un sistema de entregas pensado en que todos los grupos logren obtener un sistema funcional en la semana de la entrega final. De esta manera, las fechas de las entregas y los requerimientos de cada una son:

1. **Viernes 9 de abril de 2021: ENTREGA EN LA WIKI:**
1 ESPECIFICACIONES. 1.1 Descripción del producto. 1.1.1 Cronograma de Actividades. 1.2 Diagrama de caja negra. 1.3 Macroalgoritmo general de solución.
2. **Viernes 16 de abril de 2021: ENTREGA EN LA WIKI:**
1.4 Arq del Sistema (PARTE CIRCUITAL): Diagrama de Bloques, Señales e Inter-conexiones. 1.5 Arq del Sistema (PARTE HARDWARE): Diagrama de Bloques, Señales e Inter-conexiones. 1.5.1 Bloque X. 1.6 Arq del Sistema: (PARTE SOFTWARE): Funciones. 1.6.1 Función X.
3. **Viernes 21 de abril de 2021: ENTREGA EN LA WIKI:**
2 REPORTE TÉCNICOS. 2.1 Memorias de Cálculo. 2.2 Definición de vectores de prueba y simulaciones parciales de respaldo. 2.3 Indicadores de utilización de recursos y rendimiento de los dispositivos utilizados.
4. **Viernes 30 de abril de 2021: ENTREGA EN LA WIKI Y EN CLASE:**
3 IMPLEMENTACIÓN. 3.1 Descripción en lenguajes HW y SW (Códigos fuente). 3.1.1 Dispositivo 1. 3.1.2 Dispositivo 2. 3.2 Funcionalidad modular. 3.2.1 Vídeos y fotos de demostración de módulos del sistema. 3.3 Funcionalidad global del sistema. 3.3.1 Vídeos y fotos de demostración del prototipo final. 4 REFLEXIÓN. 4.1 Resultados y lecciones aprendidas 5 MATERIALES. 5.1 Dispositivos Hardware 5.2 Herramientas Software 6 BIBLIOGRAFIA

No olvide realizar cuantas simulaciones considere necesarias para comprobar que su sistemas cumple con todas y cada un de las especificaciones y restricciones del diseño. Plantee distintos escenarios donde se observe un flujo completo de la información. Siempre tenga en cuenta el peor de los escenarios posibles. De igual forma simule cada uno de los módulos que componen el sistema. (Sugerencia: Utilice la sentencia *assert* en la simulaciones. Investigue y describa su uso.). Muestre y explique claramente los resultados obtenidos en la simulación para cada caso. Para ello, use varios marcadores que muestren cada resultado de la simulación y explíquelos.

5. RÚBRICAS DE CALIFICACIÓN

VER MATRIZ DE CALIFICACIÓN.