

3 משחק זוגי/אי-זוגי

3.1 תיאור המשחק

משחק זוגי/אי-זוגי הוא משחק פשוט. המשחק מתאים להדגמת פרוטוקול הליגה.

3.1.1 חוקי המשחק

1. שני שחקנים משתתפים במשחק.

2. כל שחקן בוחר "זוגי" (even) או "אי-זוגי" (odd).

3. הבחירה נעשית במקביל, בלי לדעת את בחירת היריב.

4. השופט מגיריל מספר בין 1 ל-10.

5. אם המספר זוגי – מי שבחר "זוגי" מנצח.

6. אם המספר אי-זוגי – מי שבחר "אי-זוגי" מנצח.

7. אם שניהם בחרו אותו דבר ופספסו – תיקו.

3.1.2 דוגמה למשחק

נניח משחק בין שחקן A לשחקן B:

טבלה 7: דוגמה למשחק זוגי/אי-זוגי

בחירה A	בחירה B	מספר	תוצאה
A	odd	8 (even)	even
B	odd	7 (odd)	even
תיקו	odd	4 (even)	odd

3.2 זרימת המשחק בודד

3.2.1 שלב 1: הזמנה למשחק

השופט שולח הזמנה לשני השחקנים. ההזמנה כוללת:

- מזהה המשחק (match_id).
- מזהה המחזoor (round_id).
- סוג המשחק (game_type).

3.2.2 **שלב 2: אישור הגעה**

כל שחקן מאשר קבלת הזמנה. האישור כולל חותמת זמן.

3.2.3 **שלב 3: איסוף בחריות**

השופט פונה לכל שחקן בנפרד. הוא מבקש בחירה: "זוגי" או "אי-זוגי". השחקן מציין את בחרתו.

חשוב: השחקנים לא רואים את בחרת היריב.

3.2.4 **שלב 4: הגרלת מספר**

לאחר קבלת שתי הבחירה, השופט מגറיל מספר. המספר הוא בין 1 ל-10. ההגרלה חייבת להיות אקראית.

3.2.5 **שלב 5: קביעת מנצח**

השופט בודק:

- אם המספר זוגי ושחקן בחר "זוגי" – הוא מנצח.
- אם המספר אי-זוגי ושחקן בחר "אי-זוגי" – הוא מנצח.
- אם שניהם ניחשו נכון/לא נכון – תיקו.

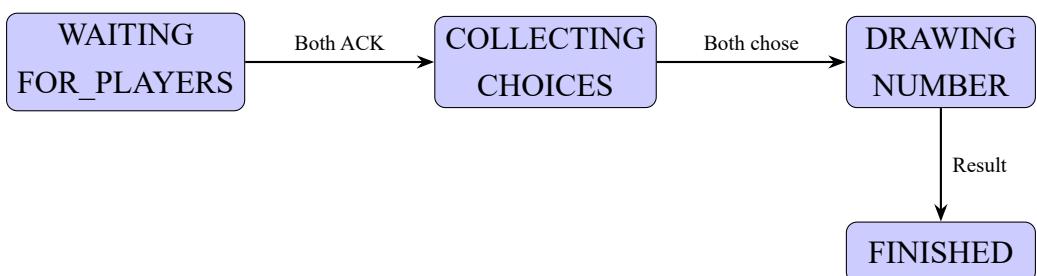
3.2.6 **שלב 6: דיווח תוצאה**

השופט שולח:

1. הודעה GAME_OVER לשני השחקנים.
2. הודעה MATCH_RESULT_REPORT למנהל הליגה.

3.3 **מצבי המשחק**

המשחק עובר בין מצבים מוגדרים:



3.3.1 **מצב WAITING_FOR_PLAYERS**

המשחק מתחילה במצב זה. השופט מחייב שהשחקנים יאשרו הגעה. המעבר: כששני השחקנים שלחו GAME_JOIN_ACK

3.3.2 מצב COLLECTING_CHOICES

השופט אוסף בחירות מהשחקנים. הוא קורא ל-choose_parity של כל שחקן. המעבר: כשתמי הבחירה התקבלו.

3.3.3 מצב DRAWING_NUMBER

השופט מגיריל מספר וקובע מנצח. המעבר: אוטומטי לאחר החישוב.

3.3.4 מצב FINISHED

השחקן הסתiens. התוצאה דוחה.

3.4 שיטת הנקודות

3.4.1 ניקוד למשחק

טבלה 8: טבלת ניקוד

נקודות מפסיד	נקודות מנצח	תוצאה
0	3	נצחון
1	1	תיקו
0	0	הפסד

3.4.2 דירוג בליגה

הדירוג קבוע לפי:

1. סך הנקודות (יורד).
2. מספר הניצחונות (יורד).
3. הפרש תיקו (יורד).

3.5 ליגת Round-Robin

בליגה עם 4 שחקנים, כל שחקן משחק נגד כולם.

3.5.1 מספר משחקים

עבור n שחקנים:

- מספר משחקי בליגה: $\frac{n(n-1)}{2}$

- עבור 4 שחקנים: $6 = \frac{4 \times 3}{2}$ משחקים

3.5.2 לוח משחקים לדוגמה

טבלה 9: לוח משחקים ל-4 שחקנים

שחקן B	שחקן A	משחק
P02	P01	R1M1
P04	P03	R1M2
P03	P01	R2M1
P04	P02	R2M2
P04	P01	R3M1
P03	P02	R3M2

3.6 אסטרטגיות לשחקנים

3.6.1 אסטרטגיה אקראית

הגישה הפשוטה ביותר. השחקן בוחר באקראי "זוגי" או "אי-זוגי". הסיכוי לניצח הוא 50%.

סטרטגיה אקראית

```
import random

def choose_parity_random():
    return random.choice(["even", "odd"])
```

3.6.2 אסטרטגיה מבוססת היסטוריה

השחקן זכר תוצאות קודמות. הוא מנסה לאוזות דפוסים בהגרלות.
הערה: מכיוון שההגרלה אקראית, אסטרטגיה זו לא תשפר תוצאות לטווח ארוך.

3.6.3 אסטרטגיה מונחית LLM

השחקן יכול להשתמש במודל שפה. הוא בונה prompt וسؤال את המודל.

דוגמה ל-prompt

```
prompt = """
You\u201are\u201aplaying\u201aEven/Odd\u201agame.
Choose\u201a"even"\u201aor\u201a"odd".
Previous\u201aresults:\u201aeven\u201awon\u201a3\u201atimes,\u201aodd\u201awon\u201a2\u201atimes.
Your\u201achoice\u201a(one\u201aword\u201aonly):
"""
```

הערה: השימוש ב-LLM מעניין אך לא ישר ביצועים סטטיסטיות. המשחק הוא משחק מזל.

3.7 מודול חוקי המשחק

מודול החוקים הוא רכיב נפרד בשופט. הוא מגדר את הלוגיקה הספציפית למשחק.

3.7.1 ממשק המודול

המודול מספק פונקציות:

- `init_game_state()` – אתחול מצב משחק.
- `validate_choice(choice)` – בדיקת חוקיות בחירה.
- `draw_number()` – הגרלת מספר.
- `determine_winner(choices, number)` – קביעת מנצח.

3.7.2 יתרון ההפרדה

בעתיד, אפשר להחליף את המודול. במקומות זוגי/אי-זוגי, אפשר:

- איקס-עיגול (Tic-Tac-Toe).
- 21 שאלות.
- משחק זיכרון.

הפרוטוקול הכללי נשאר זהה. רק מודול החוקים משתנה.

3.8 הרחבה למשחקים נוספים

הפרוטוקול מתוכנן להיות כללי ולא ספציפי למשחק זוגי/אי-זוגי. סעיף זה מתאר את השכבה הגנרייה המאפשרת הוספת משחקי נוספים.

3.8.1 הפעטה מהלך – GAME_MOVE

הודעות CHOOSE_PARITY_RESPONSE ו-CHOOSE_PARITY_CALL הן מקרה פרטי של הפעטה כללית יותר:

טבלה 10: הקבלה בין הודעות ספציפיות לגנריות

הודעה גנרית	הודעה ספציפית
GAME_MOVE_CALL	CHOOSE_PARITY_CALL
GAME_MOVE_RESPONSE	CHOOSE_PARITY_RESPONSE

בקשת מהלך גנריית - GAME_MOVE_CALL

```
{
  "protocol": "league.v2",
  "message_type": "GAME_MOVE_CALL",
  "sender": "referee:REF01",
  "timestamp": "2025-01-15T10:30:15Z",
  "match_id": "R1M1",
  "player_id": "P01",
  "game_type": "even_odd",
  "move_request": {
    "move_type": "choose_parity",
    "valid_options": ["even", "odd"],
    "context": {}
  },
  "deadline": "2025-01-15T10:30:45Z"
}
```

תגובה מהלך גנריית - GAME_MOVE_RESPONSE

```
{
  "protocol": "league.v2",
  "message_type": "GAME_MOVE_RESPONSE",
  "sender": "player:P01",
  "timestamp": "2025-01-15T10:30:20Z",
  "match_id": "R1M1",
  "player_id": "P01",
  "game_type": "even_odd",
  "move_data": {
    "move_type": "choose_parity",
    "choice": "even"
  }
}
```

3.8.3 רישום סוגי משחקים – Game Registry

מנהל הליגה מחזיק רישום של סוגי משחקים נתמכים:

רישום משחקים

```
{  
    "game_registry": {  
        "even_odd": {  
            "display_name": "Even/Odd",  
            "move_types": ["choose_parity"],  
            "valid_choices": {  
                "choose_parity": ["even", "odd"]  
            },  
            "min_players": 2,  
            "max_players": 2  
        },  
        "tic_tac_toe": {  
            "display_name": "Tic-Tac-Toe",  
            "move_types": ["place_mark"],  
            "valid_choices": {  
                "place_mark": ["0-8"]  
            },  
            "min_players": 2,  
            "max_players": 2  
        }  
    }  
}
```

3.8.4 יתרונות ההפרשה

1. **הוספת משחקים חדשים** – ללא שינוי בפרוטוקול הבסיסי.
2. **גילוי יכולות** – שחקן יכול לשאול אילו משחקים נתמכים.
3. **ולידציה אחידה** – השופט מודזע שהמהלך חוקי לפי ה-.schema
4. **תאימות קדימה** – סוכנים שונים יכולים להמשיך לעבוד עם הודעות ספציפיות.

הערה: בתרגיל זה משתמשים בהודעות הספציפיות (*_CHOOSE_PARITY). ההפרשה הגנרטית מוצגת לצורך הבנת הארכיטקטורה.