**מבוא לתכנות מונחה עצמים- פרויקט גמר**

**מגישים- בנימין רוסין (211426598), יוחאי בניטה (322636036)**

בפרויקט זה נדרשנו לבנות את המשחק חתול ועכבר ע"י שימוש בSFML ובירושה ופולימורפיזם.

התוכנית תקרא קבצי שלב מהמחשב ולאחר מכן תריץ את המשחק.

המטרה של השחקן (העכבר) היא לאסוף את כל הגבינה בכל שלב, מבלי להיתפס על ידי החתולים.

השלב מסתיים בהצלחה כאשר נגמרות כל פיסות הגבינה שעל המסך, והמשחק ממשיך לשלב הבא.

המשחק ממשיך כל עוד יש שלבים נוספים ונשארו לשחקן "חיים" וזמן. בתחילת המשחק יש לעכבר שלוש חיים, אם הוא נתפס ע"י החתול, השחקן נפסל והשלב יתחיל מחדש במידה ונשארו חיים.

המשחק יסתיים בהצלחה בסיום השלב האחרון, או בהפסד אחרי שלוש פסילות.

עבור כל שלב יוקצה זמן שונה.

**סימונים בלוח:**

|  |  |
| --- | --- |
|  | # - **קיר**. אי אפשר לעבור דרכו כלל. |
|  | D – **דלת**. רק עכבר יכול לפתוח אותה במידה ויש לו מפתח, עבור החתול-דלת שלא נפתחה נחשבת כקיר. |
|  | F – **מפתח**. רק עכבר יכול לאסוף אותו, חתול יכול "לדרוך" עליו אבל לא לקחת אותו. |
|  | \* - **גבינה**. רק עכבר יכול לאסוף אותה, חתול יכול "לדרוך" עליה אבל לא לקחת אותה. |
|  | ^ - **חתול**. |
|  | % - **עכבר**. יכול להיות רק אחד במשחק. |
| $ - **מתנות:** רק עכבר יכול לאסוף מתנות. המתנות מוגדרות באופן הבא: | |
|  | - **מתנה רגילה**. כאשר עכבר יאכל אותה, חתול אחד "ימות", חתול יכול "לדרוך" עליה אבל לא לקחת אותה. |
|  | - **תוספת חיים:** כאשר עכבר יאכל אותה, יתווספו לו חיים. |
|  | - **הקפאת חתולים:** כאשר העכבר יאכל אותה, החתולים לא יוכלו לזוז למשך 5 שניות. |
|  | - **תוספת זמן:** כאשר העכבר יאכל אותה, הזמן לביצוע השלב יתארך 5 שניות. |

**ניקוד:**

גבינה – 10 נקודות.

מתנה – 5 נקודות.

פתיחת דלת – 2 נקודות.

סיום שלב בהצלחה – 25 נקודות + 5 נקודות על כל חתול שהיה בשלב.

**קבצים שנוצרו:**

**Mouse.h**

**Mouse.cpp**

קבצים אלה מכילים את מחלקת העכבר.

מחלקת העכבר מכילה את וקטור הכיוון של העכבר, הגדרת ההתנגשויות עם אובייקטים שונים וכן צליל סאונד לפסילה. בנוסף היא מכילה את פונקציית התזוזה של העכבר.

**Cat.h**

**Cat.cpp**

קבצים אלה מכילים את מחלקת החתול.

מחלקת החתול מכילה מכילה את וקטור הכיוון של החתול והגדרת ההתנגשויות עם אובייקטים שונים. כל תנועות החתול מבוצעות ע"י האלגוריתם שנמצא במחלקה זו.

**Level.h**

**Level.cpp**

קבצים אלה מכילים את מחלקת השלב.

מחלקה זו מכילה את כל הפונקציות והמשתנים שיש בהם שימוש בזמן השלב הנוכחי.

מחלקה זו מעדכנת את הניקוד בכל ריצת המשחק.

**Controller.h**

**Controller.cpp**

קבצים אלה מכילים את המחלקה בה מתנהל המשחק.

מחלקה זו מכילה את כל הפונקציות והמשתנים שיש בהם שימוש בזמן ריצת התכנית עד אשר השחקן יפסל או יחליט לסיים את המשחק או שיגמרו השלבים.

מחלקה זו מעדכנת את הלוח ואת נתוני השחקנים בכל מהלך בשלב.

**Cheese.h**

**Cheese.cpp**

קבצים אלה מכילים את מחלקת הגבינה.

במחלקה מוחזק משתנה סטטי ובו מספר הגבינות במשחק וכן double dispatch למימוש התנגשויות.

**Door.h**

**Door.cpp**

קבצים אלה מכילים את מחלקת הדלת.

במחלקה קייםdouble dispatch למימוש התנגשויות עם דלת.

**Gift.h**

**Gift.cpp**

קבצים אלה מכילים את מחלקת המתנה.

במחלקה קייםdouble dispatch למימוש התנגשויות עם מתנה.

בנוסף המחלקה מחזיקה פונקציה וירטואלית לטובת מימוש המתנות השונות.

**Gfreeze.h**

**Gfreeze.cpp**

**Gkill.h**

**Gkill.cpp**

**Glife.h**

**Glife.cpp**

**Gtime.h**

**Gtime.cpp**

**Key.h**

**Key.cpp**

**Wall.h**

**Wall.cpp**

**Menu.h**

**Menu.cpp**

**Object.h**

**Object.cpp**

**Moving\_object.h**

**Moving\_object.cpp**

**Static\_object.h**

**Static \_object.cpp**

**Resources.h**

**Resources.cpp**

**מבני נתונים עיקריים ותפקודיהם:**

1. וקטור של unique\_ptr בשם Moving\_object שמכיל בכל תא מצביע לאובייקט של חתול ובתא הראשון מצביע לאובייקט של עכבר.
2. וקטור של unique\_ptr בשם Static\_object שמכיל בכל תא מצביע לאובייקט סטטי שנמצא על לוח המשחק.
3. מחלקת **Resources** היא מסוג סינגגלטון והיא מכילה את הקבצים החיצוניים הנדרשים לריצת התכנית (תמונות, צלילים וכו').
4. וקטור של מחרוזות שמכיל את שמות השלבים.

**תיכון (:(design**

קלט צפוי- בעת קריאת הקלט נקרא את שמות השלבים מקובץ 'playlist.txt', כל קובץ יופיע בתקיה resources במבנה של Level\*\*\*.txt בכל קובץ יצוייר השלב כפי שמצופה שנציג אותו, אין לצייר פחות מגבינה אחת בשלב וחייב להיות בדיוק עכבר אחד.

מחלקת הGameController מנהלת את המשחק, פונקציית הplay רצה בלולאה כל עוד ישנם חיים ומס' השלב קטן מכמות השלבים הנתונה. בכל ריצה של הלולאה נקצה אובייקט Level ונקרא לתוכו את נתוני השלב, במידה והשלב שקראנו אינו תקין (אין גבינה או מס' העכברים שונה מאחד) נעבור לקריאת השלב הבא.

בכל שלב נרוץ בלולאה ונבצע תנועה של העכבר ע"י קלט משתמש, נוודא שהקלט תקין ולאחר מכן נבצע תנועות בכל אחד מן החתולים. תנועת החתול הינה לפי אלגוריתם שנכתב, בכדי לרכך את האלגוריתם הוספנו פונקציית rand(). כל תנועה של חיה מבוצעת בתוך המחלקה הרלוונטית ומחזירה ומצביע למיקום החדש, מחלקת הLevel אחראית על הדפסה מחדש ועדכון הנתונים.

המשחק ירוץ עד סיום השלבים או סיום החיים, בסיום המשחק תודפס הודעה מתאימה.

**אלגוריתמים הראויים לציון:**

תנועת החתול מבוצעת ע"י המחשב.

האלגוריתם ישווה בין הקואורדינטות של החתול והעכבר ויצמצם טווח אל עבר העכבר לפי ההפרש הגדול מבין הX-ים והY-ים.

כלומר, אם ההפרש של ה-X-ים גדול מההפרש של ה-Y-ים, אזי נדאג לקרב את החתול אל העכבר בציר ה-X. באותו אופן יבוצע הפוך.

על מנת שהאלגוריתם לא יהיה "אגרסיבי", חלק מן התנועות של החתול מבוצעות באופן רנדומלי.

**באגים ידועים:**

**הערות:** הוגש במהלך שירות מילואים.