**WaterGames**

Metode de Dezvoltare Software

Radu Cosmin-Andrei

Udroiu Laura-Ioana

Georgescu Andreea-Maria

**IDEEA PROIECTULUI**

Proiectul nostru reprezinta un site ce contine doua jocuri si un mini quiz, cu si despre apa. Acestea au fost implementate folosind HTML, CSS si JavaScript si sunt destinate tuturor persoanelor ce isi doresc sa gaseasca in acelasi loc, atat jocuri ce stimuleaza creierul, cat si jocuri numai bune pentru relaxare.

**DEMO**

<https://drive.google.com/file/d/1QryYcpaUockDCGMKMuLXJ5bHILQqHfiX/view?usp=sharing>

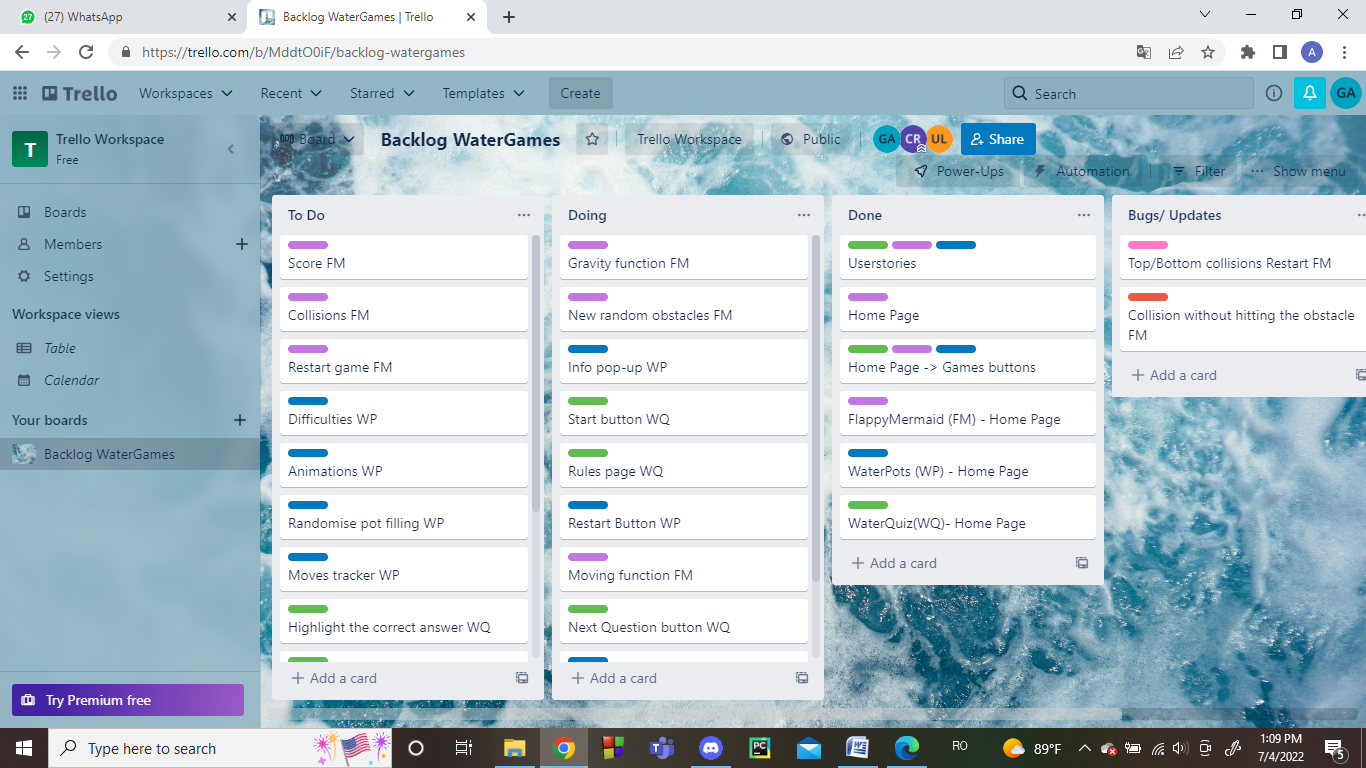
**USER STORIES**

1. As an user i would like to have a menu in which i can navigate between the games;
2. As a competitive player i would like to have a newly generated map every time i play the Flappy Mermaid game;
3. As an user i would like to see the score i got at the end of the Flappy Mermaid game;
4. As an user playing the Flappy Mermaid game i wish to restart the game whenever i want to;
5. As a person who’d like to learn thing, i would like to see the correct answer for each question in the Quiz game;
6. As a competitive player i would like to have a time tracking system in the Quiz game;
7. As a beginner i would like to have the option to play on an easy difficulty in the Water Pots game;
8. As an player i would like to be able to choose between difficulties in the Water Pots game;
9. As an user who constantly tries to beat his own highscore i would like to keep track of the number of moves in the Water pots Game;
10. As an user i would like to be able to go back to the main menu middle-game;
11. As a person trying to improve my skills i would like the colors to come random in the WaterPots game.

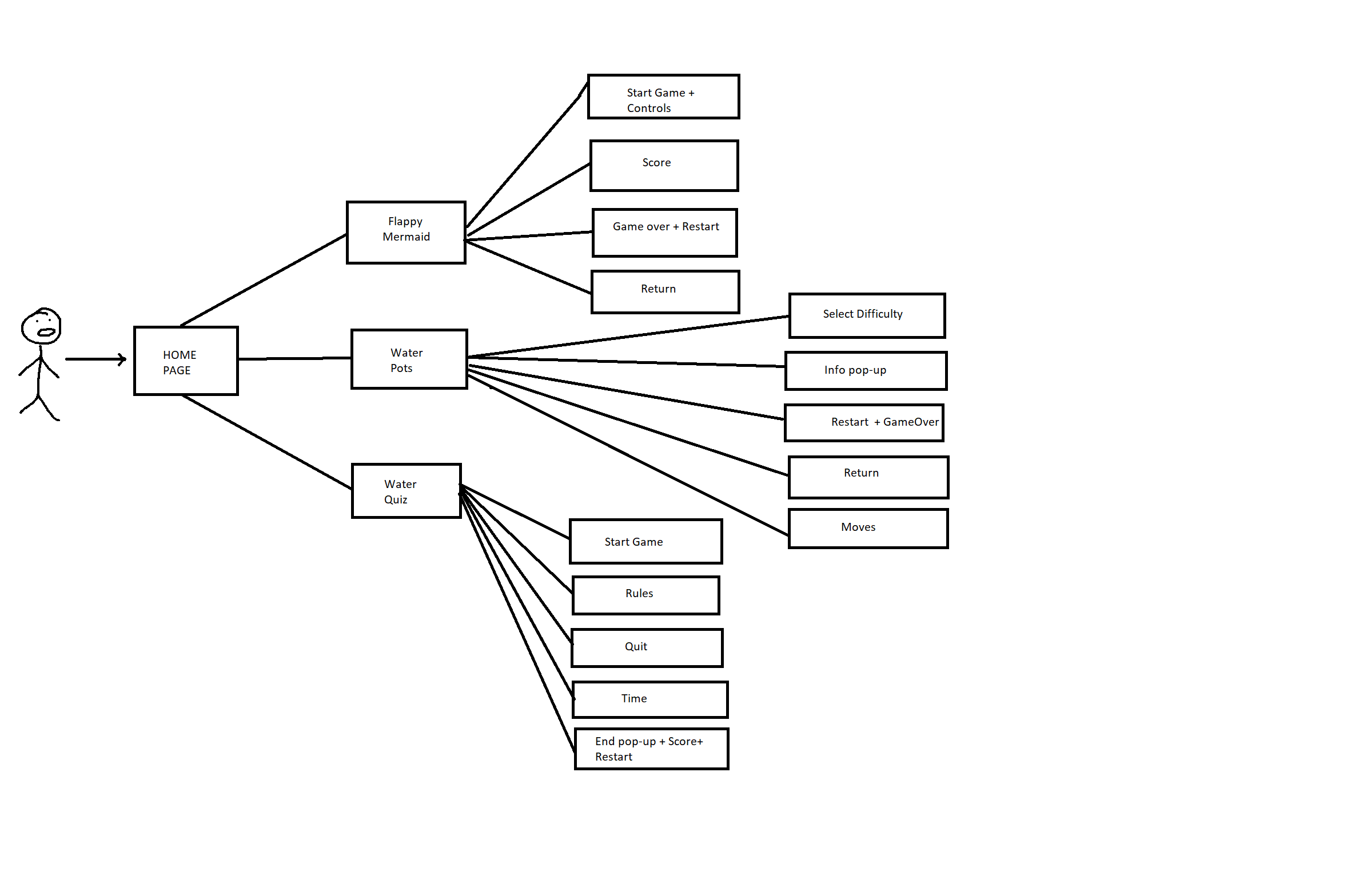
**BACKLOG**

Link: <https://trello.com/b/MddtO0iF/backlog-watergames>

Acesta a fost realizat pe Trello cu scopul de a tine evidenta task-urilor, astfel incat impartirea acestora in echipa sa fie cat mai eficienta.

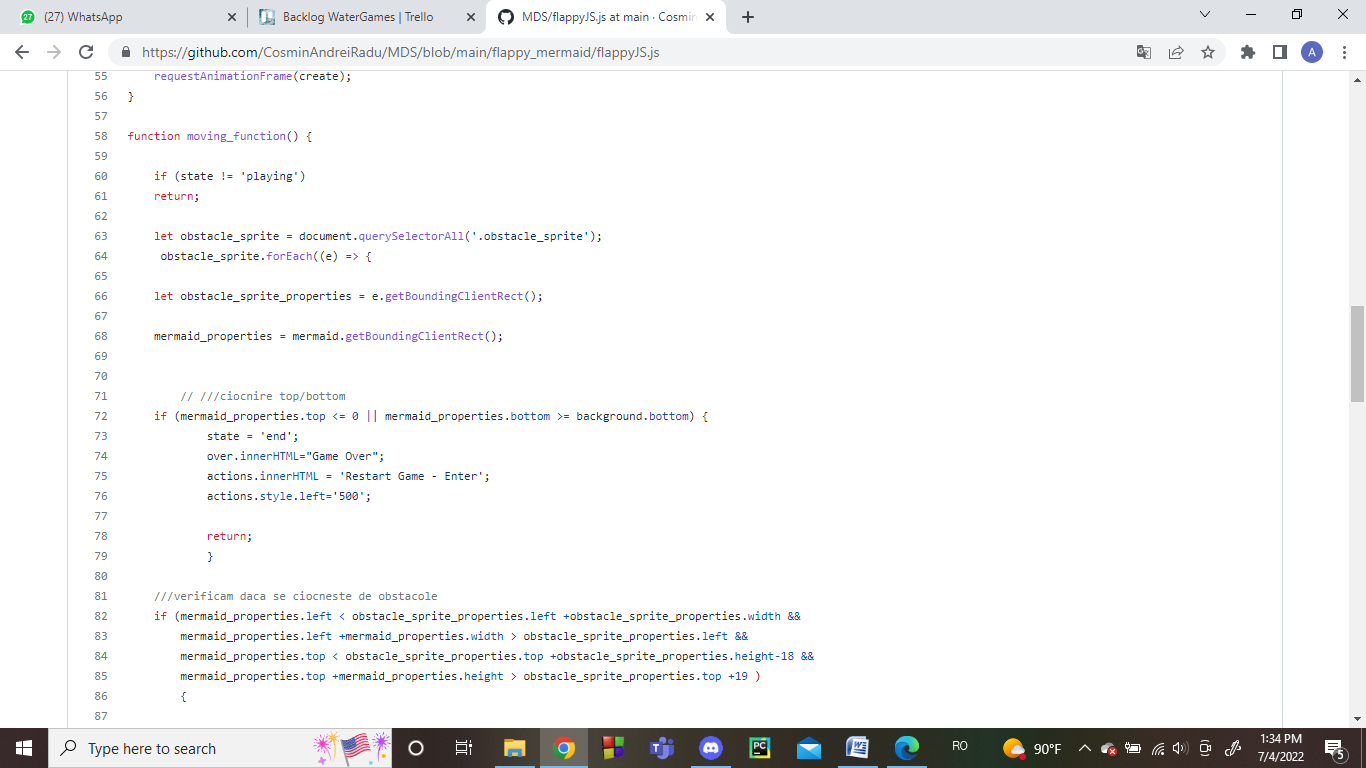


**Diagrama UML**

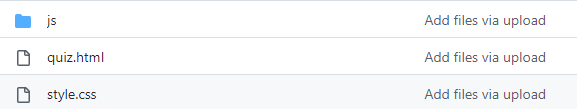


**REFACTORING:**

Refactoring: pentru rezolvarea bug-ului “[**Bug-FlappyMermaid-Restart**](https://github.com/CosminAndreiRadu/MDS/issues/1)” am reorganizat ierarhic functiile ;



**Code Standards**: pentru fiecare limbaj am folosit fisiere separate, asa cum recomanda documentatia Web Standards , variabile sugestive, spatiere si indentare, fiecare sectiune avand asignata un id/clasa/tag. De asemenea, am optimizat codul ineficient acolo unde am remarcat ca se repeta linii de cod.



**SOURCE CONTROL** <https://github.com/CosminAndreiRadu/MDS>

Am folosit platforma Github si am creat un repository la care suntem toti colaboratori:

Branch creation : cate unul pentru fiecare joc

Merging: folosind pull request-uri revizuite de toti membri echipei

**BUG REPORTING**

In cadrul dezvoltarii proiectului am gasit doua bug-uri pe care le-am raportat deschizand issues pe Github, dar si prin listarea acestora pe Trello.

**DESIGN PATTERNS**

Am folosit design pattern-ul “Module” pentru partea de JavaScript , iar pentru CSS, design pattern-ul “Atomic Style”, neavand nevoie sa mai scriem cod pentru toate elementele ci doar le atribuim o clasa.



**TESTE AUTOMATE**

Pentru partea de JavaScript am folosit framework-ul Cypress(<https://www.cypress.io/>), iar pentru partea HTML si CSS am testat folosind libraria Quixote care foloseste unit testing direct pe fisierele CSS. Rezolvarea bug-urilor intampinate nu a reprezentat o problema, datorita stanadardelor de codare folosite, modificand doar cate un fisier din aplicatie.