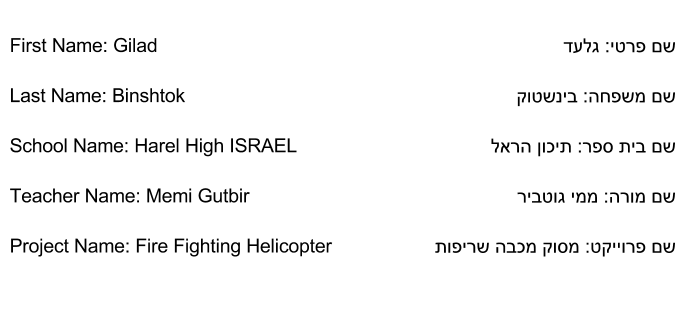


JAVA

עבודת גמר במערכות גרפיות-תשע"ג 2013

**CG Project**



תוכן עניינים

מבוא …....................................................................................................עמ' 3

הגדרת העבודה.......................................................................................עמ' 4-8

מדריך למשתמש....................................................................................עמ' 9-12

מדריך למפתח.....................................................................................עמ' 13-21

רקע תאורטי........................................................................................עמ' 22-24

בעיות שנשארו פתוחות והצעות לשיפורים......................................................עמ' 25

סיכום אישי...............................................................................................עמ' 26

תדפיס תוכנה......................................................................................עמ' 27-50

מבוא

מטרות העבודה:

מטרת העבודה שלי היא בניית משחק שבו המשתמש יכבה שריפה. כיבוי השריפה יבוצע באמצעות מסוק שאליו מחובר דלי אשר אותו אפשר יהיה למלא במים ולרוקנו מעל האש וכך לכבותה.

המסוק יהיה בנוי מגוף מסוק, פרופלור ודלי שמתמלא ומתרוקן. הדלי יתמלא כאשר הוא יוכנס לים המקיף את האי הבוער. הדלי יתמלא בהדרגה בקצב איתי ולא מיד. הדלי יתרוקן רק כאשר יש בו מים והוא מעל גובה מי הים. גם ריקון הדלי יבוצע לאט ובהדרגתיות וילווה במטר טיפות מים אשר יפלו על האש ויכבו אותה. הדלי ימטיר טיפות כל זמן שהוא מתרוקן.

בזמן כיבוי האש, להבות האש ידעכו לאט עד לכיבוי של הלהבה. כל עוד הלהבה דולקת היא תוכל להתפשט לכל צדדיה בתנאי שהם בגבולות היבשה ואינם כבר דולקים.

הרקע למשימה:

את הרעיון למשימה נתן לי המורה שלי אברהם (ממי) גוטביר אשר הציע לי את הרעיון כאשר היה לי מחסום יצירתי. מיד כששמעתי את הרעיון התלהבתי וישר עלו לי רעיונות כיצד לבצע את המשימה.

הגדרת העבודה

מימוש גרפי:

טרנסלציה לפי וקטור בין שתי נקודות

טרנסלציה לפי וקטור בין שתי נקודות מתבצעת על ידי מיצוי הווקטור משתי נקודות כלשהן והזזת הגופים לפי וקטור זה. זוהי פעולה פשוטה שלא דורשת זמן רב לתכנות אך היא מתבררת כיעילה במקרים רבים.

השתמשתי בשיטה זו בפרויקט כאשר בכל עצמי המשחק(מלבד האש) מתקבלת נקודת מרכז העצם, כאשר בעזרת טרנסלציה בין מרכז הגוף למרכז שנשלח אפשר להזיז את הגוף כך שמרכזו יהיה במרכז שנבחר.

**יצירת מסוק והצבתו במקום שנקבע**

בפעולה הבונה של המסוק, הוא מקבל בין השאר את נקודת המרכז של גוף המסוק. גוף המסוק מיוצג על ידי גוף מסוג "JavaBall" אשר מוגדל לגודל הרצוי ומוזז כך שמרכזו יהיה המרכז הרצוי כפי שתואר מקודם. בנוסף לגוף המסוק ישנם גם צריח שעליו מונח המדחף. שניהם מונחים מעל המסוק גם לפי אותה שיטה שציינתי. בנוסף מוצבים גם דלי ומים מתחת למסוק וביניהם למסוק מצויר קו בעזרת פעולה שממי יצר במחלקת השירות.

**מים בתוך הדלי**

"בתוך" הגוף שמייצג את הדלי נוצר עוד גוף שמייצג את המים. גוף זה נוצר בהתחלה כהעתק של הדלי. לאחר מכן מוגדר הצבע של המים תכלת והם משוטחים על ידי הקטנת ערך ה Y של הקיר העליון של המים. כדי ליצור אשליה של מילוי וריקון של הדלי מגדילים ומקטינים את ערך ה Y של הקיר העליון של המים.

מימוש בתכנות כללי:

דלי אוסף מים

הדלי מיוצג על ידי גוף דמוי גליל שקוף. כאשר הדלי יורד מתחת למים הוא מתחיל להתמלא בקצב איתי וקבוע. פעולת המילוי מתבצעת כאשר הדלי יורד מתחת לתיבה שמייצגת את היבשה. בתוך הדלי קיים עוד גוף בצורת גליל כאשר עם בניית המסוק הוא מכווץ ולא מצויר. עם התחלת מילוי הדלי ערך הY של הגוף המייצג את המים קטן כך שהמים גבהים. על מנת לקבוע אם צריך או לא לצייר את המים, יצרתי משתנה בוליאני בשם empty אשר בודק האם הדלי ריק ממים ובמידה שלא רק אז מציירים את הדלי.

ריקון של דלי

בדומה למילוי הדלי כך גם הריקון מתבצע על ידי שינוי ערך ה Y של הקיר העליון של הדלי כאשר מגדילים אותו וכך מכווצים את הדלי עד שהוא מתקבץ מספיק ואז מעודכן הערך empty ומפסיקים לצייר את המים. ריקון הדלי מתבצע כל זמן שיש מים בדלי והוא מעל גובה היבשה. כל זמן שהדלי מתרוקן יורדות טיפות מהדלי לפי אלגוריתם ירידת טיפות מיוחד שיצרתי.

התפשטות אש

על מנת לאפשר התפשטות של האש בחרתי ליצור את כל להבות האש בתוך מערך דו-ממדי. בנוסף למערך הלהבות הוספתי מערך של ערכים בוליאניים שבודק האם הלהבה דולקת באותו הזמן. התפשטות האש מתבצעת באמצעות הגרלת מספר רנדומלי לכל להבה דולקת. לפי ערך המספר הרנדומלי יודעים האם הלהבה צריכה להתפשט מסביב ולאן. הלהבה לא תתפשט למקום שהוגדר אם המקום כבר בוער או אם הוא מחוץ לגבולות המערך. הבערת אש במקום חדש מתבצעת באמצעות עדכון מקום הלהבה במערך הבוליאני לאמת וכך שאר התוכנה יודע לכבות לצייר ולגרום לה להתפשט בהתאם.

אלגוריתמיים מקוריים:

הסתרה בין מסוק ליבשה לאש

כאשר מורידים את הדלי מתחת למים על מנת למלאותו. במצב זה צריך לצייר קודם את המסוק ורק לאחר מכן את היבשה כך שהיבשה תסתיר את הדלי.

בנוסף ישנה גם הבעיה של הסתרה בין האש הבוערת לטיפות המים. על מנת לפתור בעיה זו יצרתי פעולה במחלקה אש אשר מקבלת ערך Z ומציירת מאחוריו ועוד אחת שמציירת מלפניו. כך אפשר לצייר את האש בחלקים ולפתור את בעיית ההסתרה. ישנה עוד פעולה שפועלת כאשר יש שני מסוקים אשר מקבלת שני ערכי Z ומציירת ביניהם.

הסתרה בין שני מסוקים

כאשר המשחק פועל עם שני מסוקים ישנה בעיית הסתרה בין שני המסוקים במקרים רבים. אחת הבעיות היא כאשר אחד ההליקופטרים מסתיר את השני תוך כדי שהוא מוסתר על ידי היבשה כאשר הוא ממלא מים. על מנת לפתור בעיה זו יצרתי אלגוריתם שמצייר את המסוק בשני חלקים. כך אפשר לצייר את החלק העליון של ההליקופטרים ממוין לפי ציר Z ואת החלק התחתון ממוין לפי ציר Y. בנוסף גם דאגתי גם להסתרה בין המסוקים ליבשה ולאש כפי שתיארתי מקודם.

הצבת טיפות בתוך מעגל ניצב לציר Y

הפעולה שבונה טיפות מקבלת את מרכז הטיפות, רדיוס המעגל בו נמצאות הטיפות וכמות הטיפות. בשביל להציב את הטיפות בתוך המעגל יצרתי אלגוריתם שמציב את הטיפה במרכז המעגל מזיזה אותה על ציר ה X בתווך רנדומלי בין 0 לרדיוס ומסובבת אותה סביב ציר Y מהמרכז בזווית רנדומלית בין 0 ל 360 מעלות.

המטרת טיפות

על מנת לקבוע באיזו מהירות תרד כל טיפה כך שתיווצר אשליה של זרימת מים יצרתי מערך זהה לזה של הטיפות המכיל מספר רנדומלי בין המהירות שניתנה לטיפות לבין חצי ממהירות זו. כך לכל טיפה שמורה המהירות שלה וכך היא נעה במהירות קבוע עד שהיא מגיעה לתחתית.

דעיכת אש

על מנת שאוכל להפוך את המשחק למציאותי ומהנה יותר החלטתי שהאש תכבה בדעיכה לאט ולא מיד. על מנת שהאש תדעך יצרתי אלגוריתם שבודק האם אחת מן הטיפות נוגעת באש אם היא נוגעת האש מופעלת במחלקה אש פעולה שגורמת לאש לדעוך לאט לאט. דעיכת האש מתבצעת על ידי הורדת ערך הצבע האדום בתוך האש. כך גוון האש נהיה יותר ויותר כהה כל פעם עד אשר ערך האדום נמוך מספיק ואז האש נכבת על ידי עדכון הערך הבוליאני של האש.

**אלגוריתמים שלמדנו בכיתה:**

מבט מלמעלה

מבט מלמעלה הינו קלי עזר למשתמש אשר נותן לו אפשרות לראות את הפרויקט מזווית שונה שמאפשרת לו להבין יותר טוב מה נמצא איפה. על מנת לאפשר מבט מלמעלה יוצרים במחלקה View עוד פאנל צדדי שבו יצויר המבט מלמעלה.

הפעולה ב Body3D אשר מציירת מבט מלמעלה של הגוף מקבלת את ה Graphics של הפאנל מבט מלמעלה שתי נקודות שמייצגות את הציר (מקביל לציר X) סביבו מסובב הגוף ואת הגדלים של הפאנל הראשי ושל מבט מלמעלה.

הפעולה יוצרת גוף חדש על ידי העתקה של הגוף המקורי ומקווצת אותו כך שיתאים לפאנל הקטן ומסובבת אותו ב-90. לאחר מכן הפעולה מציירת את הגוף ללא פרספקטיבה וכך מצויר גוף שנראה כמו מבט מלמעלה של הגוף המקורי.

חישוב מרחק בין שתי נקודות

חישוב מרחק בין שתי נקודות מתבצע על ידי חישוב אורך הווקטור העובר בין שתי הנקודות האלה. חישוב אורכו של וקטור מתבצע על ידי נוסחת פיתגורס במרחב שלפיה אורך הווקטור אלגברי שווה לשורש סכום ריבועי ערכיו של הווקטור.

חישוב זה מתבצע בפרויקט כאשר יש משחק לשני משתתפים וצריך לבדוק האם ההליקופטרים מתנגשים. מחשבים לפי הנוסחה הנ"ל את המרחק בין שני המסוקים ואם הוא קטן מאורך המדחף סימן שהמסוקים מתנגשים אחד בשני. השתמשתי בחישוב זה ולא בפעולה intersect ב Body3D משום שהמדחפים מסתובבים באופן קבוע מה שעלול לגרום למצב שבו התנגשות המסוקים תתגלה מאוחר מידי ויתקע המשחק.

**"נרמול" וקטור אלגברי**

נרמול וקטור היא שיטה של הקטנת הווקטור כך שאורכו יהיה אחד. ההקטנה מתבצעת באמצעות חילוק כל ערכי הווקטור על הצירים באורך הווקטור.

חישוב אורך הווקטור מתבצע כך: l=√x2+y2+z2

אלמנטים גרפיים:

מסך פתיחה

מסך הפתיחה Menu הוא חלון יורש מ-Jframe אשר מתפקד כתפריט הראשי שנפתח כאשר מפעילים את הפרויקט. במסך הפתיחה יש פירוט מלא של כל מקשי המשחק וגם שדות טקסט שבהם אפשר לכתוב את שמות השחקנים. בתחתית המסך כפתור START שלחיצה עליו פותחת את חלון ה View שבו מופיע המשחק.

פאנל למבט מלמעלה

פאנל אשר בו מצויר מבט מלמעלה של כל הפרויקט בעזרת אלגוריתם ציור מלמעלה של כל גוף כפי שציינתי מקודם.

הוספת השמות מתבצעת באמצעות שני שדות טקסט אשר בהם ניתן לכתוב את שמות השחקנים. את השמות שולח מסך הפתיחה לוויו שמעדכן אותם במודל. השמות מופיעים מעל ההליקופטרים כך שכל שחקן ידע איזה הליקופטר הוא ההליקופטר שלו.

תפריט צדדי ב**View**

* כפתורי רדיו **-** 5 כפתורים שאיתם בוחרים את רמת המשחק ושני כפתורים שקובעים האם המשחק ישוחק עם שחקן אחד
* כפתור רגילים **-** כפתור START שמתחיל משחק חדש ומעדכן את כל הפרטים שעודכנו בתפריט הצדדי למודל.
* **Color Chooser -** שלושה כפתורים שמעלים תפריט לבחירת צבע. פתורים אלה קובעים את הצבעים של שני ההליקופטרים ויבשת המשחק.

מדריך למשתמש

התוכנה שפיתחתי היא משחק של הליקופטרים שמכבים שריפות. המשחק מתרחש על יבשה שנמצאת בלב ים. מטרת המשחק היא לאסוף מים מהים ולכבות בעזרתם את האש שתופיע במספר מוקדים על היבשה. על מנת לנצח צריך לכבות את כל האש שעל המשטח. אפשר לבחור אם לשחק ביחידים או בזוגות.

הוראות הפעלה:

על מנת למלא את הדלי במים יצטרך השחקן להכניס את הדלי לים שמקיף את היבשה שעליה מתרחשת השריפה. על מנת לכבות את האש יש להעביר את הדלי כאשר הוא מלא במים מעל האש והמים יצאו מהדלי בצורת טיפות וכאשר ינחתו על האש, האש תדעך עד כדי כיבוי.

מצב שחקן יחיד:

|  |  |
| --- | --- |
| **מקש** | **פקודה שאותה הוא מבצע** |
|  | מזיז את המסוק ימינה |
|  | מזיז את המסוק שמאלה |
|  | מזיז את המסוק אחורה |
|  | מזיז את המסוק קדימה |
| 1 | מוריד את מחל המים של המסוק |
| 2 | מעלה את מחל המים של המסוק |

מצב שני שחקנים

במצב זה יתאפשר לשני שחקנים לשחק בשני מסוקים שונים. שני השחקנים ישחקו אחד עם השני ויפעלו ביחד על מנת לכבות את האש. שני המסוקים לא יוכלו להתנגש אחד בשני, על מנת להימנע מתאונות לכן מומלץ לנסות להימנע ממצב זה.

**שחקן מספר 1**

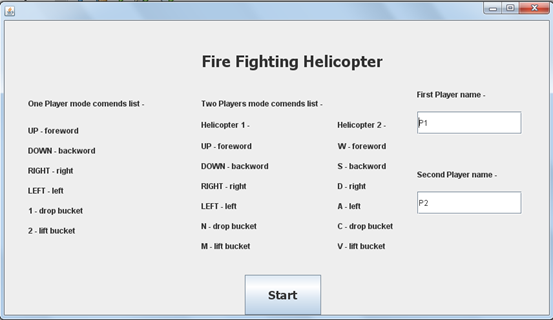
|  |  |
| --- | --- |
| **מקש** | **פקודה שאותה הוא מבצע** |
|  | מזיז את המסוק ימינה |
|  | מזיז את המסוק שמאלה |
|  | מזיז את המסוק אחורה |
|  | מזיז את המסוק קדימה |
| N | מוריד את מחל המים של המסוק |
| M | מעלה את מחל המים של המסוק |

**שחקן מספר 2**

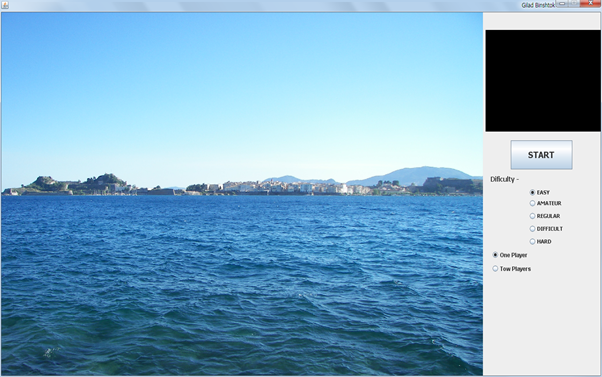
|  |  |
| --- | --- |
| **מקש** | **פקודה שאותה הוא מבצע** |
| D | מזיז את ההליקופטר ימינה |
| A | מזיז את ההליקופטר שמאלה |
| W | מזיז את ההליקופטר אחורה |
| S | מזיז את ההליקופטר קדימה |
| 1 | מוריד את מחל המים של ההליקופטר |
| 2 | מעלה את מחל המים של ההליקופטר |

מסך פתיחה

מסך זה נפתח בתחילת המשחק והוא מאפשר לך להכניס את שמות השחקנים היכן שכתוב: First Player name , Second Player name. על המסך גם מופיע הכפתור START שלחיצה עליו תעביר אותך למשחק עצמו. במסך גם מופיעה רשימה מלאה של הפקודות באנגלית.



תחילת המשחק:



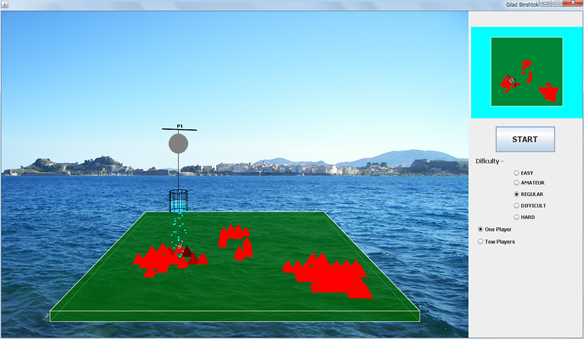
**התפריט הימיני**

זהו המסך שיופיע לכם כאשר תפעילו לראשונה את המשחק. במסך זה תהיו רשאים לבחור אחת מבין חמשת הרמות של המשחק ונקבעת לפי כמות מוקדי האש ההתחלתית כאשר ברמה הקלה יש מוקד 1 וכל רמה נוסף עוד מוקד. בנוסף אפשר לקבוע גם אם לשחק שחקן אחד על ידי סימון המקום One player או שני שחקנים על ידי סימון המקום Two Players. התחלת המשחק תתבצע על ידי לחיצה על הכפתור START שתתחיל את המשחק לפי הנתונים שהוגדרו למקודם.

מצד ימין למעלה אפשר לראות ריבוע שחור ריק. בהמשך המשחק בריבוע זה יופיע מבט מלמעלה על המשחק. המבט מלמעלה הוא כלי עזר שמאפשר למשתמשים ליראות היכן הם נמצאים מבחינת אורך ורוחב מנקודת מבט יותר ברורה ומדויקת.

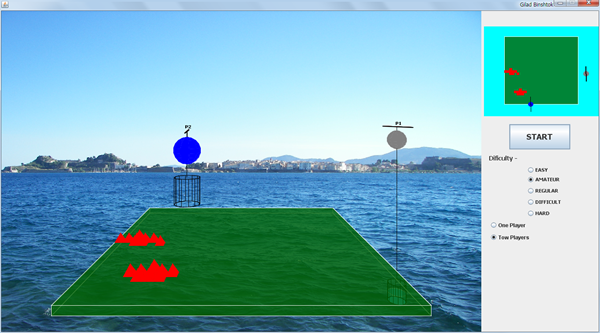
בצד השמאלי של המסך מופיע תמונה של חוף ים. לאחר הלחיצה על הכפתור START יופיע על תמונה זו יופיע המשחק.

מצב שחקן יחיד



כפי שאפשר לראות המסוק מיוצג על ידי גוף המסוק בצורת כדור, מדחף, דלי בצואת גליל, חוט שמחבר בין הדלי לגוף המסוק. אפשר גם לראות את הדלי במצב חצי מלא וטיפות יוצאות מין הדלי ומכבות את האש

**מצב שני שחקנים**

אפשר לראות במצב זה עוד שחקן שנוסף למשחק וגם אפשר לראות את הדלי מתמלא ואת שמות השחקנים מעל המסוקים.

.

מדריך למפתח

עקרונות אנימציה עם **Model View**:

במחלקה View רצה לאורך כל התוכנית לולאה אין סופית שמעדכנת ומציירת את תוכן הפרויקט באמצעות הפעולות draw ו update במחלקה Model. מודל היא מחלקה שמוחלת בתוך המחלקה וויו. המודל אחראי על ביצוע העדכונים הנדרשים לביצוע דוגמת סיבוב פרופלור ההליקופטר והתפשטות האש במרחב (למען תכנות נוח יצרתי מחלקה Game שדרכה מיוצרים, מעודכנים ומצוירים עצמי המשחק. מחלקה זו מוחלת במודל ומעודכנת במודל.) בתוך המחלקה מודל יש עצם מטיפוס טיימר שבודק כמה זמן עבר וקובע מתי הזמן הנדרש לבצע את העדכונים הנדרשים במודל.

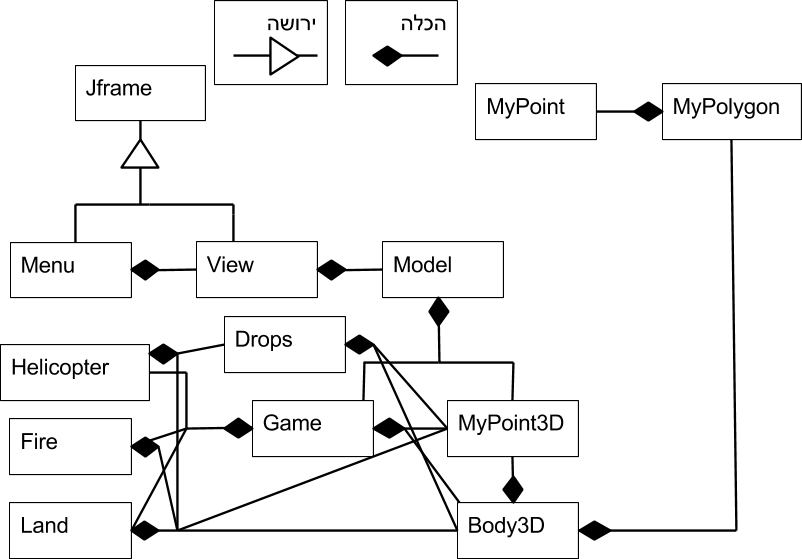
**View**

הוויו היא מחלקה שיורשת מהמחלקה Jframe שמוגדרת בjava. המחלקה מייצגת את החלון שבו רואים את התמונה של כל הפרויקט. המחלקה קולטת את כל הפעולות שמבצע המשתמש (כגון: לחיצה על מקש ימינה לחיצה על העכבר ולחיצות על כפתורים שונים) ומעבירה את הפקודה המתאימה למודל. הוויו מכיל פאנליי ציור בשמות pnlCanvas שבו מצוירים כל חלקי המשחק ועליו אפשר לראות את כל המשחק. בנוסף הוספתי פאנל בשם pnlTop שמראה מבט מלמעלה של הפרויקט לטעמי נוחות. בוויו מוגדר גם משתנה מסוג Buffer Image שבו מעודכנת תמונת המשחק בנפרד מהפאנל על מנת למנוע הבהובים על המסך

**Model**

המודל הי המחלקה החראית על עדכון כל הפעולות הנדרשות בפרויקט. המודל מכילה מערך בוליאני של כל מקשי המקלדת. בעזרת מערך זה אפשר לבדוק בכל זמן שהטיימר קובע שצריך לעדכן את הפרויקט האם המקש לחוץ ולקבוע פעולה מתאימה למקרה. בנוסף המודל מכילה משתנה מסוג טיימר שקובע מתי צריך לעדכן את התכנה. עדכון התוכנה מתבצע על ידי הפעולה update שמבצעת את כל הפעולות שהמתכנת מגדיר. עדכון הפרויקט מצריך ציור מחדש של הפרויקט המבוצע על ידי הפעולה draw, בדיקת הכפתורים הלחוצים ופעולות מתאימות ועדכונים שמבוצעים באופן אוטומטי אם לא נאמר אחרת.

דף **UML**:



קבצי תוכנה:

מחלקות שלמדנו בכיתה:

**MyPoint**

מחלקה המייצגת נקודה דו ממדית. הנקודה מורכבת משני ערכי double שמייצגים את מיקום הנקודה על הצירים X ו Y .

**My Polygon**

מחלקה המייצגת מצולע דו-ממדי. הפוליגון מיוצג על ידי מערך של נקודות שמייצגות את מיקום קדקודיו של הפוליגון. את הפוליגון ניתן להזיז, לסובב ולקבל את ערכי ה X וה Y המקסימליים המינימליים ושל נקודת המרכז שלו.

**MyPoint3D**

מחלקה המייצגת עצם מסוג נקודה תלת-ממדית. הנקודה מיוצגת על ידי שלושה ערכי X Y וZ מסוג double שמייצגים את מיקומה של הנקודה על שלושת הצירים. את הנקודה אפשר ליצור על ידי שליחת הערכים של הנקודה בקונסטרקטור או על ידי העתקת נקודה אחרת בעזרת קופי-קונסטרקטור. בנוסף אפשר לגשת ולשנות את ערכי הנקודה בכל רגע נתון.

**Body3D**

מחלקה המייצגת עצם מסוג גוף תלת-ממדי. הגוף מיוצג על ידי מערך חד-ממדי של כל נקודות תלת-ממדיות אשר מייצגות את קודקודי הגוף במרחב וגם מערך דו-ממדי של מספרים שלמים שמייצג את קירות הגוף על ידי ציון הקדקודים של כל קיר. בנוסף לגוף ישנם תכונות נוספות כגון: צבע הקירות, צבע מסגרת הקירות, נקודת מבט שממנה מקבלים פרספקטיבה, נקודת אור שממנה מקבלים צל "סטייל" הגוף שמציין כיצד לצייר את הגוף וגודל הפאנל עליו יצויר הגוף.

את הגוף אפשר להזיז, לסובב על ידי שליחת ציר הסיבוב על ידי שתי נקודות ומספר המייצג את זווית הסיבוב במעלות, להקטין ולהגדיל על ידי סילום שינוי וקבלת ערכי מיקום ועוד.

**Timer**

מחלקה המיוצגת על ידי שני ערכי long שמייצגים את הזמן האחרון שבו עודכנה התוכנה באלפיות שנייה וכל כמה זמן צריך לעדכן את התוכנה(במקרה הזה כל 10 אלפיות או מאייה). המחלקה מחזירה האם הגיע הזמן לעדכן את התוכנה.

מחלקות שיצרתי:

**Game**

מחלקת המשחק. מחלקה אשר מכילה את כל העצמים שמופיעים במשחק כגון: הליקופטר, יבשה ואש. במחלקה זו מתממשים כל המפגשים בין עצמי המשחק לדוגמא: טיפות שנוגעות באש וגורמות לה לדעוך, מניעת התנגשות בין שני מסוקים וכו'. בנוסף מחלקת המשחק גם אחראית לעדכון המשחק כל פעם שצריך בעזרת הפעולה doWhatYouHaveToDo אשר קוראת לפעולה הזהה אצל העצמים הפועלים במשחק ובנוסף גם בודקת האם צריך לכבות את האש וקוראת לפעולת הכיבוי במידת הצורך.

המחלקה גם אחראית על ציור המשחק תוך כדי התחשבות בהסתרת גופים בעזרת אלגוריתם ציור אשר יפורט בהמשך.

למרות שאפשר לבצע את כל תפקידי מחלקה זו במודל בחרתי ליצור מחלקה זו מטעמי נוחות וסדר. הוספת מחלקה זו הופכת את התכנות לפשוט יותר בכך שהיא יוצרת שתי מחלקות קצרות ונוחות לקריאה ועריכה במקום מחלקה אחת מסורבלת.

**Land**

מחלקה אשר מייצגת את היבשה שעליה בוערת האש. המחלקה מכילה גוף בצורת תיבה שטוחה אשר מייצגת את היבשה עצמה. את הגוף המייצג את היבשה מכילות מרבית מחלקות הפרויקט על מנת שאפשר יהיה לבדוק את מיקום העצמים בייחס אליו. בחרתי לייצג את היבשה במחלקה ולא בגוף בתוך מחלקת המשחק על מנת שאפשר יהיה להוסיף אלמנטים ומכשולים על היבשה.

**Drops**

מחלקה המייצגת טיפות מים. טיפות המים מיוצגות על ידי מערך של גופים קטנים בצבע תכלת. בנוסף למערך הגופים קיימים גם מערך של מהירויות ירידת הטיפות מרכז הטיפות ורדיוס הדלי בו נמצאות הטיפות(למעגל זה אפשר לקרוא מעגל הטיפות). ירידת הטיפות מתממשת באמצעות טרנסלציה לכיוון מטה במהירות רנדומלית שנקבעת על ידי המתכנת, כאשר הטיפות מגיעות לרצפה או נוגעות באש הן מיד מתורגמות לראש הדלי. בנוסף אפשר לקבוע את כמות הטיפות בקונסטרקטור.

**Fire**

מחלקה המייצגת את האש הבוערת. האש מיוצגת על ידי מערך דו ממדי של להבות שמיוצגות על ידי גופים בצורת פירמידה בגודל שנקבע על ידי המתכנת. כל הלהבות מוגדרות כבר בקונסטרקטור על מנת למנוע האטה של הפרויקט בעת ההפעלה. קיים גם מערך דו-ממדי בוליאני שבו משתמשים כדי לקבוע איזו להבה בוערת ואיזו לא. בעזרת מערך זה יודעים איזו אש צריך לצייר, האם צריך לכבות את האש והאם האש צריכה להתפשט. בעת מגע של הלהבה עם אחת מטיפות המים הלהבה דועכת על ידי הורדת ערך האדום בצבע האש עד אשר ערך האדום של האש נמוך מספיק ואז נכבת האש.

**Helicopter**

מחלקה המייצגת את המסוק אתו מכבים את האש. המסוק מיוצג על ידי כדור בתור גוף המסוק, למסוק יש שני גופים ארוכים מאונכים אחד על השני אשר מיצגים את מדחף המסוק. המדחף מסתובב במהירות קבוע לאורך כל המשחק. למסוק יש גם דלי בצורת גליל אשר מתמלא כאשר הוא מתחת למים ומתרוקן לאט תוך כדי הורדת טיפות שהוגדרו מראש בחלקו העליון של הדלי. המים שבתוך הדלי מוגדרים על ידי גוף כחול בצורת הדלי אשר גדל וקטן כאשר הדלי מתמלא ומתרוקן.

המסוק הוא העצם היחידי במשחק שבו שולט המשתמש. המשתמש יכול להזיז את המסוק לצדדים, אחורה וקדימה באמצעות מקשי החיצים על הלוח או במקשים W A S D כאשר יש שני שחקנים. בנוסף המשתמש יוכל גם להוריד את הדלי ולהרימו וכך יוכל להוציא ולהכניס את הדלי למים ולמלאותו. להליקופטר תווך תנועה שהוא אינו יכול לחרוג ממנו זאת על מנת שההליקופטר לא יוכל להתרחק מידי משטח המשחק.

**Service**

מחלקת עזר המחילה פעולות מועילות בנושאים הנדרשים בפרויקט. המחלקה מחילה את הפעולות:

* יצירת קו לציור בפרספקטיבה
* ניקוי הפאנל
* שמירת תמונה
* מכפלה ווקטורית וסקאלרית של שני וקטורים
* נרמול וקטור לאורך של אחד
* חישוב מרחק בין שתי נקודות
* הזזת גוף לפי וקטור בין שתי נקודות

**Menu**

מחלקה אשר מייצגת חלון פתיחה שמופיע בתחילת כל הפרויקט. המחלקה יורשת מ Jframe והיא נועדה על מנת שאפשר יהיה לבחור את שמות השחקנים המשתתפים במשחק בעזרת שדות טקסט אשר אליהם מוכנסים השמות.

**קבצים בהם השתמשתי:**

**קבצי טקסט**

מבנה הגופים התלת-ממדיים נכתב בקובץ טקסט אשר בנוי ממספר כותרת שמיצג את מספר הקדקודים של הגוף. לאחר מכן רשימת כאורדינטות של אותו מספר נקודות. אחרי זה כותבים עוד מספר שמייצג את כמות הקירות מהם מורכב הגוף. הסוף הוא רשימת כל הנקודות של הקיר בשורה.

קיבלנו מהמורה מספר קבצים כגון: Cube.txt ו Cylinder.txt אשר מהם יכולנו לבנות את הפרויקט.

ל Body3D קיימת פעולה מיוחדת אשר קוראת את קובץ הטקסט ומתרגמת אותו למערכים של קדקודים ושל קירות.

מבני נתונים בהם השתמשתי:

בפרויקט זה השתמשתי במבני נתונים מסוג מערך. השתמשתי במבנה נתונים זה כאשר יצרתי את טיפות המים שנשמרו במערך חד-ממדי וגם מהירות המים נשמרה במערך מספרים באותו הגודל. השתמשתי במערך דו-ממדי על מנת לשמור את להבות האש ומערך בוליאני באותו הגודל שקובע איזו להבה דולקת.

בחרתי להשתמש במבני נתונים מסוג זה משום שהם אפשרו לי להגדיר מראש את כמות המשתנים שנשמרו בתכנית. עוד יתרון של מבנה נתונים זה ההוא שקל יותר לבנות עוד מבנה נתונים באותו גודל בעל סוג משתנים זהה וכך לפעול במקביל על שני המערכים ביותר קלות.

אלגוריתמים מרכזיים:

ירידת הטיפות והצבה מחדש

במחלקה טיפות קיים מערך של כל הטיפות. לצדו יש מערך מקביל שמכיל את המהירויות של כל הטיפות.

בזמן התרוקנות של דלי המים כל טיפה יורדת במהירות שלה וכך נוצרת אשליה של טיפות טיפות. במשך כל זמן הטיפות הטיפה במקום 0 במערך הטיפות נשארת במקומה וכך נשארת נקודת ציון של מרכז מעגל הטיפות. כאשר אחת הטיפות מגיעה לקרקע או נוגעת באש הטיפה עוברת מיד למעגל הטיפות באמצעות הפעולה placeOnePoint אשר מחזירה את הטיפה למקום רנדומלי בתוך מעגל הטיפות. לאחר שהוצבה במקומה מתחילה הטיפה לרדת שוב כל עוד נקראת הפעולה. הפעולה נקראת במחלקת המשחק כל עוד שהדלי מתרוקן עד שהוא ריק.

התפשטות אש

התפשטות האש מתאפשרת ביתר קלות בשל שמירת כל הלהבות במערך דו ממדי ושמירת ערכים בוליאניים שמייצגים האם האש באותו מקום דולקת. סידור ערכים זה מקל את התפשטות האש בכך שהוא מאפשר להדליק את להבה רק באמצעות עדכון ערך וחוסך בניית להבה חדשה.

פעולת התפשטות האש נקראת לכל להבה בפעולה spread כל עוד הערך הבוליאני שלה מראה שהיא דולקת. בעת קריאת הפעולה מוגרל מספר בין 0 ל 3999(כולל) כאשר ל12 מספרים שיוצאים האש מתפשטת לאחד מבין צדדיה ומעדכנת את מקום האש במערך burning שבודק האם האש דולקת במקום זה. הדלקת האש מתבצעת באמצעות הפעולה Bearn שמקבלת את מקום האש במערך ומדליקה אותה אם היא בתוך המערך.

ציור מים טיפות ואש לפי תנאי

לאומת ההליקופטרים והיבשה שמצוירים לאורך כל הפרויקט המים והאש מצוירים רק כאשר הם מופיעים במשחק. על מנת שעצמים אלה יצוירו רק בזמן המתאים השתמשתי בערך בוליאני שבודק את התנאי הקריטי שלפיו קובעים האם צריך לצייר או לבצע פעולות נוספות:

* למים קיים משתמש בשם empty שבודק האם הדלי ריק ואם הוא לא אזי אפשר לצייר את המים. עדכון המשתנה ב false מתבצע ברגע שמתחיל הדלי להתמלא ומעודכן ל true כאשר המים התכווצו לגודל מספיק קטן.
* לטיפות קיים משתמש בשם emptying שבודק האם הדלי מתרוקן ברגע זה אם כן אפשר לצייר את המים. עדכון המשתנה ב true מתבצע כל זמן שהדלי מתרוקן ומעודכן ב false כאשר הדלי מסיים להתרוקן וכאשר הדלי מתחיל להתמלא.
* לכל להבת אש קיים ערך בוליאני שבודק האם האש באותו מקום דולקת. על מנת להדליק ולכבות את האש מעדכנים את הערך הבוליאני וכך אפשר לדעת אם כן או לא לצייר את האש.

ריקון ומילוי של הדלי

על מנת ליצור אשליה של דלי מתמלא ומתרוקן יצרתי גוף של מים. הגוף הוא העתקה של הדלי אשר צבעו שונה לתכלת והוא קובץ לגוף שטוח באמצאות שינוי ערכי Y של הנקודות של הקיר התחתון לערכי ה Y של הכיר התחתון. בזמן שהדלי של המים מתרוקן ומתמלא כך גדל באמצעות הורדת ערך ה Y של הקיר העליות עד שהוא מגיע לערך ה Y הקטן של הדלי וקטן בדרך ההפוכה.

מילוי הדלי מתבצע רק כאשר הדלי כולו מתחת למים(ערך Y מינימלי של הדלי גדול ממקסימלי של היבשה) הוא נפסק כאשר הדלי יוצא מן המים. ריקון הדלי מתבצע כל זמן שכל הדלי מעל היבשה ויש בו מים והוא מלווה בירידה של טיפות המים.

הסתרה בין גופי המשחק

בגלל העובדה שאי אפשר לקבוע לפי ערך אחד איזה מן העצמים של המשחק צריך לצייר קודם יצרתי אלגוריתם שבודק את כל המצבים האפשריים לבעיות הסתרה ופוטר אותן. על מנת שאוכל לפתור בעיות אלה יצרתי פעולות ציור מיוחדות למחלקות הליקופטר ואש אשר אפשר לקרוא להן מתוך המחלקה משחק שם מתבצע הציור. במחלקה הליקופטר יצרתי את הפעולות הבאות:

* drawArry אשר מקבלת שני הליקופטרים ומציירת אותם לפי סדר ערכי Z
* drawHelicopter אשר מציירת את חלקו העליון של המסוק ללא הדלי
* drawBucket מציירת את הדלי וטיפות המים

במחלקה אש יצרתי את הפעולות הבאות:

* drawBefore אשר מקבלת ערך Z ומציירת את כל הלהבות בעלות ערך Z קטן יותר
* drawBetween אשר מקבלת שני ערכי Z ומציירת את כל הלהבות בעלות ערך Z גדול יותר מאחד וקטן מהשני
* drawAfter אשר מקבלת ערך Z ומציירת את כל הלהבות בעלות ערך Z גדול יותר

מצב שני שחקנים

על מנת שאפשר יהיה לשחק עם שני שחקנים יצרתי משתנה בשם second אשר מתעדכן בכל פעם שהמשחק מאותחל. לפי ערך משתנה זה המחשב יכול לדעת איזה פעולות לבצע ואיזה פעולות לא לבצע כגון ציור הליקופטר שלא קיים או בכלל ליצור אותו . העבודה עם ערך בוליאני מפשטת את התאמת הפעולות בכך שכל שצריך לבדוק הוא האם משחקים בשני שחקנים או אחד. אותו ערך קיים גם במודל ומאפשר לדעת כיצד צריך להתאים את משתני המשחק.

הגדרות משחק לפי תפריט צדדי

בחלון המשחק View קיים מעין תפריט צדדי המורכב מחמישה כפתורי רדיו לקביעת רמת הקושי של המשחק, שני כפתורי רדיו שקובעים האם משחקים עם שני שחקנים או אחד, שלושה כפתורים שמפעילים Color Chooser אשר קובעים את צבעי ההליקופטר והיבשה וכפתור START אשר מאתחל מחדש את המשחק.

כל שינוי באחד מן הכפתורים נשמר מידית בערכים בוויו וכאשר נלחץ הכפתור START מתקיים אתחול מחדש של כל ערכי המשחק במודל ומעודכנים פרטי התפריט הצדדי שעודכנו בוויו קודם לכן אל המודל אשר מעביר אותן למשחק שמעביר אותן לעצמים הרצויים.

הערות

בנוסף לדברים שצוינו בפרק זה חשוב גם לדעת כי כאשר יצרתי מבט מלמעלה של המשחק הנקודה P1 שהזנתי היא נקודת המרכז של היבשה עליה מתרחש הפרויקט. עובדה זו חשובה למפתח התכנית משום שכיווצתי את כל הגופים לכיוון נקודה זו. לכן בכל מקרה שירצו לכתב מחדש את הפקודות למבט מלמעלה ויצטרכו להתחשב בבעיה זו ולא תוכל להיווצר בעיה.

בנוסף חשוב גם לדעת שהוספתי ל BODY3D פעולת ציור בשם wireSolid אשר מציירת את כל קירות הגוף ללא קשר כיוון הווקטור הניצב להם וכך אפשר לצייר גופים מסובכים יותר שמצריכים פעולה זו.

רקע תאורטי

נושאים כלליים:

וקטורים

הווקטור הוא ייצוג מתמטי בעל אורך וכיוון. בעזרת שימוש בווקטורים הצליחו מתמטיקאים רבים להוכיח עובדות רבות בקלות רבה יותר ואפשרו פיתוח מתקדם בתחום הגיאומטריה שהוביל לפיתוחים רבים בתחום המדעים. הווקטור משמש מרכיב מרכזי בתחומי מדעים רבים כגון כימיה פיזיקה וביולוגיה.

הווקטור משמש בתחום המחשבים כאשר מתכנתים בתלת-ממד ורוצים לחשר מיקומים מדויקים של ערכים על מנת שאפשר יהיה לשפר את התוכנה.

הווקטור מחולק לשני סוגים: אלגברי שמיוצג על ידי נקודה על הצירים והווקטור הגיאומטרי שמיוצג על ידי קו עם חץ בסופו אשר מסמל את הכיוון.

משוואות פרמטריות של קו ישר:

דו-ממד

משוואת הקו הדו-ממדי:

y=mx+n

משוואה זו בנויה משני ערכים:

1. ערך M שהוא המקדם של X. ערך זה קובע את שיפוע הישר, כלומר, על כל גדילה או ירידה של ערך X ערך Y עולה או יורד M פעמים.
2. ערך N אשר מייצג את נקודת החיתוך של הישר עם ציר ה Y. כלומר, כאשר X שווה ל 0 Y שווה ל N.

תלת-ממדי

לישר התלת-ממדי אין משוואה אחת כמו לדו ממדי אלה 3 משוואות שמייצגות כל אחת את מיקום נקודה כללית על ציר אחר.

משוואת הישר:

* וקטור כיוון הישר -
* נקודה על הישר -

המשוואה מיוצגת על ידי נקודה שידוע שנמצאת על הישר ווקטור כיוון של הישר כך שאפשר להגדיר את הנקודה הכללית של הישר כך: .

נושאים בגרפיקה ממוחשבת:

פרספקטיבה :

פרספקטיבה היא שיטת ציור שמטרתה להראות איזה מן האלמנטים קרוב יותר ואיזה רחוק יותר על מנת ליצור על הדף תחושה של תלת-ממד בציור של דו ממד. על מנת ליישם שיטה זו צריך לצייר את מה שקרוב גדול יותר ומה שרחוק קטן יותר.

**יישום בתכנות גרפי בתלת ממד**

בפרויקט הפרספקטיבה מיושמת כאשר מוגדרת בגוף התלת-ממדי נקודה בשם eye אשר מייצגת את נקודת המבט של המשתמש על הפרויקט.

על מנת שאפשר יהיה ליישם פרספקטיבה בפרויקט צריך לחשב לכל נקודה היכן היא תהיה על הפאנל כאשר יש פרספקטיבה. על מנת למצוא ערך זה יוצרים קו ישר תלת-ממדי מכל נקודה לנקודת העין של הגוף. במקרה זה המיקום של הנקודה על המסך הוא הנקודה על הישר שבה ערך הZ שווה ל0.

בשביל ליישם עקרון זה מוציאים וקטור בין הנקודה שרוצים לעשות לה פרספקטיבה. כאשר מציבים את הערכים במשוואת הישר כפי שצוין מקודם מוצאים את ערך T על ידי הצבת ערך 0 ב-Z הנקודה. משתמשים באותו ערך T על מנת שאפשר יהיה למצוא את ערכי ה X וה Y של הנקודה החדשה. לאחר שמוצאים את כל הנקודות ישנו גוף דו-ממדי שאפשר לצייר על הפאנל.

הפרספקטיבה מיושמת בפרויקט שלי כאשר מוגדרת לי נקודת עין אחת לכל הפרויקט של פיה נעשית פרספקטיבה לכל מה שמצויר על הפאנל הראשי של הפרויקט.

צל:

צל של גוף מיוצג על ידי גוף שטוח ושחור בגובה Y שהוגדר על ידי מי שקרא לפעולת ציור הצל ב SERVICE. על מנת ליצור את הצל פעולת הציור מעתיקה את הגוף משנה את צבעו לשחור ומשנה את הערכים של הנקודות לערכים על ישר העובר מנקודת האור שהוגדרה בגוף ובנקודות של הגוף כאשר ערך Y שווה לזה שהוגדר מקודם.

הצל מיושם בפרויקט שלי כאשר אני מצייר צל של המסוקים אשר עוזר למשתמש להבחין היכן הוא נמצא ולהיכן הוא צריך לזוז.

יצירת טיפות:

יצירת הטיפות מתבצעת באמצעות הגדרת מערך של גופים קטנים באורך שנקבע. יצירת הגופים מתבצעת על ידי יצירת פירמידה אחת במקום 0 במערך הפירמידה מוגדרת בגודל קטן וכך נוצרת אשליה של טיפה. לפירמידה מוגדר צבע תכלת והיא מועתקת למרכז שנקבע בקונסטרקטור של הטיפות. לאחר שהוגדרו כל התכונות הרצויות בטיפה רצה לולאה מ-1 עד גודל המערך שמעתיקה את הטיפה לכל המערך ומציבה אותה במקום אקראי בתוך המעגל שבו נמצאות הטיפות.

יצירת להבות אש:

כמו הטיפות כך גם להבות האש מוגדרות במערך ומועתקות מלהבה אחת שנוצרה. במקרה זה המערך הוא דו-ממדי. במקום 0,0 מוגדרת פירמידה מרובעת בצבע אדום אשר מועתקת לקצה היבשה. מילוי הערכים במערך מתבצע בשתי לולאות שרצות על המערך ומעתיקות את האש ומסדרת אותה במקום כך שכל היבשה מרושתת בלהבות.

סילום:

סילום היא שיטה שמקטינה ומגדילה עצמים. שינוי גודל העצמים מתבצע באמצעות שינוי ערכי נקודות העצם שינוי זה מתבצע בערכי X Y Z מוגדרים ומנקודת שבט קבועה. נקודת השבט היא הנקודה שביחס אליה גדלים וקטנים ערכי הנקודות אם לדוגמה ערך ה-X של הסילום הוא 3 המרחק של נקודות הגוף מנקודת השבט בציר X יגדל בשלוש.

בתחילת הסילום מזיזים את נקודת השבט והעצם באותם ערכים כך שנקודת השבט תהייה ראשית הצירים(ערכים הפוכים מנקודת השבט). לאחר מכן מכפילים את ערכי כל נקודות העצם בערכים שהוכנסו ל X Y Z.

הסילום מיושם בפרויקט שלי בכל פעם שאני רוצה להגדיל גוף מה שמתרחש בכל הגופים מכיוון שכל הגופים נוצרים בגודל 1.

הסתרה:

הסתרה מתרחשת כאשר שני עצמים מצוירים אחד על השני ואחד מסתיר את השני. כך נוצרת בעיה כאשר לא תמיד יש התאמה בין איזה עצם מוסתר על ידי השני. הפתרון לבעיה זו הוא לבדוק איזה עצם יותר קרוב אלינו ולצייר אותו אחרון כך שהוא יסתיר את השני. כאשר ישנה כמות גדולה יותר של עצמים מכניסים אותם למערך וממינים אותו לפי ערכי Z של הגופים ומתיירים לפי הסדר המתאים(רחוק יותר מצויר קודם).

בפרויקט שלי ההסתרה מתבצעת באופן מיוחד עקב מבנה הפרויקט ולכן יצרתי אלגוריתמים ייחודיים להסתרה בין גופים.

בעיות שנשארו פתוחות והצעות לשיפורים

בעיות שנשארו פתוחות

תלות בין גובה דלי למהירות כיבוי האש

הזרימה הקבועה של הטיפות מהווה בעיה כאשר כל טיפה שמגיעה לקרקע או לאש מוחזרת למעלה ומתחילה לרדת מחדש. כך ככל שהדלי נמוך יותר כך הטיפה תוכל לגעת באש יותר פעמים באותו הזמן ולכבות את האש מהר יותר. מקרה זה מאפשר לשחקן לרמות במשחק ורצוי שזה ייפתר.

בעיית הסתרה במילוי דלי בקדמת היבשה

כאשר ממלאים את הדלי מלפני היבשה נוצרת בעיית הסתרה בין היבשה לדלי כאשר היבשה מצוירת לפני הדלי בגלל שהדלי מתחת לגובה היבשה. כך למרות שהבעיה משנית ורבים אינם מבחינים בה רצוי שבעיה זו תיפטר על מנת שתוכל להיווצר חווית משחק מציאותית יותר.

הצעות לשיפורים

הוספת מד דלק ומילוי מחדש שלו

אפשר להוסיף ערכים שמייצגים את כמות הדלק שנשארה כאשר מפסידים את המשחק כשנגמר הדלק. בנוסף יוכלו להיות גם גופים מרחפים במרחב אשר אפשר יהיה לקחת אותם ובכך למלות את הדלק.

הוספת מכשולים לאש

המשתמש יוכל להוסיף מכשולים על הלוח אשר יוכלו לעקב את התפשטות האש.

מיכל שמתפוצץ במגע עם האש

מעין מיכל גז שיתפוצץ במגע עם האש והשחקן יפסיד את המשחק. מטרתו של השחקן תהיה להגן על מחל זה תוך כדי ניסיון לכבות את כל האש.

סיכום אישי

בעיות בהן נתקלתי:

דלי בחצי עיגול

במקור דלי המים היה בנוי מגוף בצורת חצי עיגול. הבעיה עם חצי העיגול היא מילוי וריקון הדלי. מילוי ריקון הדלי מתבצע באמצעות סילום של המים בתוך הדלי לכיוון נקודה בתחתית הדלי. כאשר מתבצעות פעולות אלה המים מתקבצים ונראים כמו מעין גוש כחול בתוך הדלי ולא כמו מים מתרוקנים.

בחרתי לעקוף בעיה זו כאשר שיניתי את צורת הדלי לגליל. כך יכולתי פשוט לכתוב אלגוריתם שמוריד את הקיר העליון של המים וכך ליצור אשליה טובה יותר של דלי מתרוקן ומתמלא.

יבשה מסתירה דלי מתמלא

לאחר שפטרתי את בעיית ההסתרה בין הדלי ליבשה נוצרה בעיה חדשה כאשר אי אפשר היה לראות את הדלי מתמלא. פטרתי בעייה זו באמצעות הקטנת הדחיסות של צבע היבשה. כך אפשר יהיה גם לדעת שהדלי ירד מתחת למים וגם אפשר לראות אותו מתמלא.

מה הייתי עושה באופן שונה

הדבר הראשון שהייתי עושה שונה הוא לתכנן כיצד יראו כל מחלקות הפרויקט לפני התחלת העבודה. חוסר תכנון הייתה בעייה כאשר עבדתי על הפרויקט מכיוון שבמקרים רבים הייתי צריך לשנות ולכתוב חלקים מחדש (בגלל חוסר תכנון).

הדבר השני שהייתי עושה באופן שונה הוא תיעוד מלא של כל התוכנה תוך כדי כתיבה. כך גם הייתי מבין יותר טוב כיצד עובדים חלקים של התוכנה תוך כדי עבודה. ועוד יותר חשוב לא הייתי צריך לחזור ולתעד את כל התכנית מחדש.

עם סיום העבודה יחסי למקצוע "גרפיקה ממוחשבת" השתפר במידה רבה. לדעתי זה היה המקצוע המעניין ביותר שלמדתי בכל 12 שנותיי בבית הספר גם בגלל העניין הרב שגיליתי בנושאים שנלמדו וגם ואפילו בעיקר בגלל העבודה על הפרויקט שהייתה חווייתית ומרתקת.

תודות

אני רוצה להודות למורה לגרפיקה ממוחשבת ממי גוטביר על שהדריך אותי במהלך העבודה על הפרויקט ושלימד אותי כיצד ליצור תוכנות גרפיות. אני רוצה גם להודות לחברי לכיתה על שעזרו לי לחשוב על פתרונות לבעיות שעלו לי במהלך הפרויקט.