

## מטלת ביצוע - פיתוח משחק Snake

### 1. המטלה:

יש לתכנן ולפתח משחק Snake. במשחק נחש שנע במהירות על המסך בחיפוש אוכל. בכל פעם שהנחש תופס אוכל, הוא מתארך ומהירותו גדלה, ודבר מאכל חדש מופיע על המסך. המשחק מסתיים כאשר הנחש יוצא מגבולות המסך או מתנגש בגופו. בכל פעם שהנחש תופס אוכל, השחקן מקבל נקודות והמטרה היא לצבור כמה שיותר נקודות לפני שנפסלים!

המשחק צריך להיות יפה ויזואלית וקל לשימוש. ניתן להשתמש בתמונות, צבעי רקע ואפילו מוסיקה!

המשחק יפותח על גבי פלטפורמה מיוחדת המכילה מחלקות מוכנות ומאפשרת בקלות להציג עצמים ויזואליים על משטח משחק, להזיז אותם, להשמיע מוזיקה, להציג טקסט על המסך וכו'...

### הוראות לאופן ביצוע המטלה:

המטלה תתבצע בקבוצות של 2-4 תלמידים. חלק ממטרת המטלה היא למידה והפרייה הדדית. העבודה תתבצע בשני שלבים:

1. שלב התכנון והמחקר: בשלב זה ינותחו הדרישות, יתבצע מחקר לגבי אופן השימוש במחלקות השונות בפלטפורמה או כל מחלקה אחרת בה תבחרו להשתמש ולבסוף יתבצע תכנון של איך יראה המשחק, מה הוא יכלול ואופן הביצוע של התכנה עצמה. התכנון יתועד במסמך שיתאר לפי הצורך את התהליכים המרכזיים שתוכננו. מעקב וסימולציה לביצוע הדרישות יתבצע על מנת להגיע לתכנון אופטימלי ותקין. יש לתעד את הלבטים והשיקולים שנלקחו במהלך התכנון.

2. שלב התכנות: בשלב זה תמומש התכנית לפי הנקבע בתכנון. כל שינוי מהתכנון יתועד בחזרה במסמכי התכנון. בגמר הכתיבה תתבצע בדיקה יסודית של תקינות התכנה. במידה והעבודה מוגשת עם באגים שלא נפתרו, יש לציין אותם במסמך ההגשה.

### 2. הוראות לאופן ההגשה:

תאריך אחרון להגשה: **חודש מקבלת העבודה**

העבודה תועלה בצורה אלקטרונית כ `github repository` (הסבר על כך ניתן בנפרד) ותכלול:

1. מסמך `word אחד` הכולל:

- שער ובו שמות התלמידים המגישים
  - התיעוד של שלב התכנון
  - הסבר על אופן העבודה הקבוצתית, החלוקה למטלות ומתן הערכה עצמית לפי טבלת ההערכה המצורפת למטה.
2. פרוייקט `C#` המכיל את הקוד המלא של התכנה

### 3. אופן הערכת העבודה:

הציון הקבוצתי ינתן לפי הקריטריונים הבאים:

- אופן הלמידה והתכנון של הפלטפורמה (10%)
- כתיבת קוד קריא ומתועד. שימוש נכון בפעולות ומחלקות (30%)
- איכות הפתרון (20%)
- מראה כללי, יצירתיות ואטרקטיביות המשחק (25%)
- אופן העבודה הקבוצתית (האם היה שיתוף פעולה, הפריה הדדית וכו'...) (15%)

### פירוט המחלקות הנתונות בפלטפורמה:

4. המחלקה **Position** – מחלקה זו מגדירה מיקום על המסך בקואורדינטות x (שורה) ו-y (עמודה).  
ממשק המחלקה:

פעולה	תיאור
<code>Position(int xPos, int yPos)</code>	פעולה בונה. קובעת את השורה (xPos) והעמודה (yPos) במיקום.
<code>X</code>	תכונה שקובעת או מאחזרת את מספר השורה
<code>Y</code>	תכונה שקובעת או מאחזרת את מספר העמודה

5. המחלקה **GameObject** – מחלקה זו מאפשרת להגדיר עצם ויזואלי בצורת מלבן ולהציג אותו על המסך.  
ממשק המחלקה:

פעולה	תיאור
<code>GameObject(Position pos, int width, int height)</code>	פעולה בונה. קובעת את המיקום ההתחלתי, אורך ורוחב העצם.
<code>void SetPosition(Position p)</code>	הפעולה ממקמת על המסך את העצם במיקום p.
<code>Position GetPosition()</code>	הפעולה מחזירה את המיקום הנוכחי של העצם על המסך
<code>void SetImage(Image img)</code>	הפעולה מקבלת הפניה לתמונה ומציגה אותה על פני העצם.
<code>void SetBackgroundColor(Color c)</code>	הפעולה מקבלת צבע וצובעת את הרקע של העצם. אם משתמשים בתמונה אז לא רואים את הרקע.
<code>bool IntersectWith(GameObject obj)</code>	הפעולה מקבלת אובייקט משחק אחר ומחזירה true אם הוא ממוקם בצורה חופפת לעצם עליו מופעלת הפעולה.
<code>bool OnScreen(Board board)</code>	הפעולה מקבלת אובייקט של מסך המשחק ומחזירה true אם אובייקט המשחק נמצא בתוך הגבולות שלו.
<code>Direction</code>	תכונה שקובעת את כוון תנועתו של העצם.

6. המחלקה **TextLabel** – מחלקה זו מאפשרת להגדיר כיתוב טקסט על המסך

## ממשק המחלקה:

פעולה	תיאור
<code>TextLabel(string str, Position pos)</code>	פעולה בונה קובעת את הטקסט שיוצג ואת המיקום שלו.
<code>void SetPosition(Position p)</code>	הפעולה ממקמת על המסך את הטקסט במיקום p.
<code>Position GetPosition()</code>	הפעולה מחזירה את המיקום הנוכחי של הטקסט על המסך
<code>void SetFont(string name, int size)</code>	הפעולה קובעת את גודל הטקסט והפונט שלו
<code>void SetText(string str)</code>	הפעולה קובעת מחדש את הטקסט המוצג
<code>string GetText()</code>	הפעולה מחזירה מחרוזת ובה הטקסט שמוצג.

## 7. המחלקה Board – מחלקה זו מגדירה את משטח המשחק!

## ממשק המחלקה:

פעולה	תיאור
<code>Board.resolutionFactor</code>	תכונה סטטית שקובעת את הרזולוציה של המסך. ככל שהערך גדול יותר יש פחות פיקסלים (שורות ועמודות!). ערך ברירת מחדל הוא 1.
<code>XSize</code> <code>YSize</code>	שתי תכונות שמאחזרות או קובעות את גודל המסך (שורות ועמודות)
<code>void StartTimer(int interval)</code>	הפעולה מפעילה את הפעולה <code>GameClock</code> בתכנית שלכם כל <code>interval</code> מילי שניות
<code>void StopTimer()</code>	הפעולה מפסיקה את הקריאות לפעולה <code>GameClock</code>
<code>void AddLabel(TextLabel lbl)</code>	הפעולה מוסיפה למסך את הטקסט המוגדר ב <code>lbl</code>
<code>void AddGameObject(GameObject obj)</code>	הפעולה מוסיפה למסך את העצם <code>obj</code>
<code>void SetBackgroundColor(Color c)</code>	הפעולה מגדירה את צבע הרקע של המסך
<code>void SetBackgroundImage(Image img)</code>	הפעולה מגדירה את רקע המסך כתמונה
<code>void PlayBackgroundMusic(string audioFile)</code>	הפעולה מפעילה מוזיקת רקע שתושמע ברקע
<code>void SetGameBoardTitle(string title)</code>	הפעולה קובעת את הטקסט שיוצג בכותרת החלון של המשחק.

## 1. המחלקה GameEvents – אתם כותבים את המחלקה הזו! אתם יכולים להוסיף למחלקה זו פעולות ומשתנים כמו בדוגמא ואתם חייבים שהמחלקה תכלול את הפעולות הבאות:

פעולה	תיאור
<code>public void GameInit(Board board)</code>	לפעולה הזו שתכתבו המשחק קורא פעם אחת בהתחלה. בפעולה הזו יש להתחיל משתנים, להוסיף למשחק טקסטים או עצמים (GameObjects), לקבוע את גודל המשטח והרזולוציה, להתחיל להשמיע מוזיקת רקע, לקבוע את הנראות של המסך מבחינת הרקע, להתחיל את ה timer של המשחק וכו',....
<code>public void GameClock(Board board)</code>	לפעולה הזו המשחק יקרא בכל פרק זמן כפי שתקבעו כאשר תפעילו את ה timer על ידי הפעולה StartTimer. בפעולה זו בדרך כלל מזיזים אובייקטים ובודקים אם המשחק נגמר (יש התנגשות לדוגמא)
<code>public void KeyDown(Board board, char key)</code>	הפעולה הזו נקראת בכל פעם שהמשתמש לוחץ על מקש במקלדת. בפעולה זו בדרך כלל משנים את כוון האובייקטים בהתאם למקש שנלחץ, או עוצרים / מפעילים את ה timer וכו'...