



# LIBERIA PARA JUEGOS DE MESA SAEM GAMES

SANTIAGO QUINTERO, EDGAR AMÉZQUITA Y CAMILO ARÉVALO  
PONTIFICIA UNIVERSIDAD JAVERIANA



## OBJETIVOS

**Saem Games** es un libreria para crear juegos de mesa que tengan un tablero, debe soportar 2 o mas jugadores, Este proyecto debe ser implementado en C++ y debe contar con su propia documentacion para el proyecto de objetos y programacion a media escala.

## INTRODUCCION

Para este proyecto se debia crear una librería que ayudará al usuario a crear algunos juegos de mesa como escalera, monopoly y otros juegos que maneje un tablero para jugar, la libreria no cubre todo tipo de juegos, tales como risk. Esta libreria nos permitira extender varias de sus clases para ajustarlas a un tipo específico de juego y agregar la lógica al juego.

Para probar la funcionalidad de la libreria se realizo un juego de escalera con power ups, en el cual, a diferencia del escalera normal, hay casillas que le dan ciertos poderes a una ficha de un jugador.

## HERRAMIENTAS DE DESARROLLO

En este proyecto se hizo uso de:

1. Git Hub como manejador de versiones del proyecto.
2. Doxygen para documentacion de codigo.
3. C++ 11 para la implementacion de la libreria.
4. Latex para la realizacion del poster.



## ANALISIS

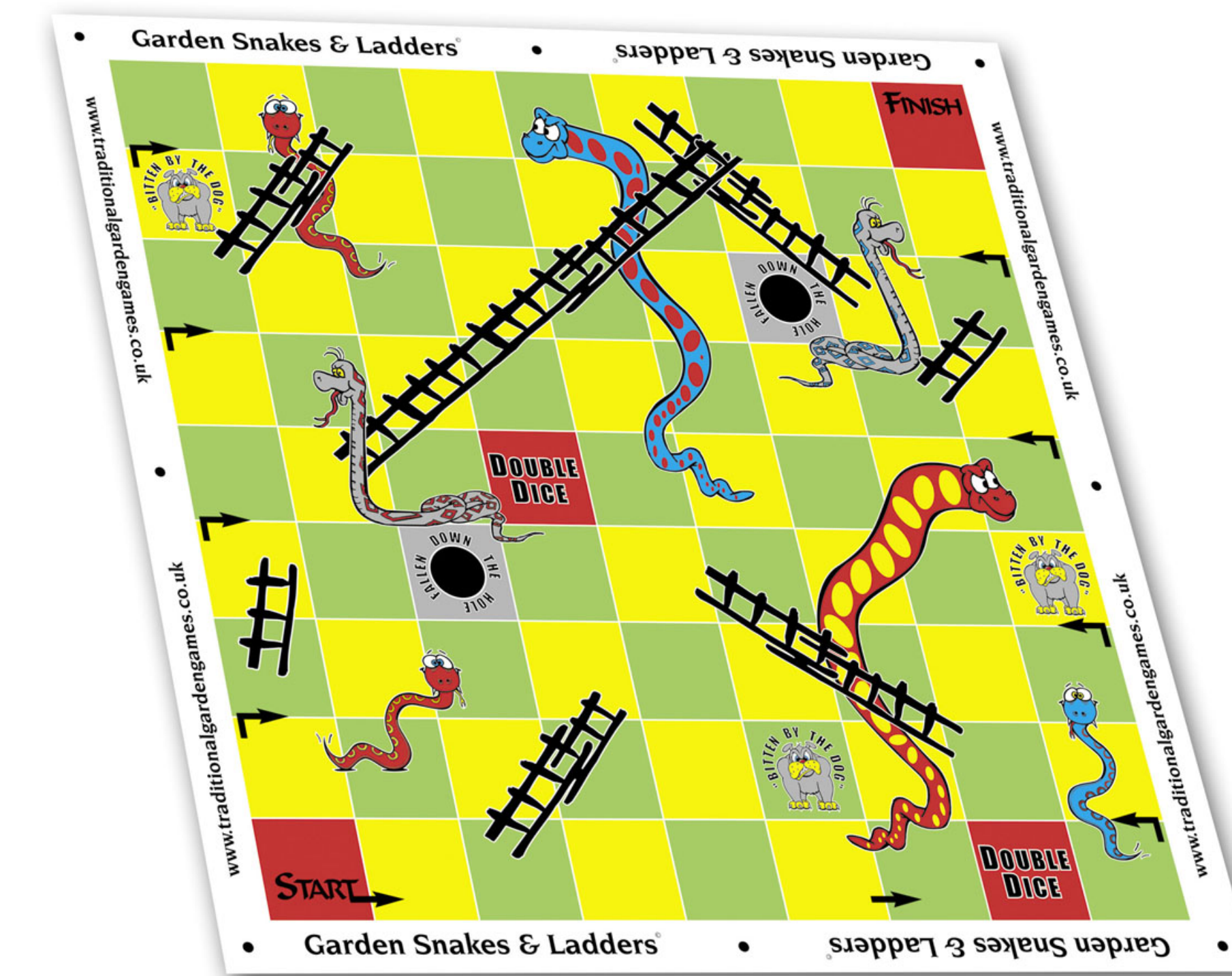
La libreria debe soportar la creacion de diversos juegos de mesa, por tanto la libreria debe poseer objetos que modelen un juego, como tablero, jugadores, fichas, cartas y dados. La libreria contendra funciones basicas para un juego, las cuales se pueden extender dependiendo la complejidad del juego.

Para muchos juegos un jugador puede poseer varias fichas en un juego, las cuales pueden ser manipuladas agregando, quitando o moviendo las fichas en el tablero. Ademas que el movimiento de la ficha debe determinarse por un numero constante relacionado a una regla o ser fijado por el resultado de uno o mas dados contenidos en el juego.

No nesariamente debe haber dados en un juego, pero en el caso de tenerlos, el dado debe generar un numero aleatorio entre un rango especificado por el usuario.

## CONCLUSION

Como conclusion podemos ver un juego de escalera con power ups(poderes) con esta libreria, la cual usa cartas que te retroceden o avanzan segun el tiro del dado.



## RESULTADO

Como resultado podemos ver una libreria con todos sus componentes para poder crear un juego de mesa:

- una clase para las cartas (**SCarta**), Se definen las funciones para modelar lo basico de una carta, como colocar un nombre a una carta u obtener el nombre de la carta.
- una clase para las casillas (**SCasilla**), la clase define funciones para asignar o obtener los atributos de casilla como las cartas o fichas, cuenta con varios constructores para hacer mas facil su construccion.
- una clase para los dados (**SDados**), Con esta clase se puede conseguir la funcionalidad de un dado, genera numeros aleatorios entre un numero entero minimo y uno maximo inclusivo.
- una clase para las fichas (**SFicha**), La clase define las funciones para modelar lo basico de una ficha como asignar a un jugador, conseguir o asignar la casilla donde esta la ficha.



- una clase para el juego en general (**SJuego**), Esta Clase es un clase semi abstracta porque toca implementar varias funciones, para que este pueda usarla y se pueda jugar.
- una clase para los jugadores (**SJugador**), La clase define las funciones basicas para modelar un jugador, este tiene fichas, un nombre y pertenece a un equipo. Se puede asignar un equipo, fichas, saber a que equipo pertenece y retirar el jugador de ser necesario.
- una clase para las pila de cartas (**SPile\_cartas**), Aqui se implementan las funciones basicas de una pila de cartas, como push(), pop(), size() y shuffle().
- una clase para reglas (**SRegla**), Es una clase abstracta la cual hay que implementar para poder poner las reglas del juego, que generalmente son reglas de que pasa con las fichas.
- una clase para el tablero (**STablero**), Esta clase implementa funciones para conseguir o agregar cartas y fichas.