

# נושאים מתקדמים בפיתוח תוכנה – פרוייקט



## המערכת:

הדמיית משחק פק-מן, מיועד ל-1-2 שחקנים.

פק-מן הינו משחק בו קיים יצור עגלגל וצהבהב שמטרתו לאכול את כל הסוכריות הפזורות על גבי לוח המשחק הבנוי בצורת מבוך. במהלך המשחק מסתובבות על גבי המבוך רוחות רפאים אכזריות שפוגעות בחייו של פק-מן.

כל שלב במשחק מתחיל כאשר כל הלוח (בין חלקי המבוך) מלא בסוכריות צהובות וכאשר מסתובבות בו מספר רוחות רפאים המתחילות את תנועתן במיקומים אקראיים.

מטרת המשחק – לאכול את כל הסוכריות הפזורות על הלוח, התחמקות מרוחות הרפאים וצבירה של כמה שיותר נקודות.

מצב משחק מרובה משתתפים:

במצב זה יצטרף עוד פק-מן ללוח ויתחרה מול השחקן הראשון על הנקודות. מי שהשיג יותר נקודות הוא המנצח.

## קישור למשחק לדוגמא

## מרכיבי המערכת:

### • פק-מן

- יצור עגלגל, צהוב וחייכן.
- בתחילת כל משחק וכל שלב פק-מן מתחיל ממקום קבוע מראש על גבי לוח המשחק.
- פק-מן נע על גבי הלוח במטרה לאכול את כל הסוכריות על גבי לוח המשחק תוך הימנעות ממגע עם רוחות הרפאים.
- בעת התחלת המשחק לפק-מן ישנם 3 נקודות חיים. וניקוד התחלתי 0.

### • רוחות רפאים

- במהלך המשחק יסתובבו על גבי לוח המשחק, רוחות רפאים. מיקומן ההתחלתי של רוחות הרפאים הוא הבסיס שלהן במרכז הלוח באיזור שפק-מן לא יכול להגיע אליו.
- קיימים 2 סוגים של רוחות רפאים:
  - רוח רפאים רגילה - רוח רפאים שמסתובבת על גבי הלוח מתחילת המשחק ועד סופו. מגע עם רוח רפאים רגילה תגרור פסילה של השחקן.
  - רוח רפאים אכילה - רוח רפאים שפק-מן יכול לאכול. לאחר אכילת סוכריית שאלה, למשך זמן מוגבל כל רוח רפאים רגילה הופכת לאכילה. אכילת רוח רפאים כזו מייצגת בחירה בתשובה לשאלה שתצוץ מיד לאחר אכילת סוכריית שאלה.

### • סוכריות צהובה

- ממלאות את כל התאים הפנויים על גבי הלוח, נעלמות בעת אכילתן על ידי פק-מן ומזכות את פק-מן בנקודה אחת.

## • סוכריות שאלה

קיים מספר מוגבל וקבוע שלהן על הלוח, אכילתן תפעיל למשך זמן קצוב את מצב השאלה במשחק. במצב שאלה:

- מוצגת שאלת ברירה מרובה ממאגר השאלות באופן רנדומלי במקביל לריצת המשחק
- כל אחת מרוחות הרפאים הופכת לאכילה, וממשיכה לנוע כרגיל.
- כל תשובה אפשרית מיוצגת ע"י רוח רפאים אכילה.
- אכילת רוח המייצגת תשובה נכונה תעניק לשחקן ניקוד בשווי השאלה ותפסיק את מצב השאלה.
- אכילת רוח המייצגת תשובה לא נכונה תוריד לשחקן ניקוד בסך חצי משווי השאלה, ותעביר את רוח הרפאים שנאכלה לבסיס הרוחות במרכז הלוח שם היא תחכה עד שמצב השאלה ייסתיים, או שמצב שאלה חדש יתחיל.
- אם הזמן הקצוב הסתיים הרוחות יהפכו מאכילות לרגילות.
- אכילת סוכריית שאלה בזמן מצב שאלה, תפעיל מחדש את מצב השאלה עם שאלה חדשה.
- רוחות רפאים שנמצאות בבסיס שלהן יהיו חופשיות לצאת.
- שווי השאלות הוא: קלה – 100 נקודות, בינונית – 200 נקודות, קשה – 400 נקודות.

## • קירות

מהווים חסמים שלא ניתנים למעבר. לא ניתן למקם סוכריות על קירות ולא ניתן לעבור דרך קירות.

## • שאלות

השאלות שיוצגו בעת אכילת סוכריית שאלה יטענו מתוך קובץ json שיסופק על ידי צוות הקורס. אין להקים בסיס נתונים. יש להשתמש בקובץ ה- json בלבד. אין לשנות את מבנה הקובץ. כל שאלה חייבת להכיל 4 תשובות אפשריות לכל שאלה תשובה נכונה אחת בלבד.

## • סיום המשחק

- המשחק יכול להסתיים בכמה דרכים:
- כל הסוכריות בלוח נאכלו.
- במצב מרובה משתתפים – המנצח יהיה השחקן בעל מספר הנקודות הגבוה ביותר.
- לשחקן לא נותרו יותר פסילות.
- במצב מרובה משתתפים – כאשר יש שחקן יחיד שנותרו לו פסילות, במקרה זה הוא המנצח.

## • זרימת התהליך

1. מסך פתיחה:  
כולל את הפעולות – התחלת משחק במצב יחיד, התחלת משחק במצב מרובה משתתפים, צפייה בהיסטורית המשחקים, ניהול שאלות.
2. התחלת משחק: הזנת כינוי לשחקן.
3. הצגת לוח המשחק:  
לוח המשחק יציג את מהלך המשחק וגם:
  - כמות חיי הפק-מן
  - ניקוד נוכחי
  - זמן מתחילת המשחק
  - במצב שאלה: את השאלה, זמן השאלה ואת התשובות האפשריות.

4. מהלך המשחק:  
 פק-מן מתחיל במקום קבוע ← זולל את כל הסוכריות הפזורות על הלוח ללא חשיבות לסדר ←  
 כשהמשחק נגמר השעון יעצר ויוצג הניקוד.
5. כלי ניהול שאלות:  
 מסך ניהול שיאפשרו הוספה, עריכה, ומחיקת שאלות למשחק. כל השאלות יישמרו בקובץ json.

#### • הנחיות נוספות

- הפרוייקט יכלול צד שרת הכתוב בשפת JavaScript בסביבת ריצה של Node.js
- הפרוייקט יכלול צד לקוח שהוא בעצם אתר אינטרנט הבנוי מ- Html Css ו- JavaScript.
- צד השרת יעשה שימוש ב- Express כ- web framework, ו- socket.io עבור תקשורת זמן אמת למצב מרובה משתתפים.
- צד הלקוח יעשה שימוש ב- Pixi.js עבור רינדור המשחק.
- Source control יהיה באמצעות GitHub
- נתוני השאלות ישמרו בקובץ json עם סכמה קבועה.
- הפרמטרים לבדיקה יכללו: מבנה נכון יעיל ודינאמי של התוכנית, פונקציונליות תקינה ומלאה, חלוקה נכונה של ארכיטקטורת התוכנה (מתודות, מחלקות), שימוש ב- Design Patterns, תיעוד ברור ומלא, ממשק נקי ונוח למשתמש, בדיקת מקרי קצה.
- תוספות והשקעה ייתרה יהוו בonus.
- בכל הגשה בכל אחת מהאיטרציות **תרגיל שלא ירוץ ולא יתקמפל הציון עליו יהיה 0**
- על אחריות הסטודנטים לוודא תקינות הפרוייקט על יותר ממחשב אחד טרם ההגשה.

#### • פירוט הבונוסים

- ינתן בonus עד 15 נקודות לציון הפרוייקט הסופי.
- לפי תחרות שתיערך בסוף הסמסטר, ייבחרו 3 פרוייקטים מצטיינים שייקבלו בonus נוסף:
- מקום ראשון – 3 נקודות לציון הסופי בקורס
- מקום שני – 2 נקודות לציון הסופי בקורס
- מקום שלישי – נקודה 1 לציון הסופי בקורס

#### • הצעות לבonus

- תוספות למשחק
- לוח דינמי
- מרובה משתתפים עם יותר מ-2 שחקנים
- שימוש במיקרופון \ מצלמה
- בינה מלאכותית
- ושאר רעיונות ייצירתיים משלכם.