# Documentatie Proiect LEX si Yacc

## Sushi GO

Student: Popa Alexandru Vasile | Grupa: 6 | Semigrupa 2

## 1. Descrierea Proiectului

Proiectul meu este un joc inspirat de Sushi Go, utilizând două parsere. Primul parser, *./sushi_go*, permite alegerea ingredientelor și desfășurarea jocului împotriva unui oponent virtual. Al doilea parser, *./score*, calculează scorul final pentru ambii jucători (player și AI) pe baza deck-urilor utilizate. Obiectivele principale includ o interfață command-line interactiva și un sistem de calcul al scorului.


### Analiza Lexicală

Analiza lexicală este gestionată de fișierele `sushi.l` și `score.l`, care utilizează specificatori de reguli pentru a identifica tokeni din intrarea utilizatorului. Aceste fișiere sunt scrise în limbajul de specificare flex și folosesc reguli regex pentru a recunoaște diferite tipuri de tokeni. Tokenii identificați sunt apoi utilizați în analiza sintactică.

- Componente Principale:
  - `sushi.l`: Identifică tokenii pentru comenzi în jocul de sushi.

- `score.l`: Identifică tokenii pentru scorul obținut din combinațiile de ingrediente.

### Analiza Sintactică

Analiza sintactică este gestionată de fișierele *sushi.y* și *score.y*, care folosesc gramatici pentru a interpreta structura și semantica comenzilor utilizatorului. Aceste fișiere sunt scrise în limbajul de specificare Bison/Yacc și definesc regulile gramaticale pentru interpretarea comenzilor utilizatorului și calcularea scorului.

- Componente Principale:
  - *sushi.y*: Interpretarea comenzilor utilizatorului în jocul de sushi și gestionarea jocului.
  - *score.y*: Interpretarea scorului obținut din combinațiile de ingrediente și gestionarea punctajului final al jocului.

### Interacțiune

Analiza lexicală furnizează tokenii identificați către analiza sintactică, care folosește acești tokeni conform regulilor gramaticale definite pentru a interpreta comenzile utilizatorului și a gestiona starea jocului.

## 2.2 Instrumente și Tehnologii Utilizate

Flex și Bison pentru analiza lexicală și sintactică, C pentru implementare, și biblioteci standard pentru operații de fișiere.

## 2.3 Configurația Mediului de Dezvoltare

Nici o configuratie speciala.
`flex 2.6.4` `bison (GNU Bison) 3.8.2`

# 3. Detalii Implementare

## 3.1 Implementarea Componentelor Cheie

Implementarea componentelor principale în proiectul dat constă în două părți esențiale: analiza lexicală și sintactică și calculul scorului în jocul Sushi Go.

În sushi.l și sushi.y, se realizează analiza lexicală și sintactică pentru comenzi precum "PICK", și "PASS", folosind Flex și Bison. Comenzile sunt asociate cu acțiuni din jocul Sushi Go, cum ar fi alegerea de cărți și afișarea stării curente a jocului.

- Funcția `ai_pick_card()` simulează alegerile AI-ului.
- La finalul rundei, mâinile jucătorilor sunt interschimbate.
- La încheierea jocului, punțile de cărți sunt salvate în `decks.txt`.
- Structura `GameState` gestionează starea și interacțiunile jocului.
- Funcția `initialize_game()` inițializează scorurile și afișează un mesaj de început.

În score.l și score.y, se implementează calculul scorului pentru combinațiile de cărți în funcție de regulile jocului. Se folosesc Flex și Bison pentru a analiza secvențele de cărți și a le asocia cu scoruri specifice, evidențiind combinațiile bune și proaste.

- Cuvintele Cheie "PLAYER DECK" și "AI DECK" construiesc punți de cărți pentru fiecare jucător.
- Combinațiile de cărți adaugă puncte, gestionate de `update_score()`.
- La final, scorurile finale ale jucătorilor sunt afișate.
- Funcția `initialize_game()` inițializează scorurile și afișează un mesaj de început.

# 4. Testare și Validare

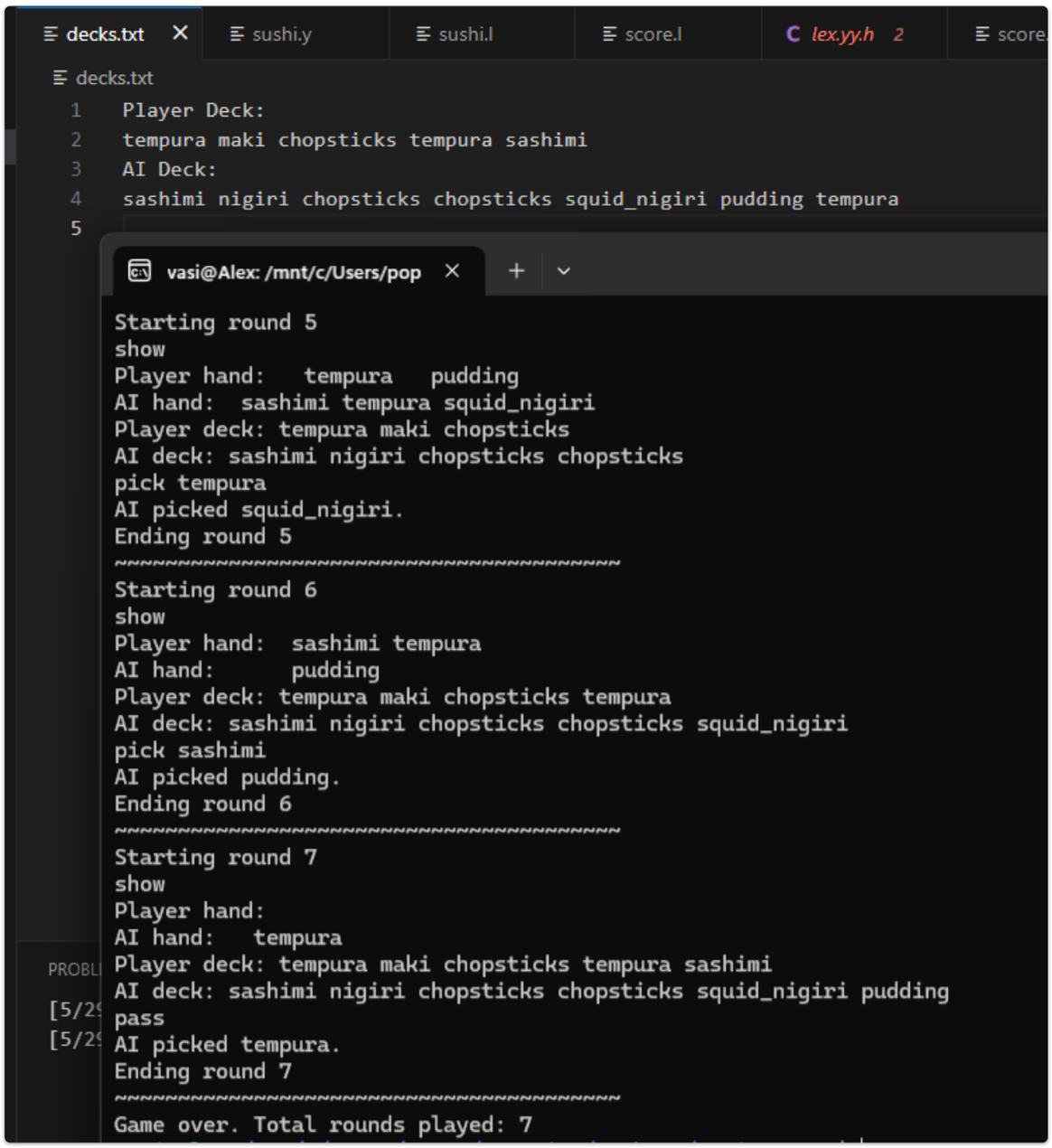
## 4.1 Cazuri de Test și Rezultate

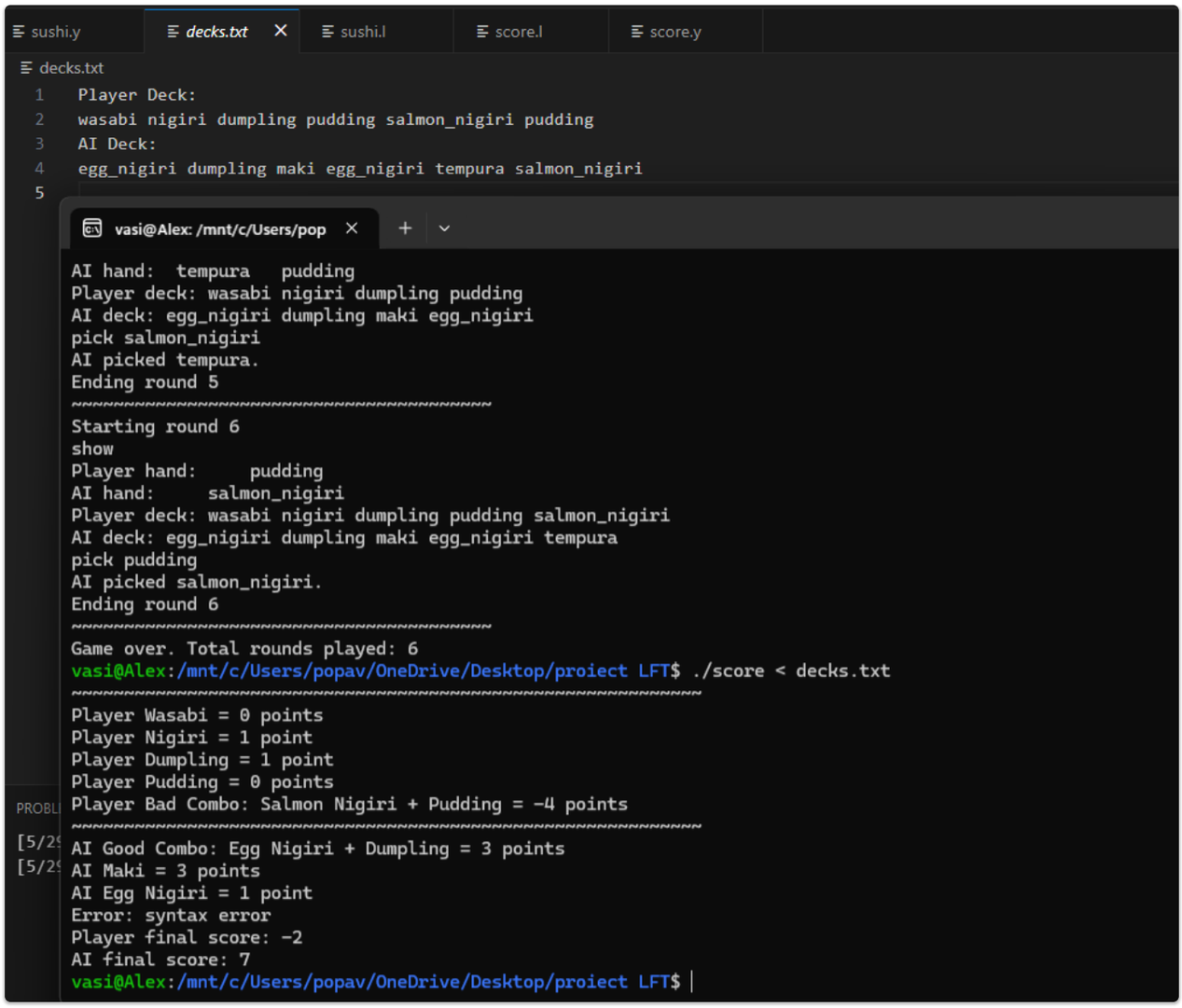
- ./sushi_go - use show and pick and just play
- ./score :

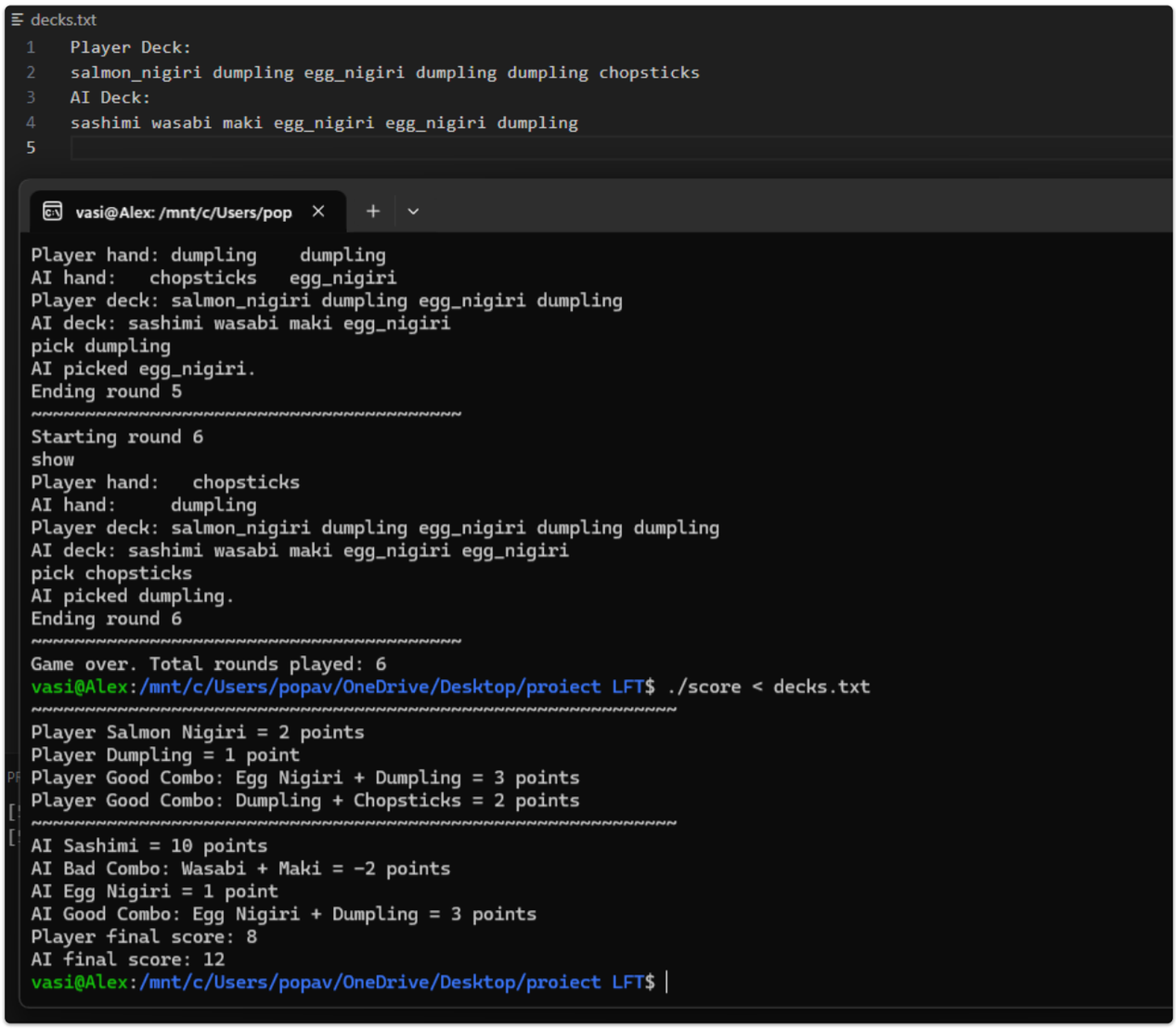
```
Player Deck:
sashimi egg_nigiri dumpling salmon_nigiri tempura wasabi
AI Deck:
dumpling pudding tempura egg_nigiri salmon_nigiri wasabi
```

```
Player Deck:
maki tempura sashimi dumpling salmon_nigiri squid_nigiri egg_nigiri nigiri pudding wasabi chopsticks
AI Deck:
maki tempura sashimi dumpling salmon_nigiri squid_nigiri egg_nigiri nigiri pudding wasabi chopsticks
```

## Screnshoturi Reprezentative:






5.1 Puncte Forte:

- Implementarea lexicală și sintactică eficientă cu Flex și Bison.
- Funcționalitatea de a construi și de a gestiona punțile de cărți pentru fiecare jucător.
- Calculul punctajului bazat pe combinațiile de cărți în conformitate cu regulile jocului.
- Utilizarea eficientă a mesajelor pentru a informa jucătorii despre acțiunile și scorurile lor.
- Structura modulară a codului, ușurând extinderea și îmbunătățirea viitoare.

5.2 Puncte Slabe:

- Lipsa unei interfețe grafice poate face jocul mai puțin accesibil sau mai puțin atractiv pentru anumiți utilizatori.
- Absența gestionării excepțiilor poate duce la comportamente neașteptate în caz de erori sau intrări nevalide.

5.3 Dezvoltări Ulterioare:

- Implementarea unei interfețe grafice pentru a îmbunătăți experiența utilizatorului.
- Adăugarea de funcționalități precum modul multiplayer sau modul single-player împotriva unui AI mai complex.
- Extinderea regulilor jocului și introducerea de noi combinații de cărți pentru a diversifica gameplay-ul și a-l face mai captivant.

6. Concluzii:

- Proiectul a reușit să implementeze funcționalitățile de bază ale jocului Sushi Go! utilizând Flex și Bison.
- Obiectivele inițiale au fost atinse prin crearea unei aplicații funcționale, dar există spațiu pentru îmbunătățiri și extinderi.