

Tecnológico de Costa Rica
Área de Ingeniería en Computadores
Algoritmos y Estructuras de Datos II (CE2103)



Primer Semestre 2015

Proyecto Diagnóstico

>> CrazyDuckHunt <<

Elaborado por:

Ariel Piedra.

Arturo Chinchilla.

Juan Navarro.

Malcolm Davis.

Grupo: 02

Profesor:

Marco Rivera M.

Segundo Semestre 2015

Introducción.

En el siguiente documento presentamos una descripción técnica y detallada de las características del Proyecto Crazy Duck Hunt, desarrollado en usando el lenguaje de programación C++ y usando GNU Linux como Sistema Operativo.

El proyecto Crazy Duck Hunt consiste en desarrollar un juego basado en un clásico de NES llamado "Duck Hunt". Nuestro juego se encuentra organizado en dos secciones, que deben funcionar independientemente, pero integrándolas finalmente.

La primera sección consiste en crear un juego para computadora, donde los usuarios se puedan conectar dependiendo de lo que quieran hacer y de las credenciales que los mismos tengan, un usuario se puede conectar al juego en condición de jugador, que crea una instancia del juego y puede interactuar con todos los objetos del mismo; de visualizador, se conecta a una instancia del juego previamente creada por otro usuario, de esta forma puede observar todo lo que pasa en el juego, pero no puede interactuar con los objetos del mismo; de administrador, que tiene control total sobre todas las instancias del juego creadas por los otros usuarios.

Temas a investigar

XML : Parser XML para cargar los archivos de configuración y crear instancias de juegos "LogicGame Editables/Configurables".

Programación C/C++ Lenguaje de programación con el cual se desarrolla la aplicación del Juego.

Pthreads : Bibliotecas de hilos de ejecución en c/c++ que abstraen la programación paralela para poder implementar el movimiento de los patos y la comunicación con el juego de forma simultánea.

Qt GUI c++: Bibliotecas para el desarrollo de una interfaz Gráfica la cual pueda servir como conexión entre el jugador y la aplicación.

NCurses: Alternativa de interfaz gráfica que inicialmente es usada como forma viable de capturar caracteres del teclado en para pruebas de captura de disparos.

UnReal Engine : Motor de gráficos optimizado para el desarrollo de juegos en 3D.

Unity Engine: Alternativa de motor de gráficos de juegos en 3d.>

Server/Client Socket: Comunicación mediante Sockets en C/C++ para abstraer el envío de datos desde el GameCenter a los GamePlayers y los GameViewers.

Boost THREAD: Librería de Hilos que implementan Pthread C/C++, alternativa en caso de que los hilos no funcionen.

Manejador de Versiones de Código:

Asamblea (SVN): Manejador de versiones de código que debe ser implementado junto un plug-in en Netbeans el cual se puede observar gráficamente los cambios en el editor y permitiendo regresar a versiones previas del código del que se llevaba seguimiento. Como requerimiento de la tarea se implementa inicialmente "Asamblea".

Bitbucket (GIT): Un manejador de código distribuido que ofrece mejores ventajas que SVN es Git, el cual permite tener una versión final del proyecto en cada uno de los ordenadores que trabajan colaborativamente, es decir Git trabaja de manera distribuida.

CGI/C++ Web Applications: Es una forma de realizar aplicaciones que sean independientes de la gui y útiles para realizar proyectos de aplicaciones multi-cliente accediendo desde la dirección IP dentro de una red de área local mediante el buscador.

HTML5: Ya que es parte de los requerimientos de la tarea, HTML será útil como interfaz gráfica que mostrará el juego, con funcionalidades de manejo de gráficos y de eventos para que el jugador interactúe con el juego.

CSS: Se debían de utilizar los estilos y diseño de la página, CSS es una forma de personalizar la página para que se pudiera llevar a cabo el diseño del juego así como ciertas funcionalidades.

JSON: Es un protocolo de información más sencillo de utilizar que XML, y este es una forma de intercambiar información entre el Servidor y los clientes.

Apache Server para Linux: Una aplicación que permite crear un server dentro de una red como forma de comunicación entre máquinas con una red de área local.

Diseño de la solución:

Diseño 1:

Inicialmente se tenía una aplicación desarrollada en C++ que se llamaba GameCenter, esta aplicación la aplicación que tenía tres funciones macro importantes:

- 1) Creación y administración de las instancias de los juegos, llamadas GameLogic.
- 2) Control de las estadísticas de los jugadores para cada juego.
- 3) Creación de perfiles de visualizador, jugador y administrador los cuales se iba a manejar su información para su respectiva comunicación.

Los componentes de GameCenter son los siguientes:

Una estructura de datos, que no se tiene definida, contendrá todas las instancias de los juegos de CrazyDuckHunt.

ServerGame: una aplicación desarrollada en C++ que implementará la función de comunicación mediante Sockets a cada uno de los clientes.

Interpreter: Peticiones desde diferentes tipos de clientes iban a provenir al ServerGame, por lo tanto se pretende abstraer una aplicación que maneje las diferentes peticiones de los Visualizadores, Administradores, y Jugadores.

Una estructura de datos que mantenga relacionado cada uno de los Jugadores con su respectiva instancia y Visualizadores. Además de otra estructura que almacene a aquellos clientes cuyo rol es el de administrar.

Un XML parser que se encargaba de leer las configuraciones iniciales y de arranque del GameCenter.

La siguiente figura describe cómo estos elementos comprenden la versión del diseño inicial:

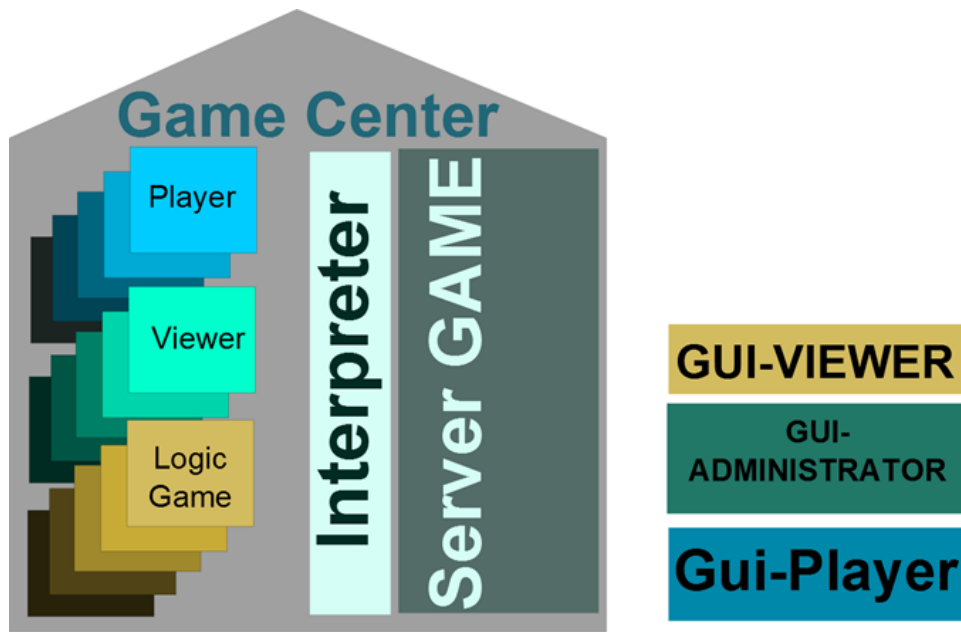


Figura 1: GameCenter, aplicación central donde se tienen las instancias de los juegos.

Las aplicaciones Gui-Viewer,Gui-Administrator y Gui-Player son aplicaciones del lado del cliente, por otro lados recuadros LogicGame,Viewer,Player son la forma de organización y relación de cada uno de estos componentes.

Cambios al diseño:

- 1) Se piensa implementar las instancias de los juegos corriendo desde el lado del cliente, es decir GameCenter no tendrá dentro de su ejecución la ejecución de cada una de las instancias, pero sí tendrá acceso de control y seguimiento mediante las interfaces de Interpreter y ServerGame.
- 2) Se tenía planeado implementar el patrón de diseño Abstract Factory por una función que mediante un número aleatorio se obtengan la creación de los patos.
- 3) Se abstraen las características de los patos a una clase madre Pato, ya que las existían comportamientos en uno tipos que no eran iguales en otros.
- 4) LogicGame se tenía implementado mediante una abstracción de las funciones de Pthread, las cuales fueron substituidas por Boost, que son una implementación más amigable que Pthread, aunque Boost es en realidad Pthread.

5) Se encuentra una alternativa para evitar el uso de una aplicación extra del lado del cliente es decir, en el mejor de los casos se utilizaría HTML5 y un Web Browser, por lo que se retorna a parte del diseño inicial con aplicaciones sin GUI corriendo del lado del servidor y la interacción del cliente mediante un web browser cuya máquina se encuentra conectada a una red de área local.

Diagramas de Clase



Descripción de Bibliotecas y Funciones utilizadas

Netinet/in define prototipos, macros, variables y la estructura `sockaddr_in` usada con los sockets de dominio de Internet.

Netdb contiene definiciones de datos para las rutinas de biblioteca de red. Define las siguientes estructuras:

- hostent y hostent_data
- netent y netent_data
- servent y servent_data
- protoent y protoent_data

Arpa/inet define prototipos para aquellas rutinas de la biblioteca de red que convierten la dirección de Internet y la notación decimal con puntos.

Boost/thread permite el uso de múltiples hilos de ejecución con los datos compartidos en código portable de C ++. Ofrece clases y funciones para la gestión de los hilos mismos,

junto con otros para la sincronización de datos entre los hilos o proporcionar copias separadas de los datos específicos de hilos individuales.

Stdio, que significa "*standard input-output header*" (cabecera estándar E/S), es el archivo de cabecera que contiene las definiciones de las macros, las constantes, las declaraciones de funciones de la biblioteca estándar del lenguaje de programación C para hacer operaciones, estándar, de entrada y salida, así como la definición de tipos necesarios para dichas operaciones. Por motivos de compatibilidad, el lenguaje de programación C++ (derivado de C) también tiene su propia implementación de estas funciones, que son declaradas con el archivo de cabecera `cstdio`.

Stdlib (*std-lib: standard library* o biblioteca estándar) es el archivo de cabecera de la biblioteca estándar de propósito general del lenguaje de programación C. Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras. Es compatible con C++ donde se conoce como `cstdlib`.

Unistd permite el acceso a la API del sistema operativo POSIX (Portable Operating System Interface). Se define por el estándar POSIX.1, la base de la Single Unix Especificación (nombre colectivo de una familia de estándares para sistemas operativos de computadora), y por lo tanto debe estar disponible en cualquier conformación (o cuasi-conforme) del sistema operativo / compilador (todas las versiones oficiales de UNIX, incluyendo Mac OS X, GNU / Linux, etc.).

String, esta biblioteca contiene el manejador de cadenas de texto, además de un grupo de funciones que implementan operaciones sobre datos de tipo String.

iostream es un componente de la biblioteca estándar (STL) del lenguaje de programación C++ que es utilizado para operaciones de entrada/salida (recibir datos del teclado y/o mouse por ejemplo). Su nombre es un acrónimo de Input/Output **Stream**. El flujo de entrada y salida de datos en C++ (y su predecesor C) no se encuentra definida dentro de la sintaxis básica y se provee por medio de librerías de funciones especializadas como `iostream`.

time es una biblioteca estándar del lenguaje de programación C que contiene funciones para manipular y formatear la fecha y hora del sistema.

sys/socket contiene la cabecera que definirá la estructura `sockaddr_storage`. Esta estructura será:

- Lo suficientemente grande como para dar cabida a todas las estructuras de direcciones específicas del protocolo.
- Alineado en un límite adecuado para que los punteros puedan ser casteados como punteros a estructuras de direcciones específicas del protocolo y se utilizan para acceder a los campos de esas estructuras sin problemas de alineación.

sys/types incluye definiciones para varios tipos, por ejemplo:

- blkcnt_t: Se utiliza para el recuento de bloqueo de archivos
- blksize_t: Se utiliza para tamaños de bloque
- clock_t: Se utiliza para los tiempos del sistema.

Vector es una clase template que funciona perfectamente para reemplazar los arrays de estilo C. Permite la misma sintaxis natural de los arrays pero ofrece una serie de servicios que libera al programador de preocuparse por la memoria asignada y ayuda a operar constantemente sobre los objetos contenidos.

Pugixml es una biblioteca liviana de C++ para el procesamiento de XML. Consiste de una interfaz tipo DOM con grandes capacidades de modificación. Es una biblioteca que cuenta con un rápido parser de XML el cual construye el árbol de DOM desde un archivo de XML. Esta es biblioteca muy portable y de fácil usabilidad e integración.

Qt es una biblioteca multiplataforma ampliamente usada para desarrollar aplicaciones con interfaz gráfica de usuario, así como también para el desarrollo de programas sin interfaz gráfica, como herramientas para la línea de comandos y consolas para servidores.

Descripción de las estructuras de datos desarrolladas.

Vectores:

Un vector es una estructura de datos que permite almacenar un conjunto de datos del mismo tipo. Los datos contenidos dentro de un vector pueden ser accedidos mediante índices de posición. Nuestros vectores, pertenecen a la clase vector que contiene una gran variedad de métodos para trabajar con ellos. En este proyecto se utiliza este tipo de estructura para almacenar las instancias de los patos creados en el juego y con ello tener acceso y control sobre las mismas.

Problemas encontrados. (Descripción detallada, intentos de solución sin éxito, solución encontrada detallada, recomendaciones, conclusiones y bibliografía consultada para ese problema en específico).

- Para la lectura de datos desde el XML se comenzó investigando sobre la biblioteca Xerces, y por falta de información se busca una alternativa. Se pregunta a estudiantes que ya han pasado el curso de Algoritmos y Estructuras de Datos II y como recomendación se recibe que se utilice la biblioteca pugixml, la cual contiene métodos muy simples y potentes para el manejo de archivos xml.
- Para el manejo de los procesos del servidor, se comenzó trabajando con la biblioteca Pthreads, mas sin embargo la forma de manejo del multi-threading se volvió muy complicada, ya que el manejo de threads por separado solo se permite utilizando la biblioteca Mutex. Como alternativa a Pthreads se usó Boost.Threads, que tiene una implementación más sencilla para el manejo individual de los threads.
- Para implementación de Boost.threads requiere una configuración adicional en el IDE. <https://stackoverflow.com/questions/9723793/undefined-reference-to-boostsystemsystem-category-when-compiling>

Conclusiones y recomendaciones del proyecto.

- **Assembla** (SVN) no resultó muy práctico para el versionamiento del código, además de que sus plugins para los IDE no resultaron muy fáciles de instalar y utilizar. Se recomienda utilizar otra herramienta para el control de versiones.
- Es muy importante hacer una adecuada **planeación de actividades** a realizar, así como la **asignación** de las mismas, para evitar la sobrecarga de trabajo en algunos pocos integrantes del grupo. Se recomienda planificar mejor el tiempo y las actividades a realizar.
- La implementación de los threads es más sencilla con la **biblioteca Boost** que con Pthreads, se recomienda la utilización de la primera y evitar complicaciones.
- Se determina que la conexión entre **C++ y HTML** es de suma complejidad, además de la falta de documentación exacta y apropiada. Se recomienda investigar más a fondo y documentar alguna posible solución.
- Unreal Engine tiene **requisitos mínimos muy altos**, a menos de que se cuente con una computadora muy potente es prácticamente imposible utilizarlo.
- Unity 3D **no tiene compatibilidad** con sistemas basados en Linux y si se está trabajando en alguna de estas versiones de SO es imposible utilizarlo.

Bibliografías

C++ Tutorial: A Beginner's Guide to std::vector, Part 1 (s. f.). Recuperado de http://www.codeguru.com/cpp/cpp/cpp_mfc/stl/article.php/c4027/C-Tutorial-A-Beginners-Guide-to-stdvector-Part-1.htm

C string handling - Wikipedia, the free encyclopedia (s. f.). Recuperado de https://en.wikipedia.org/wiki/C_string_handling

Chapter 30. Thread 4.5.0 - 1.58.0 (s. f.). Recuperado de http://www.boost.org/doc/libs/1_58_0/doc/html/thread.html

Deitel, H. & Deitel, P. (s. f.). *C++ How to Program* (8va ed.)

Felton, M. (s. f.). *CGI: Internet Programming in C++ and C*

Iostream - Wikipedia, la enciclopedia libre (s. f.). Recuperado de <https://es.wikipedia.org/wiki/Iostream>

Josuttis, N. (s. f.). *The C++ Standard Library: A Tutorial Reference*

POSIX - Wikipedia, the free encyclopedia (s. f.). Recuperado de <https://en.wikipedia.org/wiki/POSIX>

Pugixml 1.6 quick start guide (s. f.). Recuperado de <http://pugixml.org/docs/quickstart.html>

Qt (biblioteca) - Wikipedia, la enciclopedia libre (s. f.). Recuperado de [https://es.wikipedia.org/wiki/Qt_\(biblioteca\)](https://es.wikipedia.org/wiki/Qt_(biblioteca))

Single UNIX Specification - Wikipedia, the free encyclopedia (s. f.). Recuperado de https://en.wikipedia.org/wiki/Single_UNIX_Specification

Socket header files (s. f.). Recuperado de <http://publib.boulder.ibm.com/html/as400/v4r5/ic2979/info/RZAB6SOCKHF.HTM>

Stdio.h - Wikipedia, la enciclopedia libre (s. f.). Recuperado de <https://es.wikipedia.org/wiki/Stdio.h>

Stdlib.h - Wikipedia, la enciclopedia libre (s. f.). Recuperado de <https://es.wikipedia.org/wiki/Stdlib.h>

Sys/socket.h (s. f.). Recuperado de

<http://pubs.opengroup.org/onlinepubs/009695399/basedefs/sys/socket.h.html>

Sys/types.h (s. f.). Recuperado de

<http://pubs.opengroup.org/onlinepubs/007908775/xsh/systypes.h.html>

Time.h - Wikipedia, la enciclopedia libre (s. f.). Recuperado de <https://es.wikipedia.org/wiki/Time.h>

Unistd.h - Wikipedia, the free encyclopedia (s. f.). Recuperado de

<https://en.wikipedia.org/wiki/Unistd.h>

CGI Programming in C++ - C/C++ Tutorials - Codecall (s. f.). Recuperado de

<http://forum.codecall.net/topic/72818-cgi-programming-in-c/>