

Bienvenidos a iCE Climber

iCE Climber es una adaptación de un juego de NES de llamado Ice Climber.

Desarrolladores:

David Pereira Jiménez

Randall Samuel Méndez Loría

Jonathan Gonzalez Sánchez

Bibliografía

- [Socket Programming Multiple Clients](#)
- [TCP Client in C Windows](#)
- [TCP Server in C Windows](#)
- [GSON Github Page](#)
- [cJSON Github Page](#)
- [Gnome GTK3+](#)

Bitácora

Bitácora David

Actividad	Descripción	Fecha
Primera Reunión	Se estudia el enunciado y se asignan las actividades correspondientes a cada compañero	8/11/2019
Inicia implementación lógica	Se comienza a definir en papel objetos requeridos y acciones a implementar	12/11/2019
Comienza el desarrollo del código	Empieza el desarrollo en código, se eliminan objetos que no se consideran necesarios	14/11/2019
Segunda Reunión	Se aclaran dudas sobre el avance del proyecto, y se definen nuevas tareas	18/11/2019
Se plantea nueva propuesta lógica	Se comienza a encapsular procesos lógicos en una clase específica fuera del main	19/11/2019
Tercera Reunión	Se organiza cierre del proyecto, tareas prioritarias, y tiempo a dedicar	21/11/2019
Reunión final	Reunión para acoplar el proyecto y finalizar procesos restantes	22/11/2019 - 23/11/2019

Bitácora Jonathan

Actividad	Descripción	Fecha
Primera Reunión	Se estudia el enunciado y se asignan las actividades correspondientes a cada compañero	8/11/2019
Investigación sockets	Se investiga implementación de sockets tanto en C como en Java	11/11/2019
Se inicia implementación en C	Considerando C más complicado se procede a comenzar por su implementación con casos de prueba	12/11/2019
Casos de prueba para Java	Se inicia con los casos de prueba de Java y una implementación inicial de los sockets para el proyecto	15/11/2019
Segunda Reunión	Se aclaran dudas sobre el avance del proyecto, y se definen nuevas tareas	18/11/2019
Se comienza a implementar funciones prioritarias	A partir de la reunión se determina información prioritaria a transferir y se procede a la implementación	19/11/2019

Tercera Reunión	Se organiza cierre del proyecto, tareas prioritarias, y tiempo a dedicar	21/11/2019
Reunión final	Reunión para acoplar el proyecto y finalizar procesos restantes	22/11/2019 - 23/11/2019

Bitácora Samuel

Actividad	Descripción	Fecha
Primera Reunión	Se estudia el enunciado y se asignan las actividades correspondientes a cada compañero	8/11/2019
Investigación interfaz	Se inicia investigando sobre alternativas para implementar interfaz en C, se opta por utilizar GTK	9/11/2019
Desarrollo inicial de la interfaz	Se implementa una ventana básica con botones funcionales, y ventanas emergentes	12/11/2019
Dibujado de elementos	Se logra dibujar distintos elementos sobre la interfaz a partir de sprites	15/11/2019
Segunda Reunión	Se aclaran dudas sobre el avance del proyecto, y se definen nuevas tareas	18/11/2019
Movimiento jugadores	Se implementa el control de los jugadores, tanto horizontal como vertical	20/11/2019
Tercera Reunión	Se organiza cierre del proyecto, tareas prioritarias, y tiempo a dedicar	21/11/2019
Reunión final	Reunión para acoplar el proyecto y finalizar procesos restantes	22/11/2019 - 23/11/2019

Conclusiones y Recomendaciones

Conclusiones

En cuanto a la programación en C se puede destacar su utilidad a la hora de resolver problemas determinados a través de pasos específicos; dada su naturaleza imperativa, C facilita la obtención de una solución mediante un algoritmo definido, lo cual supone una ventaja para el programador como tal a la hora de enfrentar distintos retos. Mientras que orientación de objetos, dado que modela el mundo con objetos de la vida real, resulta aún más accesible plasmar un problema de la vida real en código, esto a través del uso de las clases como moldes y la herencia para compartir información entre las clases que comparten características y atributos similares.

Recomendaciones

Para adentrarse en la programación en C es importante repasar el funcionamiento de los punteros como tal, ya que el riesgo de actualizar de forma errónea una ubicación siempre está latente. Por otro lado, para iniciarse en POO, la comprensión de los conceptos de herencia, polimorfismo y encapsulamiento son vitales, pues la solución de muchos problemas en programación radica en el uso de estos tres.

Descripción de las estructuras de datos desarrolladas

Listas enlazadas

Las listas enlazadas dentro de este proyecto destacan en varios aspectos, por un lado se utilizaron para encolar las diferentes instancias de los clientes que se vayan a presentar y así poder manipularlas con comodidad, se aprovecha la calidad dinámica de su tamaño y su ordenamiento para manejar cada ingreso nuevo de clientes inesperados. Además, se utilizaron a nivel de interfaz para contener los hijos (o childrens) que poseen los distintos widgets permitidos por la librería de GTK utilizada. La relación o conexión entre los elementos de este tipo de listas fue aprovechada en este punto, ya que era necesario que los hijos de un mismo widget se mantuvieran relacionados en todo momento de alguna manera.

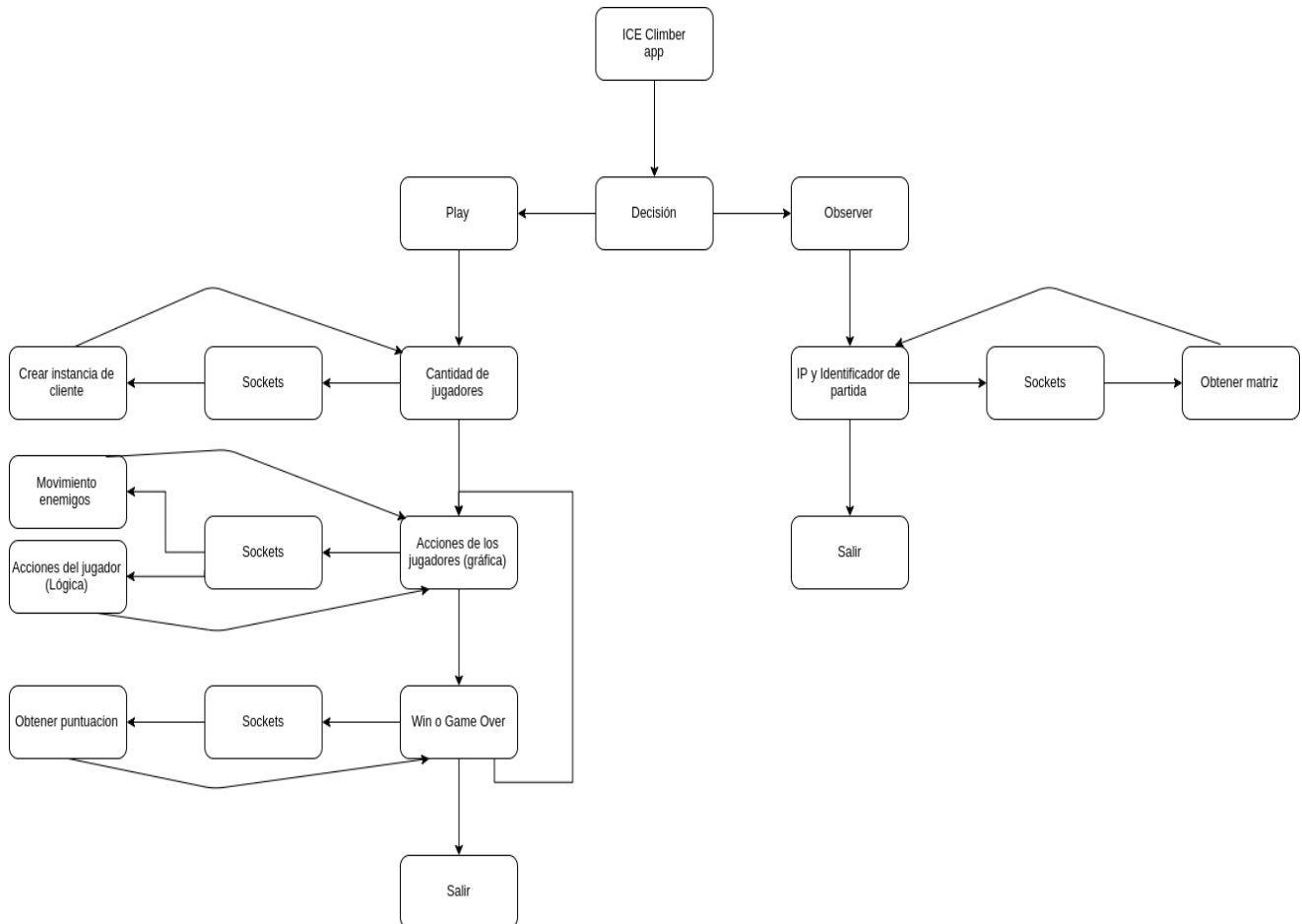
Matrices

Esta es una de las estructuras más importantes, se utilizó una matriz para manejar la interfaz gráfica del juego. La matriz de dimensiones 110x32, es la encargada de almacenar la distribución de los elementos a lo largo de la interfaz, esto mediante la utilización de enteros como identificadores, según sea el identificador que se encuentre en el espacio ese será el objeto colocado en la ubicación correspondiente de la pantalla. Se utilizan números como el 0 para espacios vacíos y del 1 al 6 para representar bloques, siendo los impares destructibles y los pares indestructibles. Además, del 7 al 10 representan enemigos y del 11 al 14 frutas y/o verduras.

Arreglos

Los arreglos también toman un lugar relevante dentro del proyecto, estos son los encargados de realizar la transferencia de información entre la lógica del juego y la interfaz a través de los sockets. Los arreglos contienen la información a enviar y recibir, mientras que los sockets se encarga de realizar el parsing de estos. Los datos almacenados por estos arreglos abarcan desde identificadores hasta los distintos scores, pasando por las posiciones de los diferentes objetos del juego.

Descripción detallada de los algoritmos desarrollados



Manual de Usuario

iCE Climber

Requisitos

Sistema Operativo: Windows 7 o mayor.

C, Java 10, GTK 3

Instrucciones de inicio

Se pueden tomar dos posibilidades:

- Play
 - Se decide cuantos jugadores van a existir.
- Observer
 - Se ingresa el IP y un identificador de partida para poder conectar a la partida que se quiere observar.

Instrucciones de juego

- Si se elige un Observer:
 - No se puede hacer nada mas que observar la partida.
- Si se elige jugar:
 - El jugador 1 se puede mover con las teclas W,A,S,D.
 - El jugador 2 se puede mover con las flechas del teclado.

El jugador se mueve de derecha a izquierda o viceversa.

Se salta con el fin de destruir un bloque arriba o para subir de piso.

Se puede atacar, para eliminar a los enemigos (Focas, Pajaros).

A partir de piso 9, se considera zona de bonus.

En la zona de bonus no hay enemigos, pero si hay frutas las cuales se pueden obtener para aumentar el puntaje.

Para ganar se tiene que tocar el Pterodactilo.

Si el jugador cae por un hueco pierde una vida, si el jugador cae en la zona de bonus, perdera un vida y ira a la pantalla de puntajes.

Instrucciones de puntaje

En esta pantalla se puede ver el desglose de los puntajes por jugador.

El boton de "Re-play" se encargara de generar una nueva pantalla de juego.

El boton de "Salir" va a terminar el programa.

Plan de Actividades realizadas por estudiante

Actividad	Descripción	Tiempo	Responsable
Interfaz	Desarrollo de la interfaz con la que el usuario interactuará.	9 días	Randall
Logica	Desarrollo de la logica del juego.	7 días	David
Sockets	Desarrollo de los Sockets entre java (servidor) y c (cliente).	8 días	Jonathan
Fecha de entrega	DeadLine	23/11 a las 11:59 a.m.	Randall, David, Jonathan

Problemas conocidos

Salto imprevisto

Si el usuario encuentra que no hay un bloque en el que se pueda parar, comienza a caer. Pero si encuentra un "hueco" mientras camina y en el proceso de caída salta, esta acción se va a generar cuando no podría.

Problemas encontrados

Pasar de Linux a Windows la interfaz

C como tal no tiene ninguna diferencia entre Linux y Windows, pero al entrar a las librerías se encontró de que la librería *GTK3*, la cual es escargada de la interfaz, se especializa a Linux y las librerías de los Sockets usada se basa en Windows, por lo cual se tenía que pasar la interfaz, la cual se estaba trabajando en Linux, a Windows para que los Sockets se pudieran acoplar. Después de investigar se encontró la respuesta en la página ([Building a CLion + GTK3 environment under Windows](#)) la cual nos une CLion, en donde se están trabajando los Sockets, a *GTK3*.

Tamaño de las transmisiones TCP

Debido a la naturaleza del protocolo TCP, los mensajes que se envían por medio de este protocolo, deben avisar de antemano cuál será la longitud de su mensaje por medio del socket, así se puede hacer un handshake. Si esto no se verifica el programa pondrá todo en espera para leer del socket.