

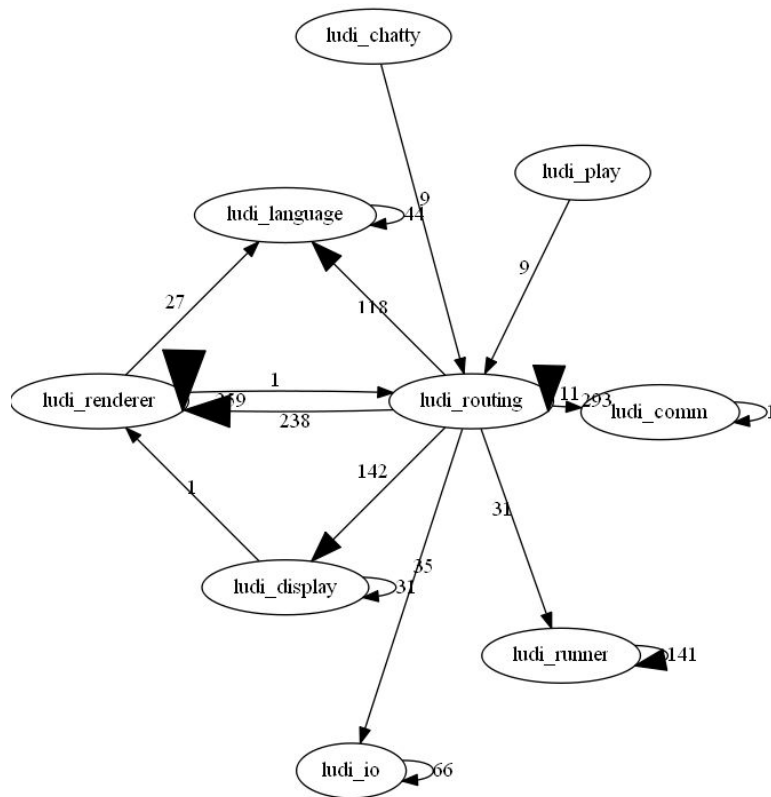
Interioridades javascript de LUDI

Si sólo te interesa ver por encima cómo se hacen los juegos en LUDI, consulta la [Guía de programación de juegos LUDI.txt]. Todo lo que sigue quizás no te interese tanto.

Por ahora sólo publicamos lo referente al lado del cliente, que es el que ejecuta los juegos.

El lado servidor está implementado en node.js (javascript) y se encarga del intercambio de mensajes entre clientes. Sólo consta de un fichero en javascript.

En \client\src están la docena de ficheros javascript de los que consta el cliente de LUDI. La estructura general es la siguiente (por simplicidad no se muestran módulos adicionales secundarios):



Explicación:

Fuente de entrada de acciones a ejecutar: ludi_play (input del jugador) y ludi_chatty (mensajes desde el servidor).

Gestor de las acciones: ludi_routing. Se encarga de:

- Cargar los ficheros de idiomas básicos kernel (usando ludi_io).
- Pasar información que se desea mostrar para que sea convertida a identificadores de mensajes independientes del idioma (usando ludi_renderer)

- Solicitar traducciones de identificadores prerrenderizados al idioma final (usando ludi_language).
- Representar los datos renderizados finales en pantalla (usando ludi_display)
- Ejecución de las acciones del juego sobre los datos internos del juego (usando ludi_runner).
- Enviar mensajes al servidor (ludi_comm).

El procesador de acciones del juego (ludi_runner), se encarga de llamar a ludi_lib_reactions (reacciones por defecto) y game_reactions, que contiene las reacciones del juego en concreto. El módulo game_reactions se carga dinámicamente cuando se selecciona un juego. Ambos módulos de reacciones, han uso del módulo ludi_lib (definición del juego de instrucciones LUDI que se pueden usar).

Posibilidad futura (vaporware): juegos jugables en el servidor

La idea sería convertir algunos de los módulos clientes, para que en vez de ser usados como ahora en el navegador web, en módulos node.js.

Esto permitiría que los juegos se ejecutaran en el servidor, permitiendo algunas novedades:

- el código de los juegos en el servidor quedaría oculto al cliente, pero sólo podría jugar online;
- crear juegos con características MUD: los jugadores competirían por ítems definidos a nivel de servidor.

Los módulos node.js son transformables en módulos de navegador usando el programa "browserify". Esto permite desarrollar de una sola vez para el servidor y el cliente.

Por ahora, se mantiene una estructura de módulos que facilite esa evolución futura, pero no es una prioridad.