Bartomeu Garcías Vidal

PROJECTE PROGRAMACIÓ

1º DAW 24-25

**Documentació Tècnica**

**1. Anàlisi de Requisits**

**Objectius:**

* Simular el Sistema Solar de manera interactiva.
* Permetre a l'usuari seleccionar quins planetes vol simular.
* Gestionar planetes (afegir, eliminar) mitjançant una interfície web d'administració.
* Garantir la persistència de dades mitjançant una base de dades MySQL.

**Funcionalitats:**

* Visualització del sistema solar en moviment.
* Controls de zoom, moviment de càmera i velocitat temporal.
* Traçats de les òrbites.
* Gestió d'objectes astronòmics des de l'administració (afegir, eliminar, reinicialitzar).
* Actualització dinàmica de la simulació.

**2. Disseny del Sistema**

**Arquitectura:**

* **Frontend**: HTML + CSS + JavaScript.
* **Backend**: Java EE (Servlets) amb connexió MySQL.
* **Base de dades**: MySQL, normalitzada.

La totalitat de la lògica del programa ha esta dissenyada emprant java, als arxius de javascript només s’ha introduït el codi necessari per a poder visualitzar les simulacions calculades en el backend.

**Classes principals:**

* SimulationServlet: Comunica al forntend en temps real les posicions i velocitats de cada un dels planetes calculades al backend així com els diversos ajustes del viewport.
* StartSimulationServlet: Comunica el frontend amb el backend per tal de poder iniciar la simulació nomes amb una selecció concreta de de cossos.
* ResetServlet: Comunica el frontend amb el backend per tal de poder reiniciar la simulació amb els valors inicials correctes per poder realitzar els càlculs de forma correcta.
* PlanetAdminServlet: Servlet que comunica la base de dades amb el java per poder intercanviar informació entre ells.
* ViewportController: Codi java que s’encarrega de convertir les coordenades calculades a coordenades que es puguin representar a la pantalla.
* PhysicsEngine: Dins aquesta classe es duu a terme la majoria dels càlculs necessaris dins la nostra simulació.
* SimulationState: Gestiona tant l’estat actual de la simulació com les funcions que permeten recarregar i reiniciar la nostra simulació.
* CelestialBody: Aquesta classe defineix el constructor de cada un dels cossos amb els que estarem treballant durant la simulació.
* DbManager: Gestiona directament la comunicació entre el nostre la base de dades i la resta del nostre projecte.

**Seqüència resumida:**

1. L'usuari selecciona accions al frontend.
2. El JavaScript fa peticions HTTP a servlets.
3. Els servlets processen les dades i accedeixen a MySQL.
4. La resposta JSON actualitza el frontend.

**3. Model de Dades**

**Taula celestial\_bodies:**

| **Camp** | **Tipus** | **Descripció** |
| --- | --- | --- |
| id | INT AUTO\_INCREMENT | Clau primària |
| name | VARCHAR(100) | Nom del cos celeste |
| mass | DOUBLE | Massa |
| semi\_major\_axis | DOUBLE | Semi-eix major de l'òrbita |
| eccentricity | DOUBLE | Excentricitat de l'òrbita |
| radius | INT | Radi (representació gràfica) |
| color | VARCHAR(7) | Color en format HEX |
| x | DOUBLE | Posició X |
| y | DOUBLE | Posició Y |
| vx | DOUBLE | Velocitat X |
| vy | DOUBLE | Velocitat Y |

**4. Guia d'Instal·lació i Ús**

**Requisits previs:**

* Eclipse IDE instal·lat.
* Tomcat 9 configurat com a servidor.
* XAMPP o MySQL Server actiu amb base de dades solar\_system.
* Llibreries gson.jar i mysql-connector-java.jar dins /WebContent/WEB-INF/lib/.

**Instal·lació:**

1. Importa el projecte a Eclipse com a Dynamic Web Project.
2. Configura Tomcat al projecte.
3. Importa el fitxer solar\_system.sql a MySQL.
4. Assegura't que el DB\_USER i DB\_PASSWORD són correctes a DbManager.java.

**Ús:**

1. Inicia el servidor Tomcat.
2. Accedeix via navegador a http://localhost:8080/SolarSystemSimulation/.
3. Usa els botons per:
   * Moure la càmera
   * Canviar la velocitat
   * Afegir o eliminar planetes
   * Iniciar noves simulacions seleccionant planetes
   * Reiniciar la base de dades

**5. Proves Realitzades**

**Proves unitàries:**

* Test de la creació de cossos celestes (CelestialBody).
* Test del càlcul de velocitats orbitals (PhysicsEngine).

**Proves d'integració:**

* Comunicació correcta entre frontend i servlets.
* Inserció i eliminació de planetes a MySQL.

**Proves funcionals:**

* Visualització correcta de la simulació a diferents nivells de zoom.
* Traçats orbitals ajustats a l'excentricitat.
* Correcció del reset de traçats.
* Funcionament de l'administració de planetes.