

מסמך ארכיטקטורה

תיאור הפרוייקט

מטרת הפרוייקט, היא לממש משחק מחשב מסוג יריות. קיימים בשוק מגוון רחב של משחקים מסוג זה אך, שמתי לב, כי קיימת בעייה אחת ברוב המוחלט של המשחקים. הבעיה היא, ששחקנים רבים זוכרים לאחר מספר מועט של משחקים את המפה וכל דבר קטן שאמור להתרחש במשחק. בעקבות סיבה זו, עלה לי רעיון של עשיית משחק מחשב שבו המפה והמיקומים של האויבים יהיה אקראי לחלוטין וכך למעשה – השחקן שמשחק במשחק אינו יכול לזכור כלום וכל פעם הוא משחק כאילו בשלב חדש. זה למעשה החדשנות שתבוא לידי ביטוי בפרוייקט שלי ובנוסף, ניתן לציין שאוכל להוסיף אלמנטים רבים בפרוייקט זה אשר קיימים גם במשחקי מחשב אחרים מהסוג הזה – כמו לדוגמה לשחק עם שחקנים נוספים, או אפשרות של לבחור רמת קושי לשלב ולאויבים שמולם השחקן משחק. בדרך זו, המשחק שיתקבל כתוצאה סופית – גם יכיל אלמנט כלשהו חדשני שלא קיים במשחקים אחרים וגם יכיל את הדברים העיקריים שלמשחקים אחרים דומים יש להציע. בדרך זו אנו יכולים לקבל משחק יריות טוב יותר מאשר שאר המשחקים שיש בשוק.

לפרוייקט קיים סוג אחד של לקוחות בלבד – וזה למעשה השחקנים שמשחקים במשחק. סוג האנשים שיירצו להשתמש במשחק ולשחק בו הם אך ורק אנשים שמכירים את התחום ואוהבים לשחק במשחקים מסוג זה.

היקף הפרוייקט

מכיוון שמדובר במשחק מחשב, הפרוייקט חייב להכיל שימוש במנוע גרפי כלשהו וגם יכול להכיל מימוש של מבנה של שרת ולקוח(לצורך ריבוי משתתפים). בנוסף, המשחק יכול גם להכיל בסיס נתונים שישמור את השחקנים שמשחקים במשחק. אך, עקב זאת שמדובר בעבודה של בן אדם יחיד ולא של חברה גדולה כמו שבדרך כלל פרוייקטים מהסוג הזה נעשים, הגרפיקה של המשחק אינה יכולה להשתוות למשחקים שקיימים היום בשוק וגם השרתים שאצור בשביל המשחק, אינם יכולים לתמוך בכמות גדולה של שחקנים אלה לכל היותר שני שחקנים.

בנוסף הפיזיקה של המשחק והאלמנטים השונים אינם יכולים להיות מסובכים מדי - כיוון שלדבר זה נדרש תחזוק רב ועבודה רבה מאוד.

דרישות הפרוייקט

המשחק יכלול את האלמנטים הבאים:

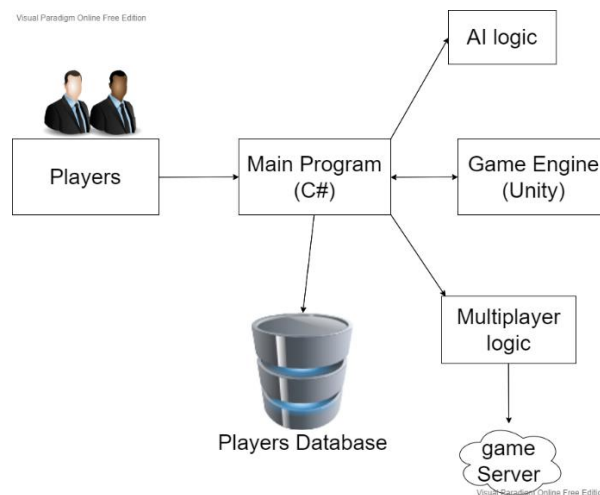
- יכולת של שחקן להתמצא בעולם של המשחק בצורה נוחה וכן לירות באויבים שונים שהוא ייפגוש במהלך השלב.
- אפשרות של שני שחקנים לראות זה את זה באותו השלב ולתקשר אחד עם השני בצורה נוחה
- תהיה קיימת דמות ממוחשבת כלשהי שתוכל לזוז ברחבי המפה ותהווה לשחקן יריב ראוי שנגדו הוא יוכל לשחק

משתמשים

כפי שהסברתי לעיל, מכיוון שזהו משחק מחשב – קיים אך ורק סוג אחד של משתמש לפרוייקט זה. המשתמש זה יהיה למעשה השחקן שישחק במשחק. זאת כיוון שמטבעו של אופי הפרוייקט, היחיד שיצטרך להשתמש בפרוייקט זה השחקנים שישחקו בזה – ומלבדם לאף אחד לא אמורה להיות הרשאה כלשהי לבצע שינויים במשחק.

ארכיטקטורה

נראה להלן את תרשימים הפרוייקט:



כפי שניתן לראות הפרוייקט יהיה מורכב מהחלקים הבאים:

1. קודם כל, יהיה את התוכנית הראשית שתהיה אחראית על כל הלוגיקה בפרוייקט וכל המבנה שלו – תוכנית זו תהיה כתובה ב-C# כיוון ששפה זו מתממשת הכי טוב אם המנוע הגרפי של Unity.
2. יהיה בסיס נתונים של כל השחקנים השונים שמשחקים במשחק. בסיס נתונים זה ישמש לצורך בדיקת אילו שחקנים מחוברים עכשיו לרשת וגם ישמש כדי להוציא סטטיסטיקות שונות ומידע על השחקנים השונים
3. תהיה מחלקה שתהיה אחראית על כל הלוגיקה של מצב המרובה משתתפים – מחלקה זו תהיה אחראית להעביר מידע דרך האינטרנט ולעדכן את המשחק בהתאם
4. לבסוף, תהיה מחלקה שתהיה אחראית על האויבים השונים במשחק ותהיה אחראית על התזוזה שלהם במשחק וגם תהיה אחראית על רמת הקושי שלהם.

תרשימי זרימה

מכיוון שמדובר במשחק מחשב, מבטבעו של הפרוייקט לא קיימים לו תרשימי זרימה מסובכים או אפשרויות שונות שהשחקן יכול לבחור כשהוא נכנס. לכן אין צורך במגוון רחב של מסכים שונים אלא, העיקר כאן יהיה אופציות שונות של משחקים שהוא יוכל לשחק כמו לדוגמא:

1. משחק בשחקן יחיד ללא אויבים כלל
 2. משחק עם שני משתתפים אחד מול השני
 3. משחק של שני שחקנים נגד אויב ממוחשב שהוא יהיה המחשב
 4. משחק של שחקן יחיד נגד אויב ממוחשב – ברמה משתנה או רמה שנקבעת מראש.
- בנוסף יהיה מסך של הגדרות שונות וסטטיסטיקות שדרכו יהיה ניתן לכייל דברים שונים במשחק.

אפיקים וסיפורי משתמש

אפיק 1: משחק של שחקן יחיד:

1. כשחקן, כל פעם שאני נכנס למשחק אני אמור מפה אקראית שונה שכנראה לא ראיתי קודם ולשחק בשלב אחר מאשר השלבים ששיחקתי בעבר כדי שלא אאבד עניין במשחק ולא אזכור איך בנויים השלבים השונים

Test case: מכיוון שהיזור סטורי מדבר אך ורק על יצירת מפה שונה - כל מה שנדרש על מנת לבדוק את הנכונות במשחק זה להריץ את התוכנה מספיק פעמים. לאחר הרצה מספר רב של פעמים, ניתן לראות אם המפות אכן משתנות כל פעם והאם נוצרות מפות הגיוניות מבחינת מראה.

2. כשחקן, אני רוצה לראות סטטיסטיקות שונות ונתונים שונים בסיום המשחק, כדי שאדע איך להשתפר בפעם הבאה וכדי שאדע איך שיחקתי.

Test case: כדי לבדוק האם הסטטיסטיקות נשמרות היטב בבסיס הנתונים, צריך לבדוק את בסיס הנתונים אחרי כל משחק. אם שיחקנו מספר משחקים בתור משתמש כלשהו ואנו רואים שהשורה המתאימה בבסיס הנתונים התעדכנה, אזי ניתן להסיק שחלק זה עובד כראוי.

3. כשחקן, אמורה להיות לי שליטה בסיסית בדמות בה אני משחק: כדי שאוכל ללכת לכל הכיוונים ולהסתובב בעולם בצורה נוחה

Test case: כדי לבדוק אם הדמות שיצרתי היא טובה ואם ניתן ללכת איתה ברחבי המפה בצורה נוחה – ניתן פשוט להסתובב עם הדמות ברחבי המפה ולראות את ההתנהגות שלה במצבים מסוימים ובמקרי קצה (לדוגמא התנגשות עם קירות שונים או מעבר בעלייה מסוימת או מדרגות).

4. כשחקן, אני רוצה יכולת של להחליף נשקים שונים ושכל נשק ייפעל בצורה ייחודית ויבזבז כמות אחרת של חיים לאויבים כדי שיהיה לי מגוון רב יותר במשחק וכדי לדמות משחקים אחרים באותו תחום.

Test case: כדי לבדוק שהחלפת הנשקים במשחק עובדת כראוי, ניתן פשוט לתת לגרום לשחקן להחליף נשקים שונים – ולבדוק גם שלכל נשק שאנו מחליפים אם אכן יש לו את המאפיינים היחודיים לו – כלומר, לדוגמא לבדוק אם אכן הנשק הוא ידני או אוטומטי ולבדוק שכל נשק מבזבז כמות אחרת של חיים (יש לבדוק שכמות התחמושת וכמות החיים אכן מנוהלים כמו שצריך ומתעדכנים כמו שצריך).

5. כשחקן, אני רוצה שיהיה תפריט כלשהו לפני תחילת המשחק עצמו, כדי שאוכל לבחור בקלות את השלב שאני רוצה ולשלוט בדברים שונים.

Test case: מכיוון שפה מדובר אך ורק על יצירת תפריט ולא אלגוריתמיקה מסובכת, כאן פשוט יש לבדוק שכל אופציה בתפריט מגיעה לאן שהיא צריכה להגיע ושהתפריט נראה טוב מספיק ולא בסיסי מדי.

6. כשחקן, אני רוצה שכשאני אהיה בתוך המשחק, לראות על המסך נתונים אשר מעניינים אותי, כמו כמות התחמושת והחיים, כדי שהמשחק יהיה כמו רוב המשחקים מהסוג הזה ואדע על המצב שלי במשחק.

Test case: כדי לבדוק אם דבר זה אכן מתקיים במשחק – יש לבזבז תחמושת ולבדוק אם כמות התחמושת שהשחקן יורה בפועל תואמת את כמות התחמושת שמוצגת על המסך – וגם אחרי פעולה של החלפת מחסנית – לבדוק שהמספר נשאר תואם בשני המקרים. בנוסף, ניתן להסתכל כל הזמן על מדד החיים ולבדוק כל פעם שהוא תואם לרגע שבו אכן השחקן מת – אם השחקן מת עם הרבה חיים או עם הוא חי ללא חיים – סימן שהתוכנית כאן לא עובדת ונדרש לתקן משהו.

אפיק 2: ריבוי שחקנים

1. כשחקן, אני אמור להתחבר לשחקן אחר שאני מכיר ולשחק איתו באותו בשלב ולאפשר אפשרות של תפריט כלשהו המראה את כל החדרים הפנויים והתחברות לחדר מסוים בקלות, כדי לאפשר אופציה בסיסית של ריבוי שחקנים

Test case: ניתן לבדוק את היוזר סטורי הזה על ידי בדיקת האפשרות לגבי אם אכן כל השחקנים רואים את אותם חדרים – לקחת כמה שחקנים שנמצאים על מחשבים שונים שכמה מהם יפתחו חדרים במשחק. אם כל השחקנים יכולים לראות את כל החדרים שנוצרו וכל שחקן יכול להיכנס לכל חדר – סימן שתכונה זו עובדת במשחק.

2. כשחקן, אני אמור לראות את השחקן השני שנמצא איתי באותו שלב: ואם הוא זז או עושה פעולה כלשהי אני אמור לראות את זה גם במשחק שלי כדי שהמשחק יתעדכן לשנינו באותה צורה ולא ייתקע

Test case: ניתן ביוזר סטורי הזה שכמה שחקנים ייכנסו לאותו החדר ולבדוק את הסנכרון בין השחקנים – אם שחקן אחד מבצע פעולה כלשהי לבדוק אם כל השחקנים אכן רואים שפעולה זו התבצעה בהצלחה. וגם אם שחקן אחד הורג שחקן אחר או הורס אובייקט כלשהו – ניתן לבדוק שאכן האובייקט נהרס ושהשחקן אכן מת לכל השחקנים באותו שלב (שלא יוצר מצב שחלק מהשחקנים חושבים שהוא מת כשהוא עדיין בתוך המשחק).

3. כשחקן, אני אמור שתהיה לי אפשרות לשוחח עם שחקן אחר שמשחק איתי בכתב ולשלוח להם הודעות כדי שתהיה לי תקשורת כלשהי אם מי שאני משחק מולו (אפשרות של צ'אט בין שחקנים).

Test case: ניתן לבדוק אם הצאט במשחק עובד כראוי – ששחקן אחד יישלח הודעות ולבדוק אם אכן כל השחקנים יכולים לענות לו וגם לשלוח הודעות בעצמם.

4. כשחקן, אני אמור לראות פרטים בסיסיים של שחקנים שאני משחק איתם: כמו השם שלהם או סטטיסטיקות שונות לגביהם כדי שאדע מול מי אני משחק ולמי התחברתי

Test case: לבדוק האם דרך התפריט בתוך המשחק ניתן אכן לראות את שמות השחקנים האחרים ומידע עליהם – ושמידע זה אכן מעודכן על סמך הפעולות שהם מבצעים.

5. כשחקן, אני רוצה שיהיה לי לפחות שני אפשרויות שונות של משחק: ללא אויבים או עם אויבים, כדי לאפשר גיוון במשחק

Test case: לבדוק שאכן כל המצבים במשחק עובדים כראוי – גם המצב עם האויבים וגם המצב ללא אויבים. מצבים אלה הם למעשה לב המשחק ויש בהם קוד רב והרבה דברים שיש לקחת בחשבון. ניתן לשחק במשחק כמה פעמים וברמות קושי שונות ולבדוק שהכל עובד כנדרש.

אפיק 3: אויב במשחק

1. כשחקן, אני רוצה שכל אויב יוכל להרוג אותי או אני להרוג אותו אם אני פוגע בו מספיק פעמים, ושהוא יעקוב אחריי בצורה הגיונית מבלי לעבור דרך חפצים, כדי שגם התנועה של האויב תהיה הגיונית וניתן יהיה להתחרות מולו בצורה הוגנת.

Test case: ניתן פעם אחת ליצור אויב בכל מפה ולבדוק את התנועה שלו – שהוא אכן מבין את מבנה המפה רואה איפה יש קיר או מכשול ואיפה הוא יכול לעבור. אם הוא יכול אכן לעבור איפה שאני יכול לעבור ואם הוא אכן מבזבז לי חיים אך ורק כשהוא מתקרב אליי מספיק – אזי האויב במשחק עובד כראוי וניתן גם לשים אותו בתוך המשחק בתוך השלבים השונים.

2. כשחקן, אני רוצה שיוצג לי על גבי המסך כמה אויבים נשארו לי להרוג, ואם נוצרו אויבים אז כמה נוצרו ולנהל את המספר שלהם כל הזמן על פי מספר הגל של השחקן, כדי שכל הזמן אדע לצפות מה נשאר לי לעשות עד לסיום השלב.

Test case: בתוכנת העריכה של unity – ניתן לראות מחוץ למשחק כמה אויבים יש ברשימת האובייקטים. צריך לבדוק שאכן המספר שרשום בעורך אכן תואם את כמות האויבים שיש, גם אם אויבים נהרגו וגם אם יש גל חדש או שנוצרו אויבים חדשים. המספר שמוצג על המסך תמיד צריך להיות מעודכן לא משנה מה קורה וגם מספר האויבים תמיד צריך להיות מתואם לפי מספר הגל שהשחקן נמצא בו – כל פעם צריכים להיווצר כמות קבועה שהוגדרה מראש.

3. כשחקן, אני אמור לבחור את הרמה של האויבים מראש, ושנתון זה יישמר בין כניסות שונות למשחק, כדי שכל הזמן הרמה שלהם תהיה מותאמת לרמה שלי

Test case: יש לבדוק שאכן התפריט של בחירת רמת הקושי של המשחק עובד כראוי – אם אני משנה שם את גודל המפה או נתון אחר – לבדוק שאכן נתון זה משפיע על חווית המשחק ומשנה אותה – ובפרט יש לוודא שאם יש מספר שחקנים שלכולם יש אותה רמת קושי ואותם נתונים והגדרות ושהגדרות אלה יהיו תואמות למה שהאדמין הגדיר.

4. כשחקן, אני רוצה שכל אויב ייוצר במקום אקראי, כדי שלא אדע לצפות בתחילת השלב היכן כל אויב ממוקם ואוסיף לרמת האקראיות של השלב.

Test case: כדי לבדוק את רמת האקראיות של האויבים במשחק – צריך לבדוק שאכן כל אויב נוצר במיקום אקראי לא משנה איזה מפה בחרנו. צריך פשוט לבצע הרבה הרצות של המשחק ולבדוק שכל הרצה המיקום של האויבים הוא מיקום אחר לגמרי.