JESUÏTES El Clot
Escola del Clot

M06. Desenvolupament Web en Entorn Client

# UF4. Comunicació asíncrona client-servidor

4.1 Node.JS
4.2 AJAX
4.3 Funcions asíncrones i promeses
4.4 Fetch API
4.5 MongoDB

(RA1)

curs 2020-21

Sergi Grau

Descripció.

Es vol desenvolupar una aplicació web que permeti jugar a un joc de torns multijugador: domino.

Podeu trobar les regles del joc a https://ca.wikipedia.org/wiki/D%C3%B2mino

Per a fer-ho s'utilitzarà Node.JS. Les comunicacions seran asíncrones utilitzant la tecnologia AJAX, API Fetch i us de promeses i funcions asíncrones.

Per a fer més senzill el joc, suposarem que només poden jugar 2 jugadors. Les dades que s'envien són la fitxa que mou un jugador i la posició (esquerra o dreta), i quina fitxa se li ha assignat. En començar una partida cal repartir les fitxes per a cada jugador. Cal enviar la informació de les fitxes que es van agafant.

Es tracta d'un joc multijugador,  on NO es pot jugar contra la màquina.

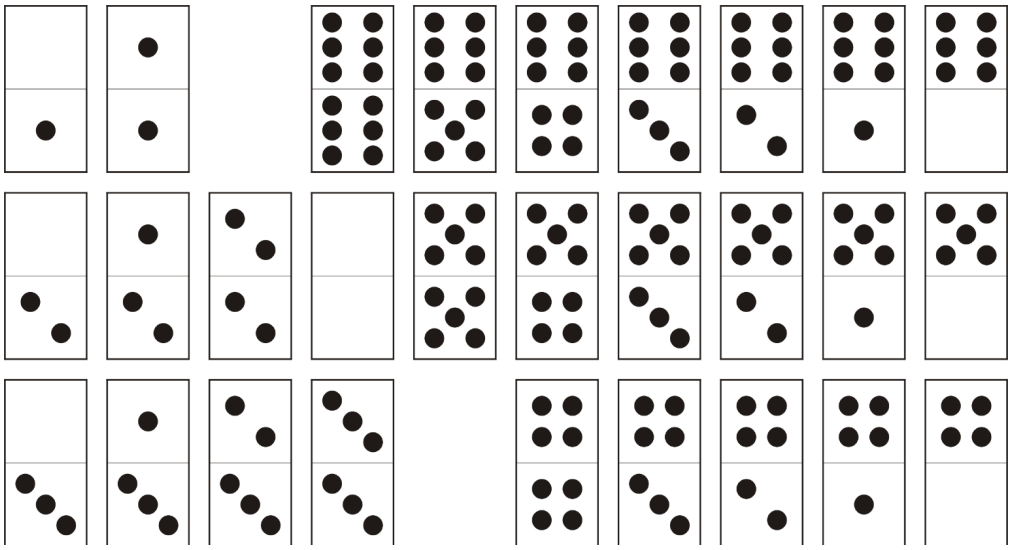
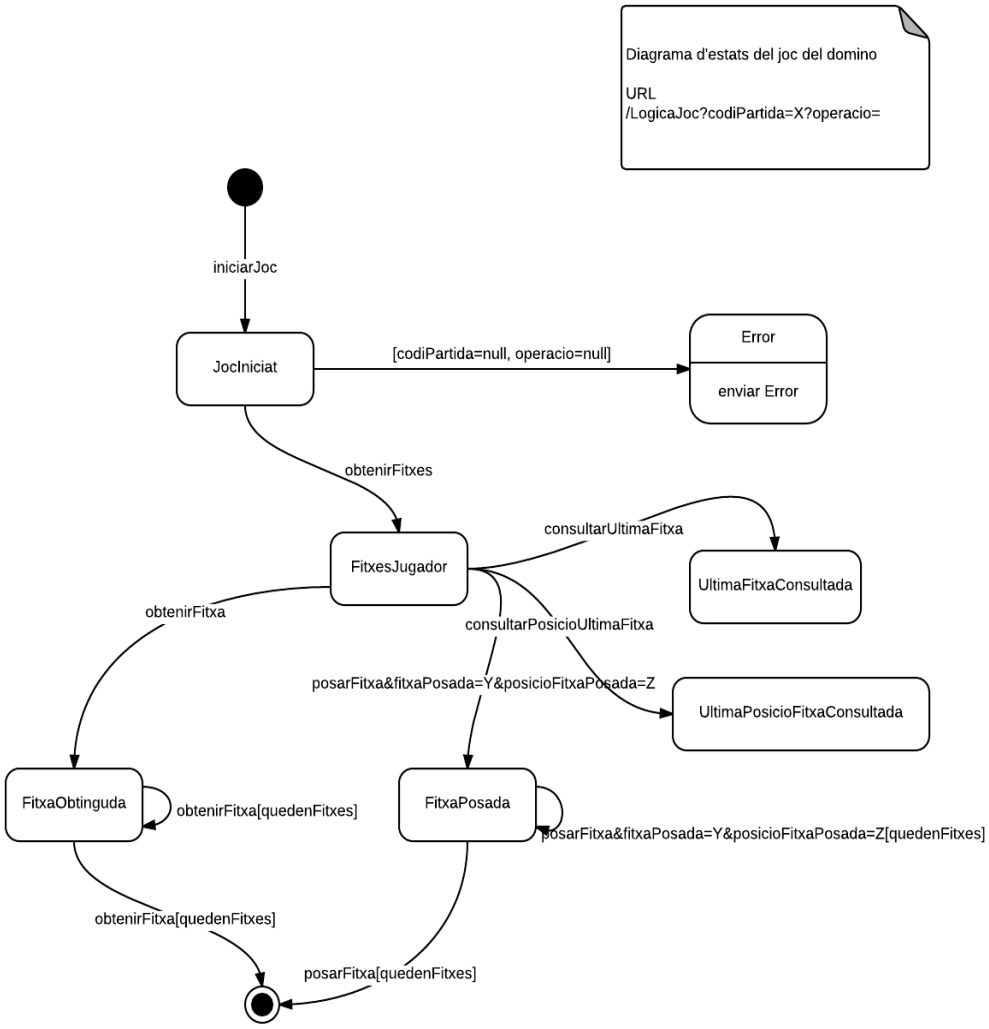


Diagrama d'estats.



Rúbrica d'avaluació

La rúbrica d'avaluació del projecte és la següent:

| Criteris | 4 | 3 | 2 | 1 |
|---|---|---|---|---|
| Sistema d'autenticació (10%) | El sistema de login està implementat, i poden jugar-se diverses partides a la vegada. | El sistema de login està implementat però només poden jugar 2 jugadors. Hi ha logout. | El sistema de login està implementat però només poden jugar 2 jugadors. No hi ha logout. | El sistema de login no està correctamente implementat |
| Lògica del joc en NodeJS( 40%) | La lògica és correcta. És veu l'efecte d'encertar de guanyar un moviment, i de posar les fitxes. S'utilitza encaminament. | La lògica és correcta. És veu l'efecte d'encertar de guanyar un moviment, i de posar les fitxes. No s'utilitza encaminament. | La lògica és correcta però molt bàsica. No s'utilitza encaminament. | La lògica és incorrecta i no permet jugar al joc de manera completa. |
| POO (15%) | S'ha creat la classe Partida i Jugador i s'han implementat mètodes per a les diverses operacions dels moviments. Es fan les verificacions necessàries i associades a la lògica del joc. | S'ha creat la classe Partida i Jugador i s'han implementat mètodes per a les diverses operacions dels moviments, però sense verificar la lògica del koc. | S'ha creat la classe Partida i Jugador. Però només per a desar dades. | Només s'ha creat una classe. |
| AJAX (12%) | S'ha utilitzat XHR nivell 2 amb JSON. Les imatges dels avatars de cada jugador es transfereixen per XHR2. | S'ha utilitzat XHR nivell 2. S'utilitzen objectes JSON. | S'ha utilitzat XHR nivell 1. No s'utilitzen objectes JSON. | La implementació de AJAX és molt bàsica e incompleta. |
| API Fetch, Promeses i funcions asíncrones (13 %) | L'aplicació fa us de XHR en el procés d'autenticació i de creació de la partida, | L'aplicació fa us de XHR en el procés d'autenticació i | Es fan servir promeses i l'API Fetch de manera parcial. | Només es fan servir promeses. |

| | | | | |
|---|---|---|---|---|
| | i en la resta de la lògica de l'aplicació utilitza, Fetch i promeses. A més s'utilitzen funcions asíncrones en bona part del codi. L'ús de promeses i funcions asíncrones es fa servir també en el backend. | de creació de la partida, i en la resta de la lògica de l'aplicació utilitza, Fetch i promeses. | | |
| MongoDB (10%) | S'ha utilitzat persistència de dades amb MongoDB, inclou el desament de les puntuacions, consulta de les mateixes i enregistraments dels jugadors. El tractament es fa mitjançant un patró DAO. | S'ha utilitzat persistència de dades amb MongoDB, inclou el desament de les puntuacions, consulta de les mateixes i enregistraments dels jugadors. | S'ha utilitzat persistència de dades amb MongoDB, però és molt bàsica. | No s'ha utilitzat persistència de dades. Només es llegeixen dades des del MongoDB. |

Funcionalitats a implementar.

- El Backend de l'aplicació ha d'estar desenvolupat amb Node.JS
- El Frontend de l'aplicació cal desenvolupar-ho amb JS (és opcional utilitzar ES6).
- Les dades del Frontend s'envien mitjançant AJAX i objectes JSON
- Es fa ús de promeses, funcions asíncrones i API Fetch.
- cal informació en el backend a la base de dades NoSQL MongoDB.

Característiques del disseny.

- Cal utilitzar el paradigma de POO.
- El disseny d'interfície web és lliure.

- Les comunicacions amb XHR2 i Fetch utilitzen JSON.
- Només es pot utilitzar Node.JS per a resoldre la part del backend.

Cal lliurar-la abans del dia especificat al Classroom de l'assignatura.
S'ha d'entregar en fitxer zip o tar, amb el format
COGNOM_NOM_PRACTICA_ENTORN_CLIENT_UF4.zip

- Aquest fitxer contindrà un fitxer en format PDF descrivint les característiques de la implementació realitzada.
- Aquest fitxer zip contindrà tot el codi font, biblioteques emprades i el contingut estàtic de la aplicació web.