**Introducere**

Proiectul prezentat in continuare este un flight simulator 2D realizat cu Arduino la care a fost conectat un accelerometru. Arduino este programat să recunoască datele trimise de la accelerometru și să le trimită mai departe pe portul serial. Portul serial este in continuare citit in mediul Processing, care reprezintă și interfața grafică a proiectului. În mediul Processing a fost codat un joc de evitare de obstacole, constituit dintr-o nava ce este controlată de utilizator și din niște asteroizi ce trebuiesc a fi evitați.

**Implementare Processing**

Pentru proiectul de simulare zbor am implementat in processing un joc de evitare de obstacole. Jucătorul are la dispoziție o nava spațială, iar sarcina acestuia este de a evita asteroizii care îi sunt puși în cale. Jocul a fost implementat in limbajul de programare java.

Descrisă pe larg, aplicația se conectează la portul serial de unde citește datele transmise de arduino prin intermediul accelerometrului. Acesta parsează inclinația pe cele trei axe de coordonate și deplasează nava corespunzător. De asemenea, jocul oferă posibilitatea măririi sau scăderii vitezei de deplasare a navei dacă jucătorul efectuează un tap, respectiv double tap. O descriere in detaliu a claselor programului precum și a pașilor efectuați de acesta o voi realiza in continuare.

Aplicația definește trei clase: Ship, Asteroid și Block. Aceste clase conțin informațiile specifice pentru cele trei obiecte modelate. Pentru Ship, acestea sunt poziția, viteza precum și hit blocurile acesteia. Pentru Asteroid, acestea sunt poziția și viteza. Pentru Block informațiile conținute sunt marginile acestuia.

În pasul de inițializare, aplicația instanțiază obiectele necesare și se conectează la portul serial pentru a citi datele de la arduino. În faza de citire, jocul determină tipul input-ului (poate fi de tip accelerare, tap, etc.) și i-a acțiunea corespunzătoare acesteia. Dacă acțiunea este de tip accelerare, se determină in ce direcție trebuie mutată nava, urmând ca poziția acesteia să fie updatată corespunzător. Dacă acțiunea este de tip tap sau double tap, viteza de deplasare a navei crește, respectiv descrește. La fiecare mișcare a navei marginile hit block-urilor acesteia sunt recalculate. De asemenea, se asigură că nava nu părăsește ecranul.

Determinarea direcției de deplasare a navei se realizează cu ajutorul unor praguri predefinite de sensibilitate. Dacă una din coordonatele trimise de arduino depășește un prag înseamnă că utilizatorul a inclinat placa în direcția respectivă și dorește deplasarea pe acea rută.

Pe lângă updatarea navei, aplicația se ocupă și de generarea asteroizilor. Numarul maxim de asteroizi prezenți pe ecran este predefinit, in acest moment la 5, urmând ca in momentul in care un asteroid parasește marginile ecranului, un alt asteroid să îi ieie locul din partea superioară a jocului.

Un alt aspect care este tot procesat la nivel de processing este detectarea coliziunii. Atât timp cât jocul este încă în desfășurare, la fiecare pas, marginile hit block-urilor navei sunt comparate cu asteroizii prezenți pe ecran. Dacă este prezentă o coliziune, atunci jocul se sfârșeste.

**Concluzie**

În concluzie, proiectul a constat în realizarea unui flight simulator 2D. Au fost utilizate placa Arduino la care a fost conectat un accelerometru, precum și mediul de programare Processing, in care a fost realizat un joc de evitare obstacole.