COMPUTADORES III

VIRTUAL CYBERTECH: SYSTEM DESIGN DESCRIPTION

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INDUSTRIALES

EQUIPO YELLOW:

Marta Dorado (Project Manager), Álvaro López, Antonio Díez, Carlos Sampedro, Francisco Suárez, Marie Destarac y Ricardo Espinoza.

Índice

1. Seguimiento de documento

Histórico de modificaciones

Versión Número	Fecha <<mm/dd/aa>>	Cambios	Autor	Revisor	Secciones afectadas
1	04/13/12	Creación del documento	Equipo Yellow	Guadalupe Sánchez	Todas
2	06/23/12	Diagramas de secuencia y de clases	Equipo Yellow	Guadalupe Sánchez	4 y 5

2. Introducción

El diseño del sistema tiene como objetivo dar una primera idea de cómo se concibe la aplicación, así como cuáles serán sus principales componentes y de qué forma interaccionan entre ellos.

Sus principales beneficios son evitar una programación caótica que lleve continuamente a rehacer una y otra vez el trabajo y dar a los arquitectos del sistema, programadores y clientes una idea de cómo funcionará el sistema. Esto, además, protege a los constructores y usuarios de la aplicación ante futuros cambios en la plantilla, ya que el sistema deberá comportarse como se especifica en este documento.

El primer paso es realizar el Modelo del Dominio que tiene como objetivo identificar las clases principales del sistema así como las relaciones entre ellas, dando así un primer concepto del sistema que lógicamente se irá complicando de acuerdo avanza este documento y se produce la realimentación usuario-cliente.

Posteriormente, incluiremos los Diagramas de Clases, donde daremos una visión de todos los elementos del sistema, incluidas clases que pudieran no haber sido identificadas durante el Modelo del Dominio o clases abstractas que los arquitectos o programadores utilicen como base para dar forma al sistema.

Por último, incluiremos unos Diagramas de Secuencia para que veamos casos de cómo interaccionan entre sí las distintas clases, para que los ejemplos sean lo más claro posible, realizaremos los Diagramas de Secuencia correspondientes a los Casos de Uso implementados en el documento correspondiente a los Requisitos de Usuario.

3. Modelo del dominio

El artefacto o documento más importante en el Análisis Orientado a Objetos (AOO) es el Modelo del Dominio. El Modelo del Dominio es una representación visual de las **clases conceptuales** del dominio.

Este diagrama del AOO se irá perfeccionando en el proceso iterativo que marca el UP (*Unified Process*), a medida que se descubran nuevas clases y relaciones entre ellas.

3.1. Identificación de clases

El primer paso en el AOO es la identificación de clases (conceptuales) que hay en el entorno del problema:

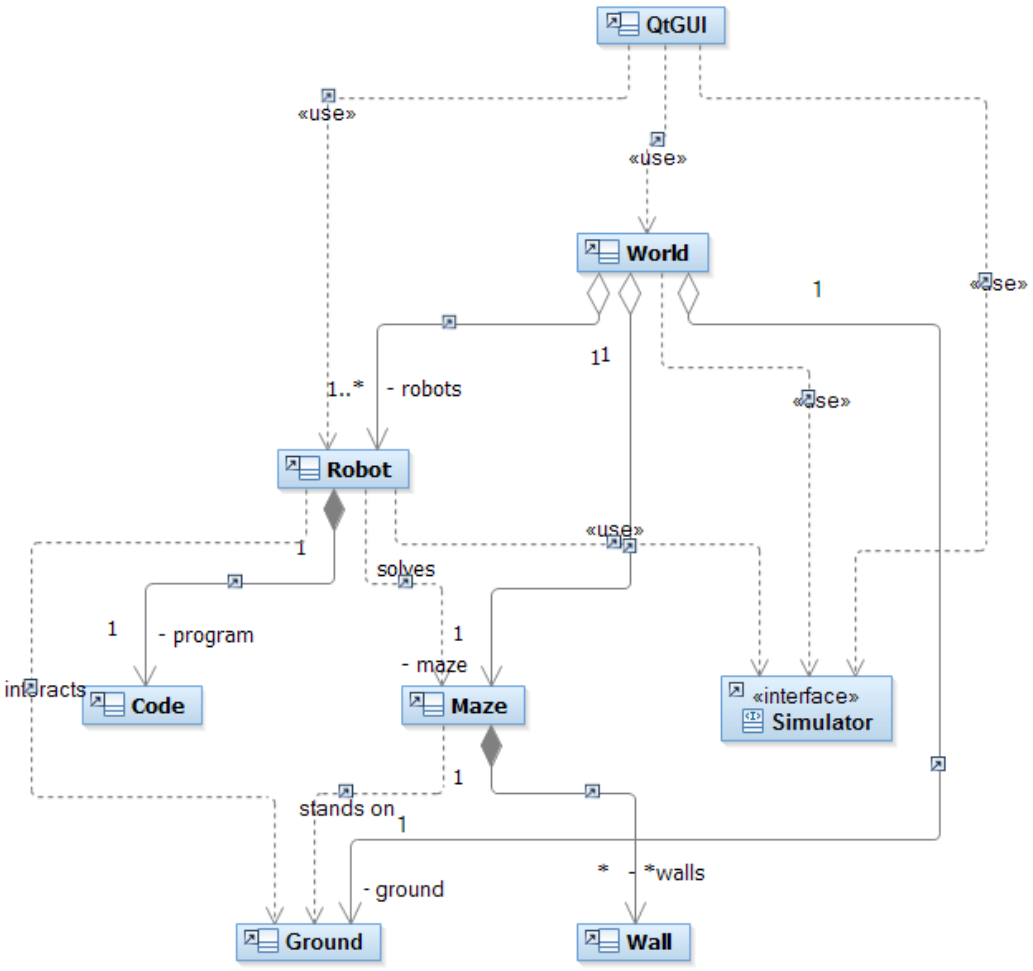
- ***QtGUI:*** interfaz grafica que interacciona con el usuario.
- ***Laberinto*****:** Conjunto de paredes entre las cuales se moverá el robot e interaccionará con ellas.
- ***Pared*****:** objeto que limita la pista, y un conjunto de ellas forman parte del laberinto.
- ***Robot*****:** vehículo que se moverá en el interior del laberinto y sobre el cual se escribirá el código.
- **Mundo:** objeto que contiene al robot, laberinto y suelo.
- **Código:** conjunto de instrucciones que controlan a robot.
- ***Suelo*****:** Suelo en el que se apoyará el laberinto, y sobre el cual se moverá el robot.
- **Simulador:** elemento donde se lleva a cabo la interacción entre el mundo y el código.

3.2. Identificación de relaciones entre clases.

El siguiente paso es identificar las relaciones entre las clases que hemos identificado anteriormente:

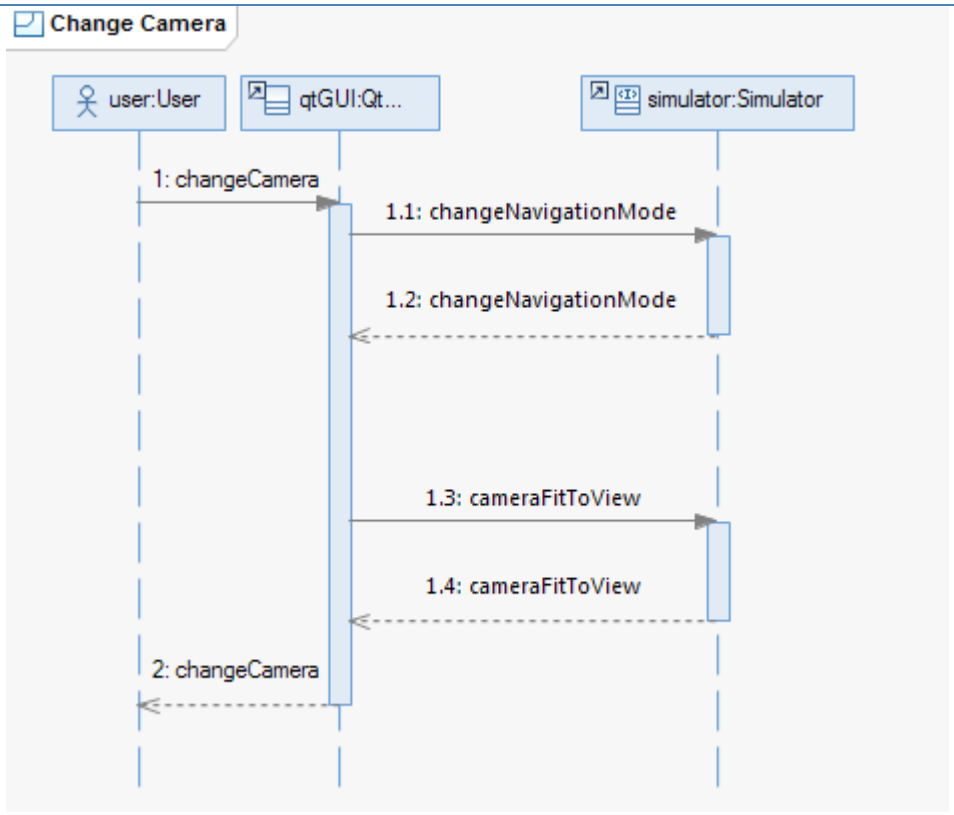
- La GUI usa mundo, robot y simulador.
- Robot contiene al código y usa a simulador, laberinto y suelo.
- Mundo está compuesto por robot, laberinto y suelo, y hace uso del simulador.
- El simulador es usado por el robot, el mundo y la GUI.
- El robot lleva contenido en él al código que lo controla.
- Laberinto está constituido por paredes que se encuentran sobre suelo, y es usado por robot.
- Suelo está contenido en el mundo y sobre él se encuentra el laberinto.

El *Modelo del Dominio* para las clases conceptuales de nuestro sistema podría quedar:



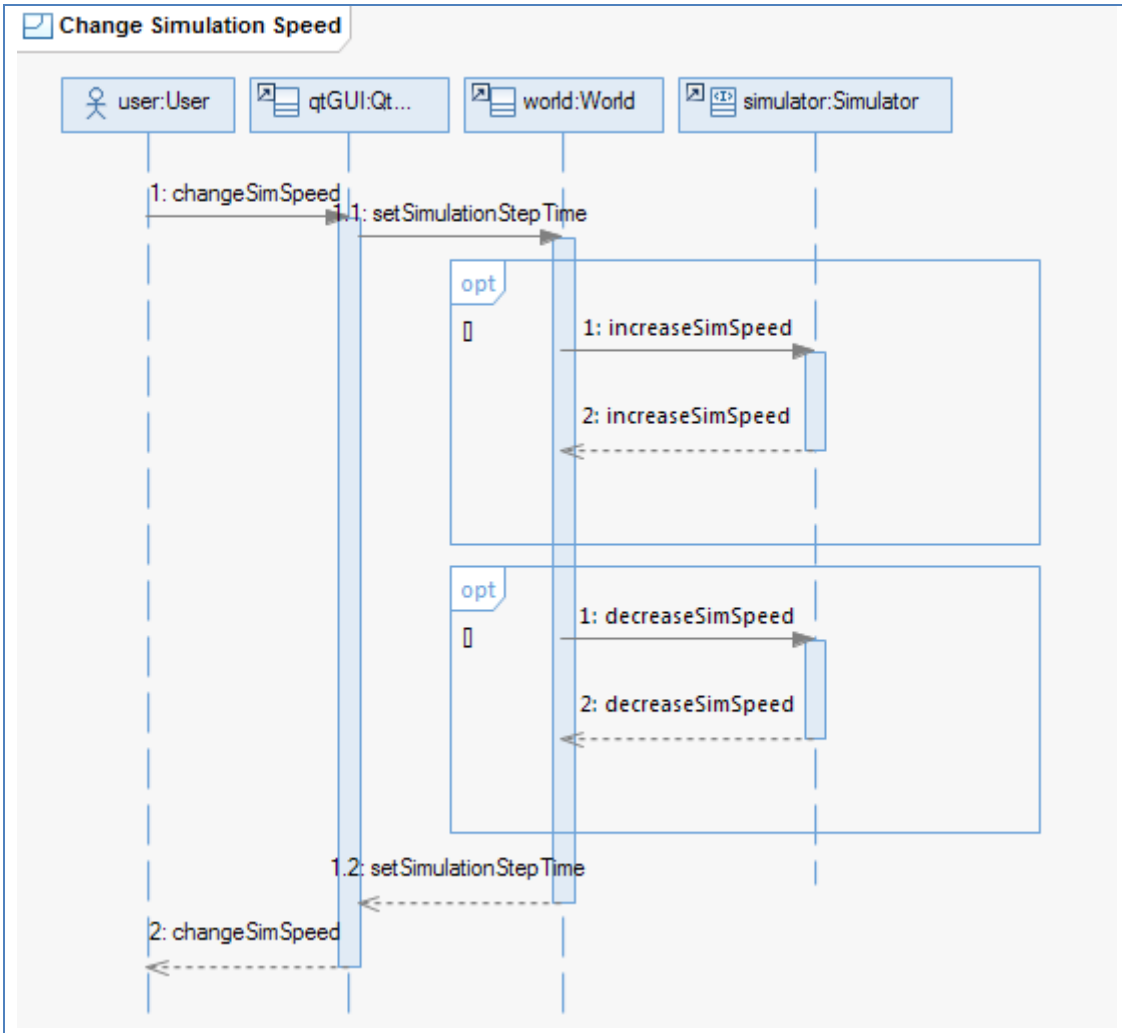
4. Diagramas de secuencia

4.1. Change Camera (Sequence Diagram)



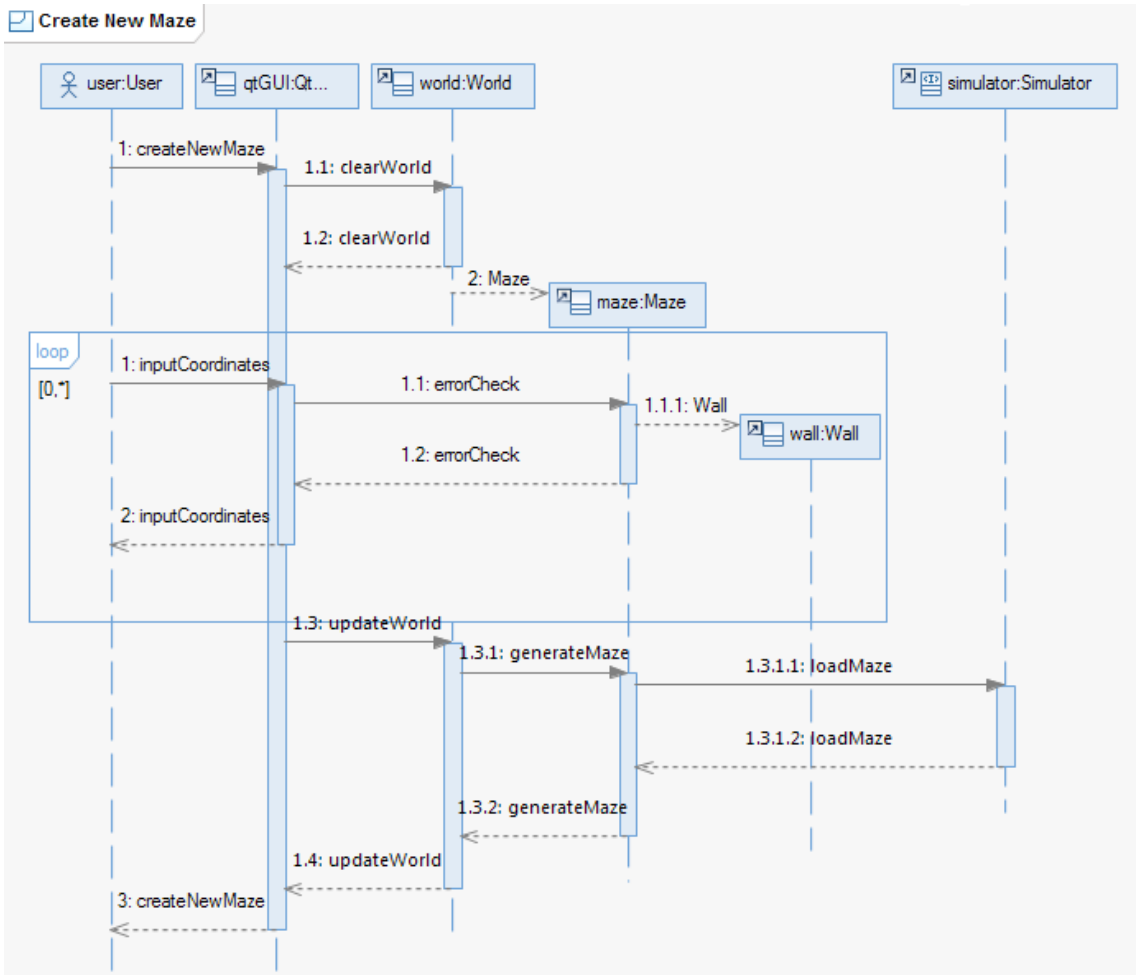
Descripción: Cuando el usuario desea cambiar la cámara actual de la simulación, la GUI se comunica con el simulador para que éste ajuste la vista seleccionada y devuelve en la ventana de simulación la imagen modificada, percibiendo el usuario el cambio realizado.

4.2. Change Simulation Speed (Sequence Diagram)



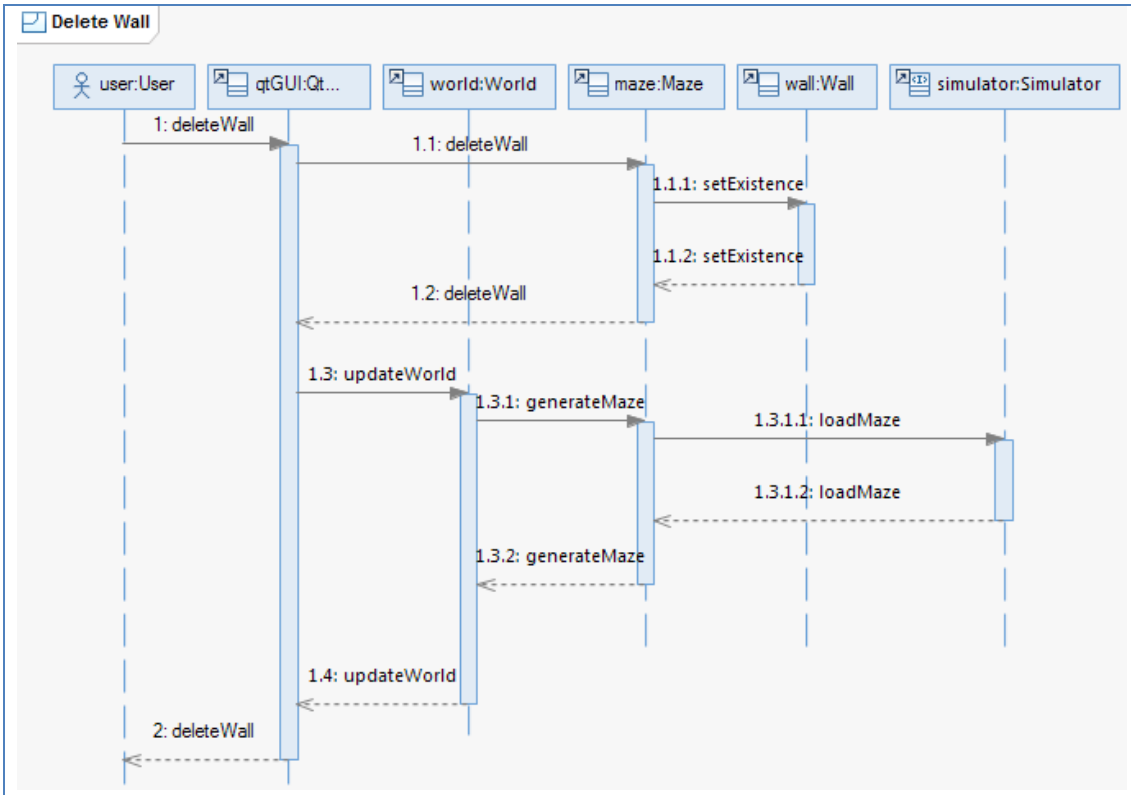
Descripción: Cuando el usuario desea cambiar la velocidad de la simulación, la GUI se comunica con Mundo para que éste fije la nueva velocidad y éste a su vez le solicita al simulador que realice el ajuste en tiempo real, de manera que el usuario perciba de manera inmediata el cambio.

4.3. Create New Maze (Sequence Diagram)



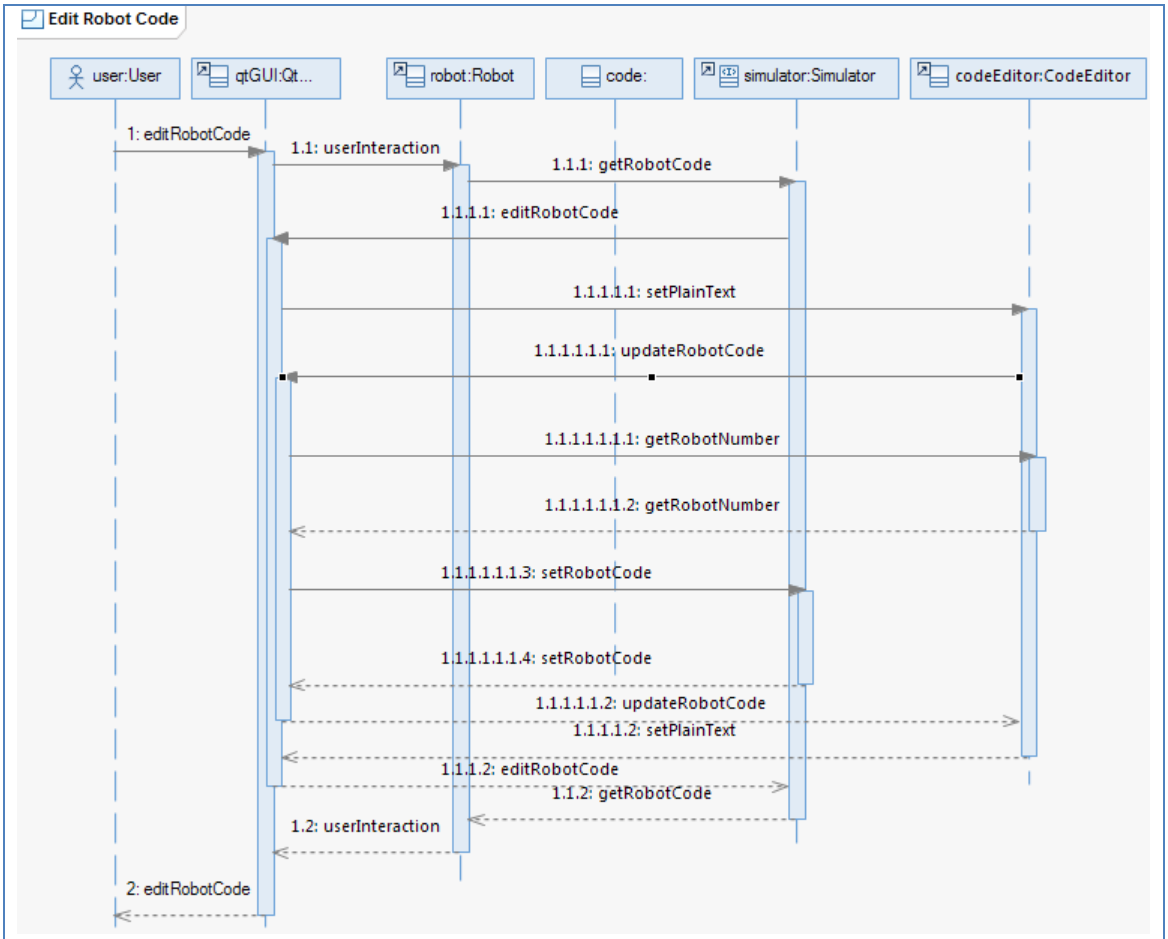
Descripción: Cuando el usuario desea crear un nuevo laberinto, la GUI se comunica con la clase Mundo para que éste cree el laberinto. Cuando el usuario coloca las paredes, la clase Laberinto realiza la verificación de que éstas sean correctas. Si no existen errores, y después de que el usuario presione el botón de actualizar, el simulador cargará las paredes que en ese momento se encuentren en el editor, para que sean visualizadas en el simulador.

4.4. Delete Wall (Sequence Diagram)



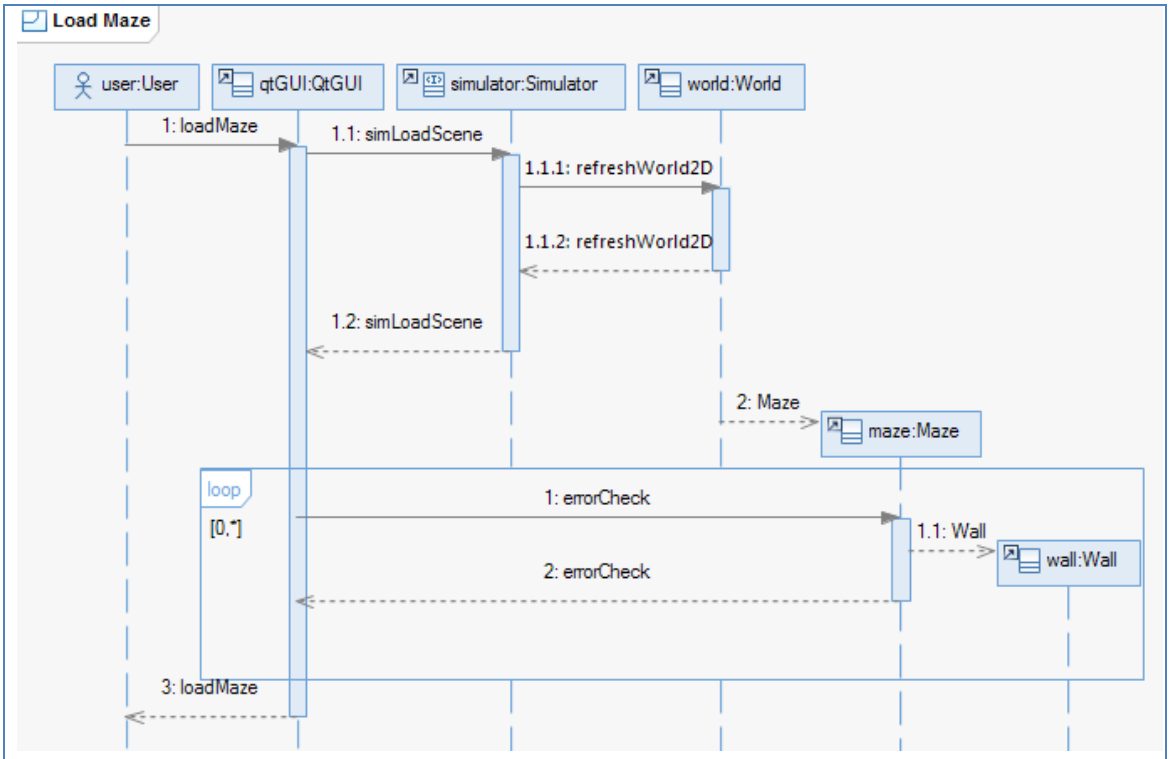
Descripción: El usuario selecciona una pared ya creada para borrarla, la GUI se comunica con la clase laberinto y éste con la clase pared, eliminándola. Después de que el usuario presione el botón de actualizar, el simulador carga la nueva escena sin la pared seleccionada.

4.5. Edit Robot Code (Sequence Diagram)



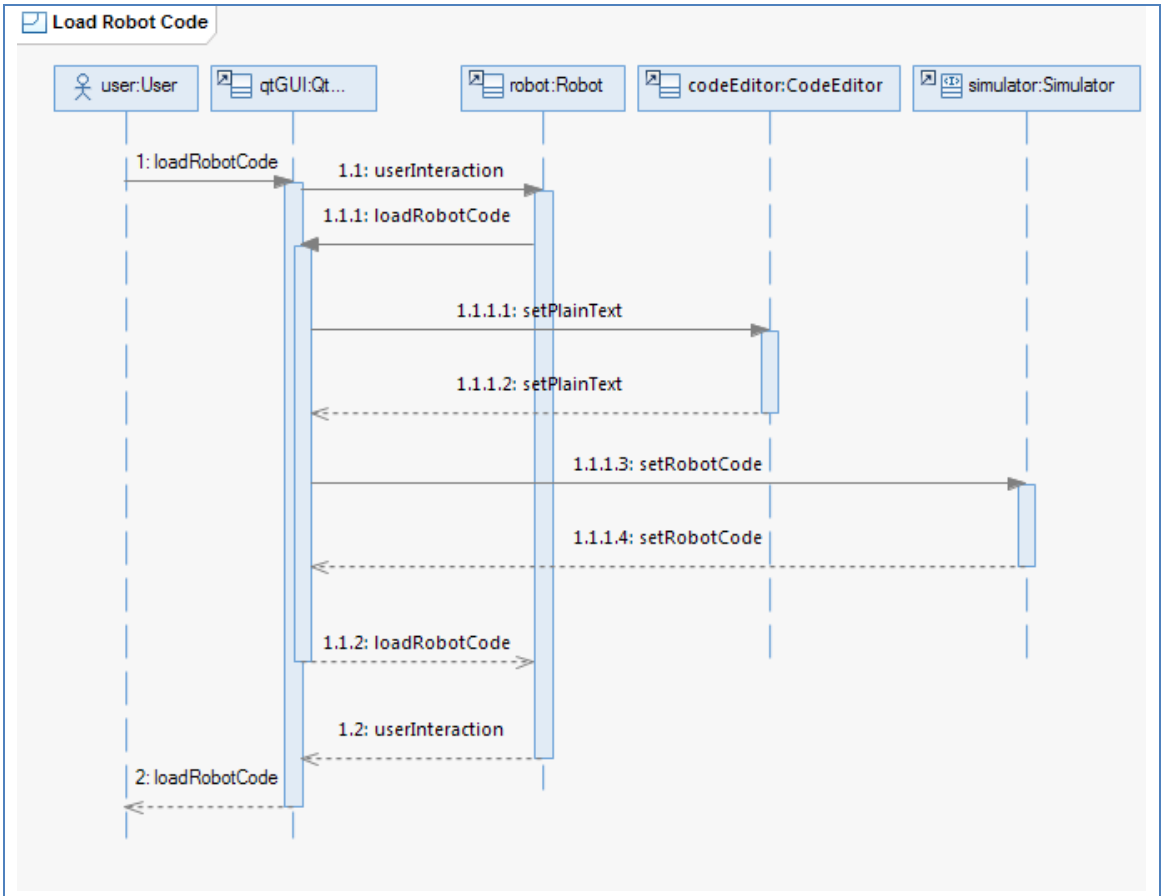
Descripción: El usuario selecciona editar el código del robot, la GUI se comunica con la clase Robot que a su vez le solicita al simulador que despliegue la ventana de código asociado al robot. Al editar el código éste es actualizado y cargado de nuevo.

4.6. Load Maze (Sequence Diagram)



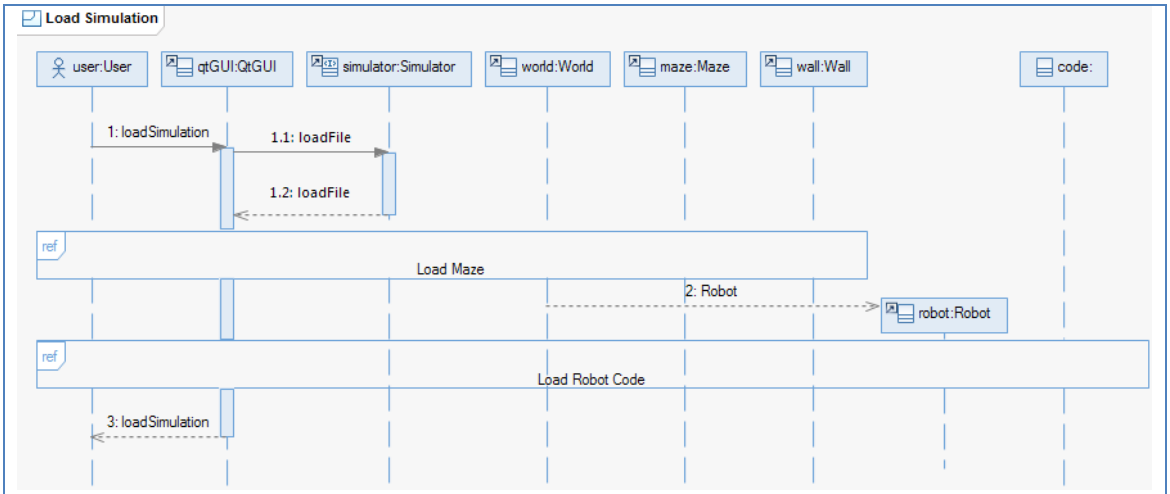
Descripción: El usuario selecciona cargar un laberinto, la GUI le solicita al simulador que cargue la escena e internamente se crea el laberinto seleccionado y se hace una detección de errores para cada pared. Si no los hay, se muestra el laberinto.

4.7. Load Robot Code (Sequence Diagram)



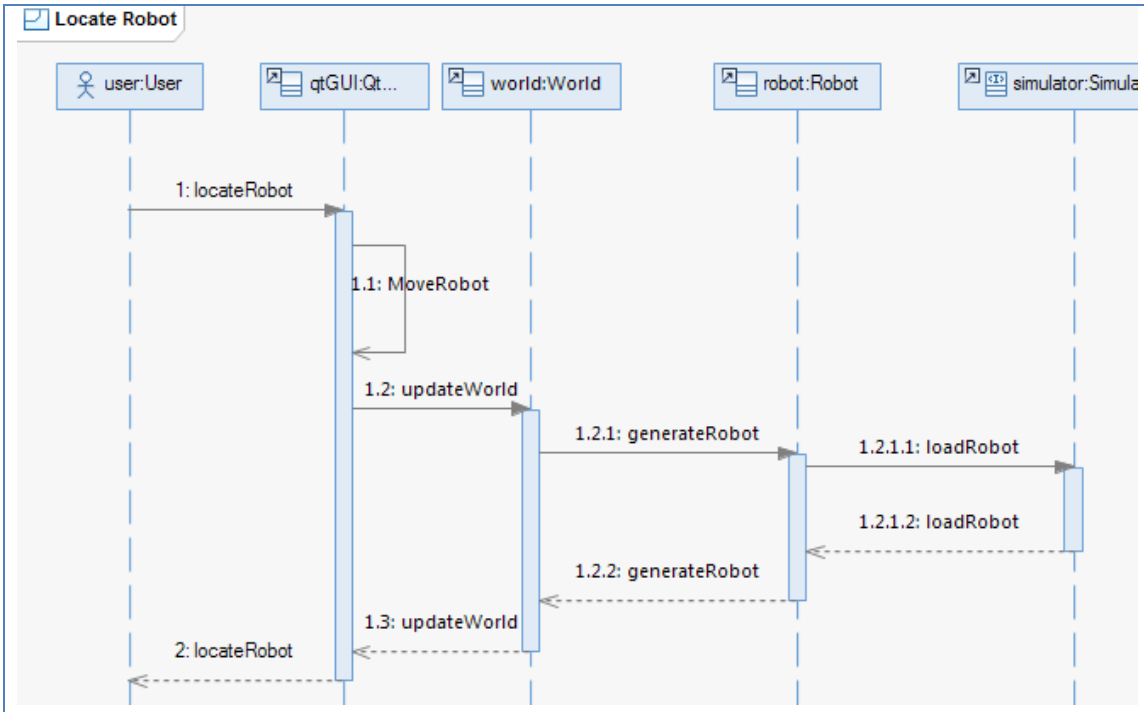
Descripción: El usuario selecciona cargar un código de robot, la GUI le solicita a la clase Robot que cree el código y muestra la ventana para el ingreso de código. El simulador muestra el código asociado con el robot.

4.8. Load Simulation (Sequence Diagram)



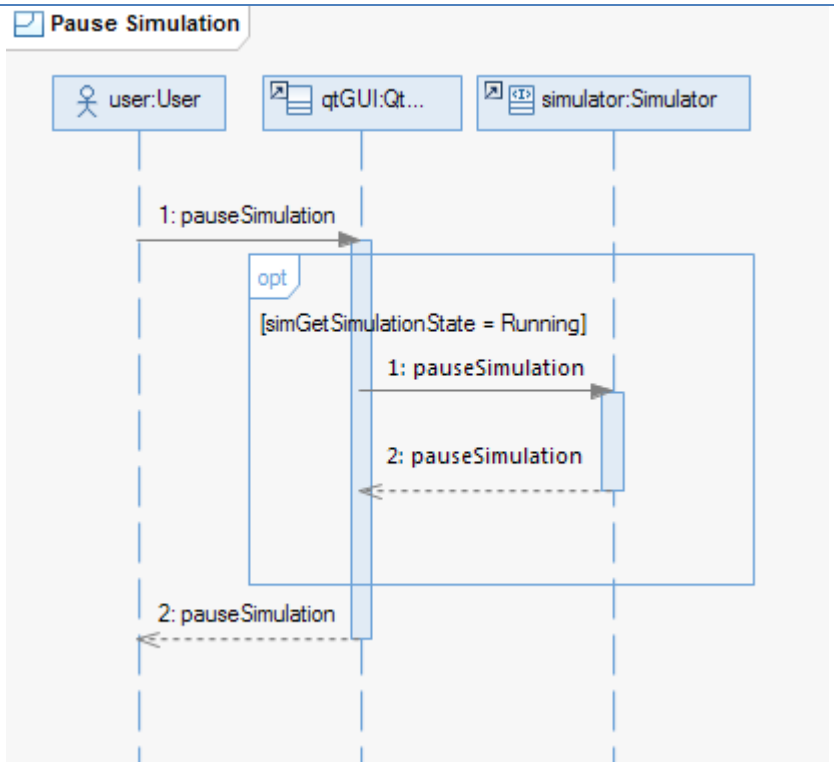
Descripción: El usuario selecciona cargar una simulación completa. La GUI le solicita al simulador que cargue la escena. En éste diagrama se hace referencia a Load Maze y Load Robot Code, descritos anteriormente.

4.9. Locate Robot (Sequence Diagram)



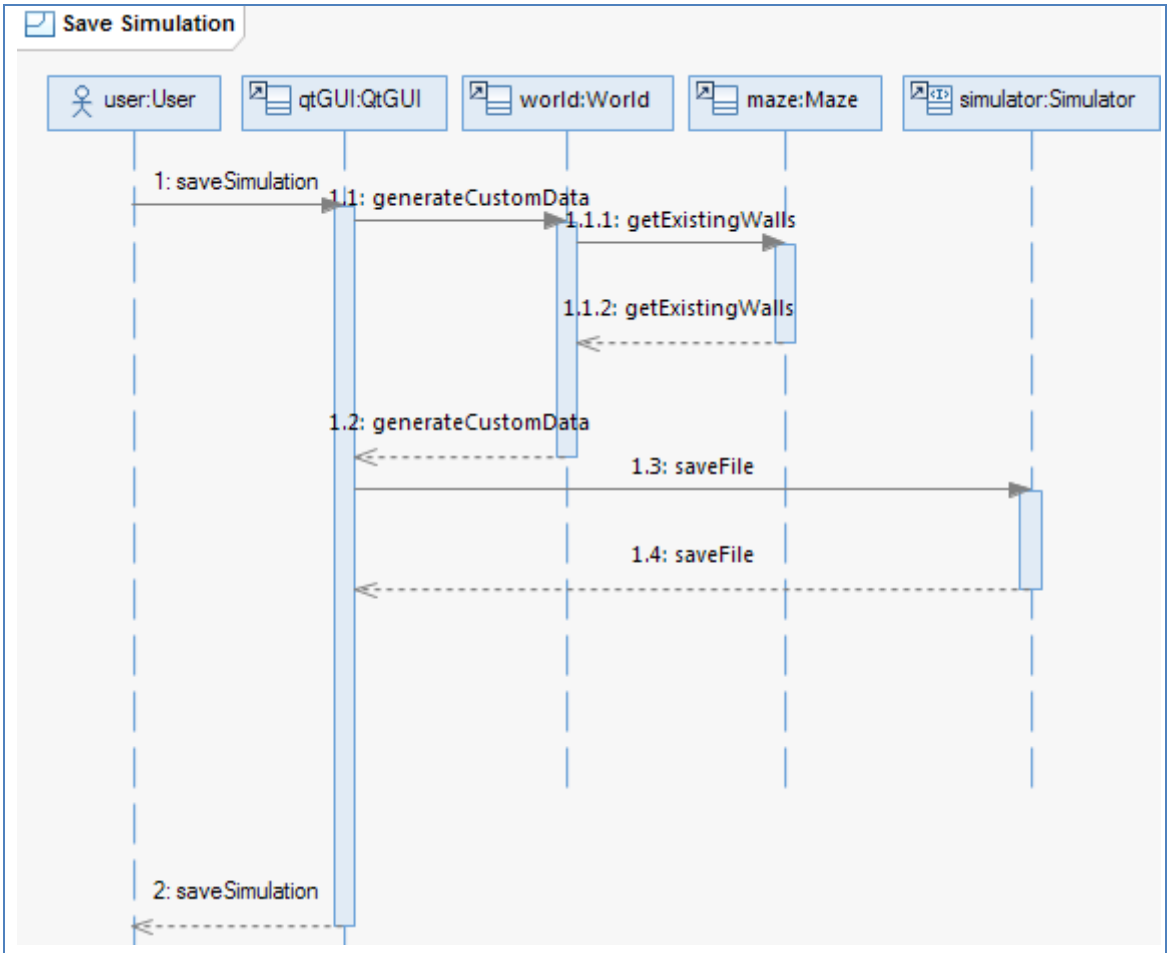
Descripción: El usuario ubica al robot en una determinada posición dentro de la ventana de edición de laberinto, creándose así un objeto de la clase robot. El simulador muestra éste una vez que el usuario haya presionado el botón de actualizar.

4.10. Pause Simulation (Sequence Diagram)



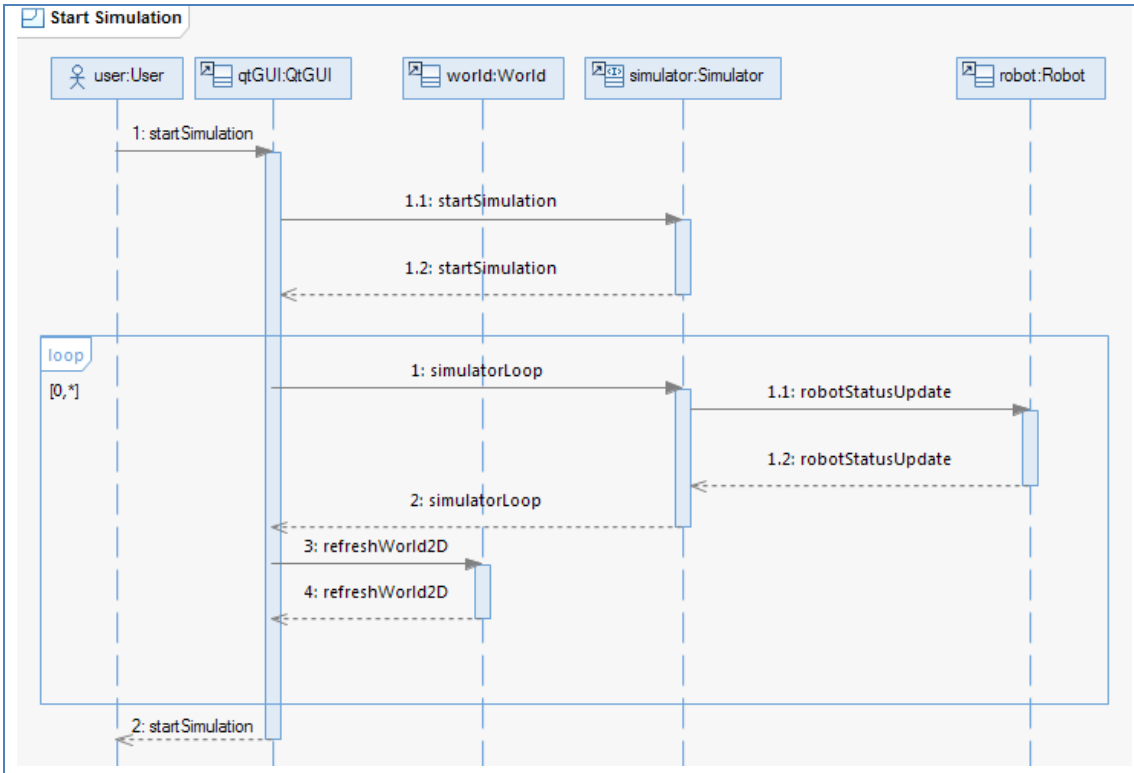
Descripción: El usuario solicita pausar la simulación y si ésta se encuentra activa, el simulador la detiene hasta que el usuario la reanude.

4.11. Save Simulation (Sequence Diagram)



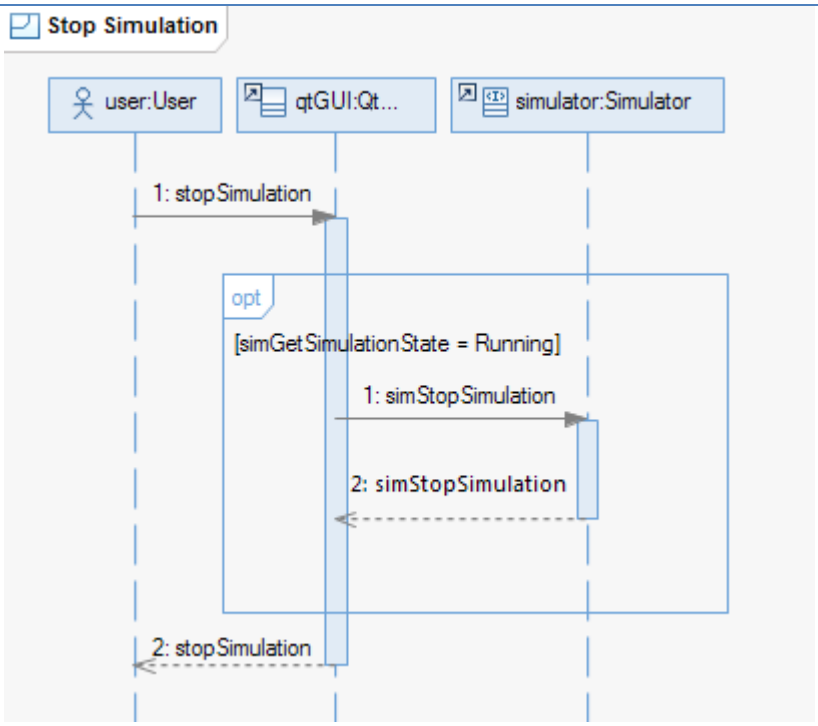
Descripción: El usuario solicita guardar una simulación. La clase mundo recoge las paredes que conforman el laberinto, así como la ubicación del robot. Se crea un archivo que es guardado por el simulador.

4.12. Start Simulation (Sequence Diagram)



Descripción: El usuario solicita iniciar una simulación y para ello el simulador recoge los datos necesarios de la clase Robot, como el código, la posición. Además, en la ventana de edición de laberinto se muestra el movimiento del robot en tiempo real.

4.13. Stop Simulation (Sequence Diagram)



Descripción: El usuario solicita detener la simulación y si ésta se encuentra activa, el simulador la detiene sin posibilidad de reanudarla.

5. Documentación de clases

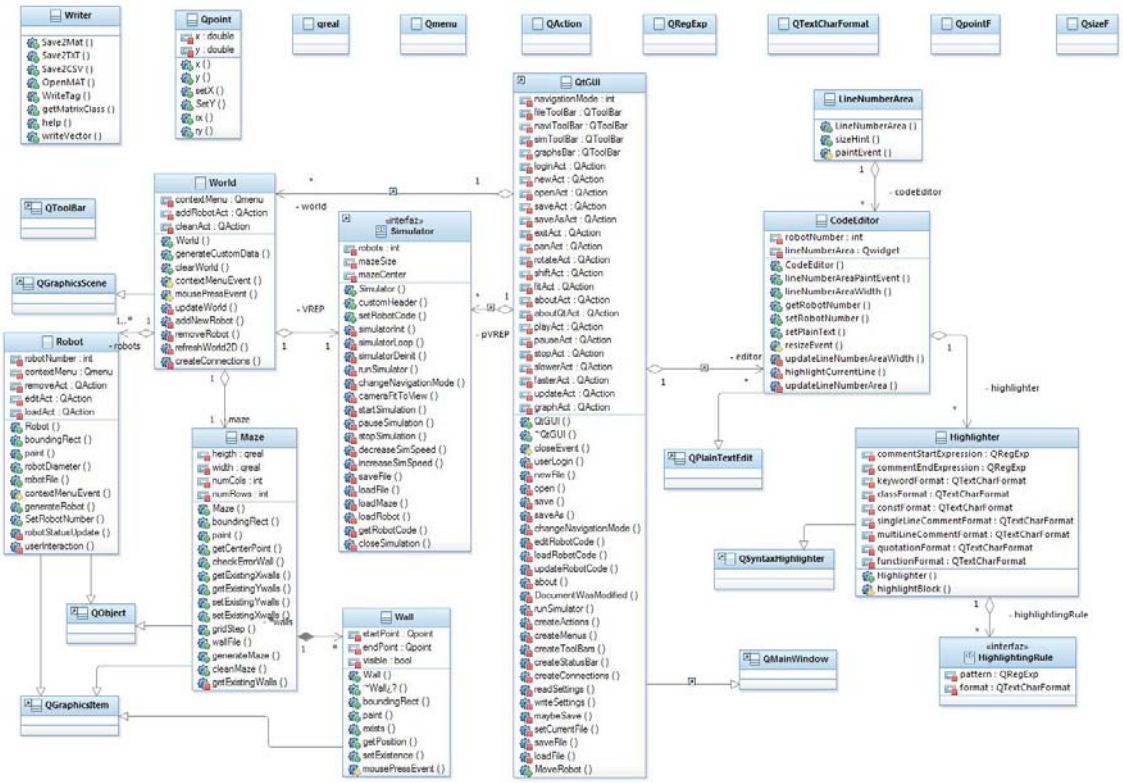


Figura 1. Diagrama de Clases del Sistema

En la figura 1 se muestra el diagrama de clases general del sistema. La documentación detallada se puede encontrar en el sitio oficial del simulador:

http://www.romin.upm.es/fsuarez/YellowSimulator/index.html