

Desarrollo e implementación de La Conquista en Smalltalk

^aBader, Cain; ^bBernal, Tiago; ^cGonzález, Micaela; ^dQuiróz, Marina; ^eSuárez, Gonzalo

^aU.T.N F.R.Re, ISI B, cain.bader@gmail.com

^bU.T.N F.R.Re, tiago.bernal28@gmail.com

^cU.T.N F.R.Re, micagonza4832@gmail.com

Resumen

Se presenta el desarrollo del juego La Conquista en el lenguaje de programación SmallTalk y su implementación en Pharo. Esto se lleva a cabo con el uso del paradigma de programación orientado a objetos. Se detalla la metodología realizada para llevar a cabo el proyecto. Para ello se desarrolla una explicación de las clases y métodos utilizados para crear un juego dinámico, interactivo e innovador.

1. Introducción

Este documento describe el proceso de diseño, desarrollo e implementación del juego 'La Conquista' en Smalltalk. Se describen las clases de las que se compone el juego, las relaciones entre ellas y el desarrollo de la interfaz para lograr un juego dinámico y entretenido.

La Conquista consiste de un tablero con una cantidad de casillas determinada por el usuario, cada una con una temática. Los jugadores deben enfrentarse a duelos mediante preguntas, donde el ganador del duelo conquista las casillas que le pertenecen al contrincante. El juego finaliza cuando un jugador conquista todo el tablero.

Como extra, el juego incluye mecánicas innovadoras que fomentan la estrategia y premian el avance en la partida. Dichas mecánicas pueden ser obtenidas de dos maneras: el jugador consigue una racha de cinco casillas conquistadas o al ganar una ronda especial que consiste en un duelo entre los dos jugadores con mayor territorio.

2. Desarrollo

Modo de trabajo

La comunicación y organización del equipo se llevó a cabo mediante las plataformas WhatsApp, Discord, Google Drive y Trello. Las versiones más recientes del juego eran subidas a una carpeta en Drive asignando el número de versión correspondiente, y dando un aviso por WhatsApp. Las actividades a realizar eran anotadas en Trello. Mediante Discord, se realizaban llamadas periódicas para puesta en común de avances y necesidades.

Lógica del juego

A continuación se describen las clases que conforman el juego y cómo éstas se relacionan.

Tablero: La clase Tablero se encarga de gestionar toda la dinámica del juego desde el inicio hasta el final. Su función incluye inicializar el tablero con una cantidad de casillas definida por el número n ingresado por el usuario al comienzo, resultando en un tablero de $n*n$ casillas. Además, actualiza el tablero de forma continua, gestiona la captura de casillas tras cada duelo y determina el ganador del juego.

Para iniciar el juego se crea una instancia de la misma y se utiliza el método *iniciarJuego*. Al comenzar, si se desea abrir el Transcript durante la partida, se podrá visualizar un mensaje que indica que comenzó el juego, y que un jugador al azar fue seleccionado para comenzar. Durante toda la partida, en el Transcript se podrá visualizar el estado actual del juego, por ejemplo qué jugadores participarán del duelo, a qué casillas puede atacar un jugador, qué jugador ganó/perdió el duelo, etc

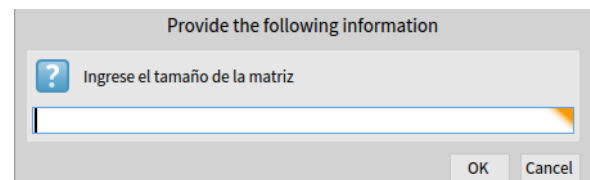


Fig. 1: Pantalla de bienvenida

En esta ventana se recibe el tamaño de la matriz para luego crearla de acuerdo al tamaño que ingresa el usuario

Jugador 1 geografía Posición: (1,1)	Jugador 2 deporte Posición: (2,1)	Jugador 3 geografía Posición: (3,1)	Jugador 4 historia Posición: (4,1)	Jugador 5 historia Posición: (5,1)	Jugador 6 deporte Posición: (6,1)	Jugador 7 ciencia Posición: (7,1)	Jugador 8 geografía Posición: (8,1)
Jugador 9 ciencia Posición: (1,2)	Jugador 10 arte Posición: (2,2)	Jugador 11 ciencia Posición: (3,2)	Jugador 12 deporte Posición: (4,2)	Jugador 13 geografía Posición: (5,2)	Jugador 14 arte Posición: (6,2)	Jugador 15 historia Posición: (7,2)	Jugador 16 deporte Posición: (8,2)
Jugador 17 historia Posición: (1,3)	Jugador 18 arte Posición: (2,3)	Jugador 19 historia Posición: (3,3)	Jugador 20 historia Posición: (4,3)	Jugador 21 geografía Posición: (5,3)	Jugador 22 deporte Posición: (6,3)	Jugador 23 arte Posición: (7,3)	Jugador 24 geografía Posición: (8,3)
Jugador 25 deporte Posición: (1,4)	Jugador 26 historia Posición: (2,4)	Jugador 27 geografía Posición: (3,4)	Jugador 28 geografía Posición: (4,4)	Jugador 29 historia Posición: (5,4)	Jugador 30 ciencia Posición: (6,4)	Jugador 31 deporte Posición: (7,4)	Jugador 32 deporte Posición: (8,4)
Jugador 33 ciencia Posición: (1,5)	Jugador 34 historia Posición: (2,5)	Jugador 35 arte Posición: (3,5)	Jugador 36 deporte Posición: (4,5)	Jugador 37 arte Posición: (5,5)	Jugador 38 geografía Posición: (6,5)	Jugador 39 deporte Posición: (7,5)	Jugador 40 ciencia Posición: (8,5)
Jugador 41 historia Posición: (1,6)	Jugador 42 arte Posición: (2,6)	Jugador 43 deporte Posición: (3,6)	Jugador 44 deporte Posición: (4,6)	Jugador 45 deporte Posición: (5,6)	Jugador 46 geografía Posición: (6,6)	Jugador 47 geografía Posición: (7,6)	Jugador 48 geografía Posición: (8,6)
Jugador 49 historia Posición: (1,7)	Jugador 50 geografía Posición: (2,7)	Jugador 51 arte Posición: (3,7)	Jugador 52 ciencia Posición: (4,7)	Jugador 53 arte Posición: (5,7)	Jugador 54 ciencia Posición: (6,7)	Jugador 55 geografía Posición: (7,7)	Jugador 56 deporte Posición: (8,7)
Jugador 57 ciencia Posición: (1,8)	Jugador 58 ciencia Posición: (2,8)	Jugador 59 arte Posición: (3,8)	Jugador 60 arte Posición: (4,8)	Jugador 61 historia Posición: (5,8)	Jugador 62 geografía Posición: (6,8)	Jugador 63 historia Posición: (7,8)	Jugador 64 arte Posición: (8,8)

Fig. 2: Tablero al iniciar la partida



Fig. 3: Presentación visual del logo del videojuego

Transcript			
Se creo el tablero de forma exitosa			
----INICIA EL JUEGO----			
----JUGADOR AL AZAR ECONTRADO----			
Cantidad de Jugadores: 64			
Jugador: Jugador 62 Selecciona a cual de las siguientes			
---COLUMNA---FILA---CATEGORIA-----JUGADOR---			
5	8	historia	Jugador 61
7	8	historia	Jugador 63
6	7	ciencia	Jugador 54

Fig. 4: Información actual de la partida en el Transcript

La partida es controlada mediante el mismo método (*iniciarJuego*), con un bucle que permite que el juego continúe hasta que solo quede un jugador. El mismo lleva la cuenta de la cantidad de duelos de la partida e

inicia la ejecución del Duelo Extra en caso de que el contador de duelos llegue a 10.

Casilla: Con esta clase se establece la categoría de cada casilla con el método *asignarCategoría*, y se determinan las casillas adyacentes a una casilla con el método *encontrarVecinos*.

Duelos: Mediante la clase *Duelos* se gestionan y ejecutan los enfrentamientos entre los jugadores, la mecánica principal a través del cual los participantes compiten por territorios. Esta clase administra el flujo de preguntas, los turnos, el temporizador y el resultado del enfrentamiento. El método *'iniciarDueloEntre.y:categoría:dueloExtra:'* inicializa el duelo entre dos jugadores, 'player1' y 'player2', creando una nueva instancia de la clase 'Pregunta' para determinar las preguntas.

El flujo del enfrentamiento inicia con la selección de una categoría de preguntas basada en la casilla atacada, se inicializa el temporizador para ambos jugadores y se establece el turno inicial el cual corresponde al atacante. Los jugadores alternan turnos para responder preguntas con el método *pasar turno*; pausa el temporizador de un jugador y reanuda el temporizador del jugador contrario hasta que éste responda correctamente y viceversa, si la respuesta es incorrecta; el temporizador actual continúa corriendo, si se elige *pasar*; se cambia la pregunta y se penaliza al jugador correspondiente restando tres segundos en su temporizador. El duelo continúa hasta que uno de los jugadores se quede sin tiempo.

En caso que *Tablero* determine que se ejecute un Duelo Extra o *Duelo de Titanes*, se invoca al método *dueloExtra*.

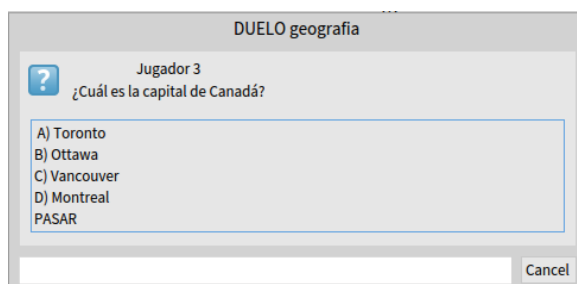


Fig. 5: Menú de selección de respuestas

Jugador: Esta clase manipula los *PowerUps*, lleva la cuenta de la *racha* del jugador, indica durante la partida el nombre del jugador actual y su lista de adversarios.

Pregunta: Esta clase selecciona las preguntas que se presentan durante los duelos, verifica que no se repitan, analiza las respuestas y determina si son

correctas o no. Las preguntas están basadas en cinco categorías: arte, ciencia, historia, deporte, geografía.

Timer: Controla el tiempo que el jugador dispone para responder las preguntas durante los duelos. Al momento de iniciar un duelo, esta clase asigna un temporizador a cada jugador y comunica cual llegó a cero primero. Esta clase también cuenta con una interfaz gráfica.



Fig. 6: Representación visual del temporizador

Bonos especiales

Existen dos tipos de bonos especiales:

Cañón: permite lanzar un ataque estratégico a cualquier oponente, incluso si no está en la periferia del jugador.

Escudo: Protege el territorio conquistado del jugador, evitando su conquista incluso tras ser derrotado en un duelo. Este escudo tiene un uso y se desactiva cuando el jugador pierde, por lo que en el próximo duelo estará vulnerable a perder su territorio nuevamente.

Cada diez duelos normales se presenta el '*Duelo de Titanes*', una ronda especial diseñada para recompensar a los jugadores que han destacado por su dominio del tablero. Este enfrentamiento involucra exclusivamente a los dos jugadores con mayor territorio conquistado hasta el momento, brindándoles una oportunidad para consolidar su posición en el juego. El Duelo de Titanes se rige por las mismas reglas y formato que los duelos normales: las preguntas, turnos y tiempos siguen la misma dinámica de 45 segundos por jugador. Sin embargo, esta ronda tiene una particularidad que la distingue: las preguntas son completamente aleatorias, seleccionadas de manera independiente de las temáticas de las casillas. Este elemento introduce un nivel adicional de imprevisibilidad, obligando a los jugadores a demostrar su conocimiento general y

adaptabilidad. El jugador ganador del Duelo de Titanes no solo se lleva el territorio de su oponente, sino que también adquiere la posibilidad de elegir uno de los dos bonos especiales: ‘Cañón’ o ‘Escudo’.

3. Conclusiones

Adaptarse a la documentación limitada disponible acerca del lenguaje SmallTalk y de Pharo representó uno de los mayores retos al inicio del desarrollo del juego.

El desarrollo grupal de forma remota, la complejidad del juego y por lo tanto la necesidad de manejar múltiples datos y estructuras para su buen funcionamiento representaron desafíos constantes pero supimos resolverlos mediante la comunicación efectiva, una adecuada distribución de tareas y el uso de herramientas colaborativas que facilitaron la organización y el seguimiento del progreso del proyecto.

Destacamos el aprendizaje adquirido acerca de la programación orientada a objetos, ya que este proyecto fue el primero que desarrollamos utilizando el paradigma de la POO, por lo tanto, el conocimiento adquirido es valioso y funciona como punto de partida para desarrollar proyectos similares o como recurso informativo para profundizar en este lenguaje y su paradigma. Mediante este trabajo, se nos permitió explorar a fondo el paradigma y sus beneficios.

Bibliografía

GNU Smalltalk Library Reference

https://www.gnu.org/software/smalltalk/manual-base/gst-base.html#index-linesDo_003a

Watt, D. (2004). Programming Languages Design Concepts. Scotland: John Wiley & Sons.

Joyanes Aguilar, L. (1996). Programación Orientada a Objetos. Madrid (España): McGraw-Hill.

Stéphane Ducasse and Gordana Rakic with Sebastijan Kaplar and Quentin Ducasse (March 26, 2022). *Pharo 9 by Example*

Anexo

UML con los atributos más relevantes de 'La Conquista' :

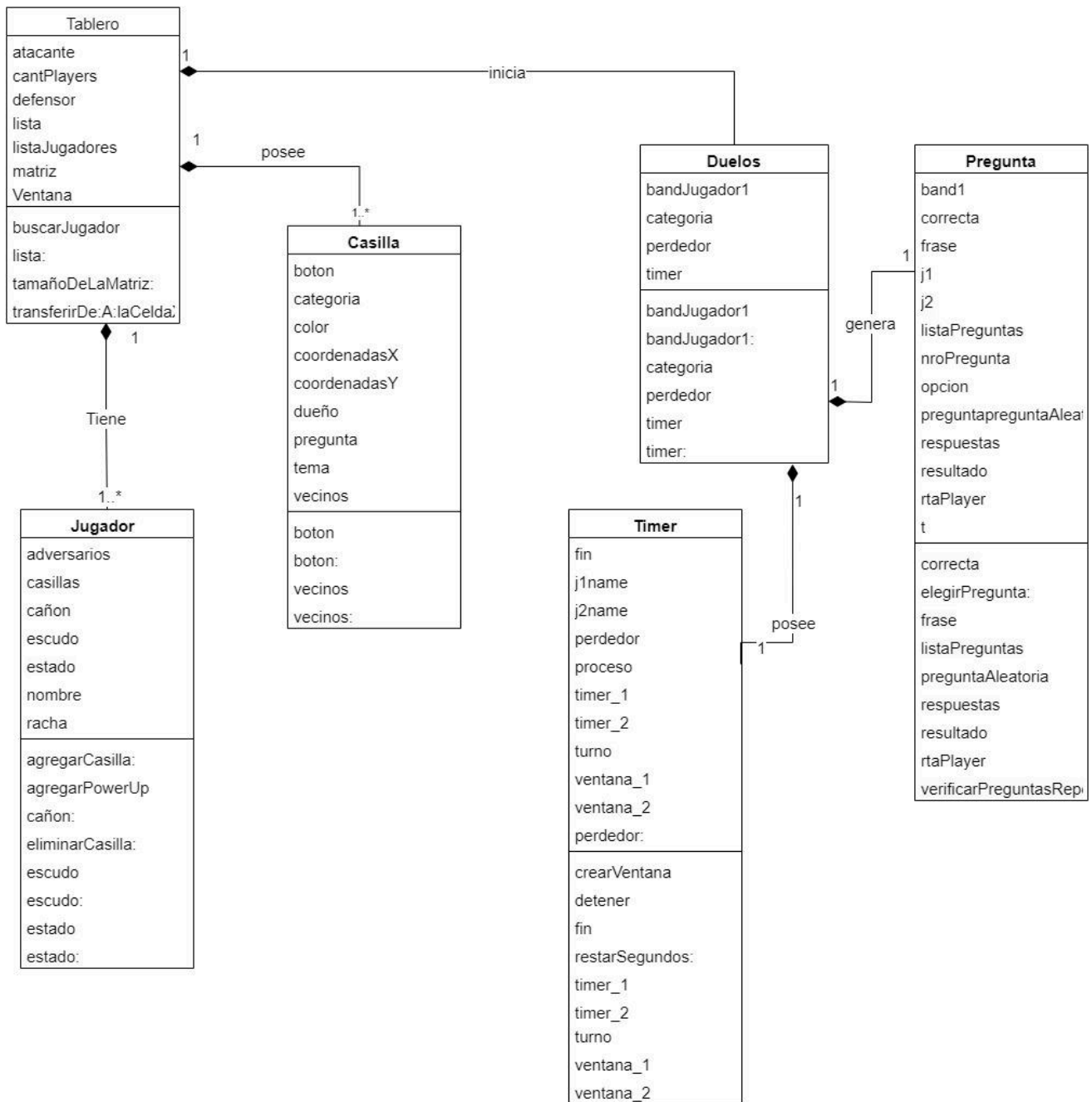


Fig. 7: Diagrama de Clases en UML