עבודה במדעי המחשב

מגישים: יונתן וולסקי, איתמר גפר, נועה אביטוב ואלונה כבביה
29/4/24

שלב 1- שלב התכנון והמחקר: (דברים ששונו סומנו בורוד והסבר מופיע בסוף)

התחלנו בלקרוא את ההוראות בצורה יסודית, ועברנו לקרוא את המחלקות בGameObject.

לאחר מכן, המשכנו לחשוב על העיצוב והתחלנו לנסות לשים תמונות: רקע, חיה, אוכל ובונוס פוד, ומוזיקת רקע וסאונדים של אכילה. בחרנו ברעיון של כלב תחש שהולך וגדל ואוכל כלשהו (התלבטנו איך נעשה את הכלב, איך נוסיף את הגוף בצורה שתראה אסטית ואחידה, ומה נעשה עם הרגליים שלו- שלבסוף החלטנו שנגדיר ראש וגוף שיתארך כל הזמן ולפחות לבנתיים נוותר על הרגליים).

לבסוף, תכננו את המשחק במחברת.

מהלך המשחק:

<u>תחילת המשחק-</u> הTextLabel של הניקוד מתאפס, הכלב חוזר לגודל ההתחלתי (2 מקומות במערך, ראש וחלק גוף אחד), direction:RIGHT, ליצור אוכל במקום רנדומלי בגבולות המסך שהוא לא חלק מהכלב.

בונוסים:

- הודעה על תחילת המשחק (לחץ __ כדי להתחיל!)
- הודעה על הימצאות מקש השהייה
- מהירות התחלתית- איטית.
- עם לחיצה על מקש ההפעלה תתחיל להתנגן מוזיקת רקע.

<u>תנועה-</u> תנועת ברירת מחדל: ימינה (y+)

מקש שמאלה: שמאל (y-)

מקש למעלה: למעלה (x-)

מקש למטה: למטה (x+)

<u>אכילה:</u>

-תנאי- אם מיקום הכלב הקטן ומיקום האוכל שווים ((התמונה של הכלב הקטן)IntersectWith)

- הוספת גודל לנחש בהתאם לתנועה קודמת - לפי האיבר האחרון והזזה קדימה.
- הוספת אוכל- ליצור אותו במיקום רנדומלי על המסך (אחרי מחשבה הבנו שצריך להגדיר ראנדום x וראנדום y במספרים שבין 0 לרוחב/אורך של המסך +1, ולהגדיר את המיקום של האוכל במקום (x,y) ולבדוק שהוא לא נוצר על הנחש עצמו (סריקה של המערך -> IntersectWith), ונוצר בגבולות המסך.
- score- הוספת ניקוד בהתאם לאכילה- 10 נקודות על כל אוכל רגיל (שינוי בTextLable).
- bonus food- כל 5 אוכל רגיל יופיע אוכל מיוחד שמביא יותר נקודות. נבדוק מתי הוא צריך להיווצר בעזרת counter שסופר את מספר הפעמים שנוצר אוכל רגיל, וכאשר מתקיים התנאי if(counter%5==0) נוסיף 30 נקודות.
- סאונד- כשהכלב יאכל יהיה סאונד של אכילה. אם יאכל bonus food, ישמע צליל אחר.
- האצה של המשחק (להפחית מהGameTimer).
- הודעה על אכילה (בסגנון !Good job!, Excellent וכו')

<u>פסילה:</u>

<u>אופציה 1: הכלב פוגע בעצמו:</u>

הבנו שהפעם הראשונה שהכלב יכול לפגוע בכלבים האחרים (כלומר המשך המערך) תהיה כשהוא יהיה בגודל 4 (לפי התמונה המצורפת), ולכן נבצע בדיקה מהמקום השלישי במערך (אם הוא קיים) אם קיים איבר שמיקומו חופף למיקום הראש. בהמשך החלטנו לעשות את זה בעזרת פעולה בשם SelfHit שרצה על המערך ומחזירה true אם ([dog[0].IntersectWith(dog[i) ואחרת היא מחזירה false.

הסבר למה צריך לבדוק מהמקום השלישי במערך:

מקום 2	מקום 1
מקום 3	הראש (מקום 0)

כיוון התנועה ←

אופציה 2: פגיעה בגבולות המסך:
בדיקה בעזרת פעולת OnScreen אם הראש של הנחש נמצא בגבולות המסך.

סדר הפעולות: בדיקה של פסילה -> תנועה -> בדיקה אם נעשתה אכילה.

משתנים, קבועים ועצמים:

TextLabales:
TextLabel lblScore; -טקסט של הניקוד
TextLabel lblEnd;- הודעה של פסילה
TextLabel lblPause; - הודעה של השהייה
TextLabel lblStartCredit; TextLabel lblPresent;- טקסטים המופעים במסך הקרדיטים בהתחלה
TextLabel lblDifficulty; - טקסט של דרגת הקושי
TextLabel lblStart; TextLabel lblNameOfGame1; TextLabel lblNameOfGame2;- טקסטים המופיעים (במסך ההתלחתי אחרי הקרדיטים)
TextLabel lblGoodMsg; TextLabel lblSuper; - הודעות של אכילה
TextLabel superMsg- הודעה שמופיעה לפני סופר פוד

GameObjects:
GameObject food - עצם המייצג את האוכל
;GameObject paw; - (עצם המשמש למסך ההתחלתי (התמונה
GameObject[] dog = new GameObject[2]; - (מערך המייצג את איברי הנחש (כלב
GameObject upWall, leftWall, downWall, rightWall; - עצמים המייצגים את גבולות המסך

Variables:
int points = 0, - משתנה המייצג את הניקוד
Int countEaten = 0; - משתנה המייצג את כמות האכילות שנעשו
const int BONUS = 5; - (קבוע המייצג את אורך מחזוריות האוכל המיוחד (כל כמה אכילות יופיע אוכל מיוחד
const int SIZE = 4; - קבוע המייצג את הגודל של העצמים על המסך
int timerInterval = 200; - משתנה המייצג את את מהירות המשחק (ms)
bool isPaused = false - משתנה המייצג האם המשחק בהשהייה
Bool start = true - משתנה המייצג האם המשחק במסך ההתחלתי שלו
Bool gameOver = false; - משתנה המייצג האם נעשתה פסילה
int timerCounter = 0, - משתנה שמייצג את כמות המחזורים שהמשחק פעל
Int current = 0; - (משתנה ששומר מספר מחזור מסויים למען חישובים (בשביל הודעות האכילה
const int START = 20, MID = 160, HARD = 110; - קבועים שמייצגים את כמות הנקודות שצריך בשביל כל רמת קושי.
string difficulty = "difficulty: easy";- משתנה המייצג את רמת הקושי המשחק
const string FontNotForTitle = "Aharoni"; - קבוע המייצג את הפונט של ההודעות
Direction dirImg = Direction.RIGHT; - עצם המייצג את כיוון התמונה של האיבר הראשון

פעולות:
-פעולות לבדיקת פסילות
-פעולה למיקום של האוכל
-פעולה להגדלת שורת הכלבים
-פעולה לבדיקת תקינות מיקום האוכל
-פעולות שמשנה את כיוון התמונה של הכלב הראשון
-פעולות המחזירות TextLabel
פעולה המאתחלת את המערך של הכלבים

שלב התכנות- שינויים:
- החלטנו לשנות את העיצוב של כלב אחד אחריו רודף כלב רע שמנסה לתפוס אותו וככל שהוא אוכל, יותר כלבים רודפים אחריו- כדי שלא יהיו סיבוכים עם הרגליים של הכלב.
החלטנו שהאוכל יהיה תה, והבונוס פוד יהיה ביסקוויט, אוטובוס, וכתר המלכה (מופיעים ברנדומליות כל פעם שיש סופר פוד, עם הודעת סופר פוד מותאמת לכל אחד) כדי להתאים לקונספט הבריטי שעליו החלטנו בהמשך.
- שינינו את ההיווצרות של האוכל בתחילת המשחק למקום קבוע כדי שיהיה אסתטי, ורק מהשני הוא יתחיל להיווצר במקום רנדומלי.
- בבדיקה של הcounter שסופר את האוכל הרגיל (עד הבונוס פוד) הוספנו לif עוד תנאי, counter!=0.

באגים:
התמונות מופיעות עם רקע לבן ולא שקוף.
כשלוחצים על שני מקשים בו זמנית לפעמים זה גורם לפסילה.

רפלקציה והערכה:
לדעתנו אופן הלמידה והתכנון של הפלטפורמה היה טוב, מכיוון שחקרנו את המחלקות באופן יסודי וכתבנו את התכנון במפורט.
הקוד קריא, בזכות שימוש בפעולות ובקבועים עם שמות משמעותיים.
בנוסף איכות הפתרון טובה (מלבד הבאג שציינו של המקשים), לאור הזמן הרב שהשקענו בתכנון שלה, והמראה של המשחק אטרקטיבי, חוץ מהבאג של הרקע הלבן.
הרגשנו ששיתוף הפעולה היה טוב ומפרה. שיטת העבודה הייתה שחילקנו חלקים מסוימים לאנשים מסוימים, ואם נתקלנו בקושי והיינו צריכים עזרה תקשרנו ועזרנו אחד לשני דרך וואטסאפ. בנוסף, על כל עדכון שערכנו בקוד עדכנו בוווטסאפ.

אופן הלמידה והתכנון של הפלטפורמה: 100
כתיבת קוד קריא ומתועד, שימוש נכון בפעולות ובמחלקות: 100
איכות הפתרון: 98
מראה כללי, יצירתיות ואטרקטיביות המשחק: 98
אופן העבודה הקבוצתית: 100
ציון סופי: 99