



אב-טיפוס דיגיטלי

עכשיו שיש לך ניסיון ליצור אב טיפוס פיזי, אתה יכול לראות את התועלת שלהם בחשיבה דרך העיצוב שלך וקבלת משב מוקדם משחקנים. עם זאת, לאבות טיפוס פיזיים יש מגבלות משלהם; אם המשחק הסופי שלך ישוחרר על פלטפורמה דיגיטלית, בשלב מסוים בתהליך הפיתוח, תצטרך ליצור אב טיפוס דיגיטלי של הקונספט שלך. זה לא אומר להתחיל מאפס - האב טיפוס הפיזי שלך עזר לך בפורמליזציה ולבדוק את הבסיס של מכניקת המשחק שלך. האב-טיפוס הדיגיטלי מרחיב את עבודת העיצוב לצורה דיגיטלית ומאפשר לך לבדוק את מהות המשחק בתבנית המיועדת לו. ההבנה של המערכת הפורמלית שצברת מהחווייה הפיזית של אב-טיפוס, תתרום לך בעיצובים של המשחק הדיגיטלי שלך. זה ייתן לך מידע על ההחלטות שאתה עושה וייתן לך רעיונות אחרים שלא חשבת עליהם.

כחלק מתהליך האב-טיפוס הדיגיטלי שלך, תרצה לבנות מודלים של מערכות ליבה שיש לך שאלות עליהן: לוגיקה משחק, פיזיקה מיוחדת, סביבות, שלבים, וכו'. בנוסף, שתיים מהמשימות המרכזיות של אב-טיפוס דיגיטליים יהיו לראות את המשחק שלך באמצעות התקני הקלט והפלט של הפלטפורמה הדיגיטלית שלך. משמעות הדבר היא אב-טיפוס של מערכות הבקרה שלך, כגון מקלדת, עכבר, מסכי מגע, מחוות, בקרים קנייניים וכו'. זה גם אומר להמחיש את המשחקים שלך בצורה אינטואיטיבית.

חשוב לזכור כאשר אתה עושה אב טיפוס דיגיטליים תשקול את הסיבות שלך עבור כל אב טיפוס שאתה מבצע. האם אתה מנסה לענות על עיצוב המשחק או שאלות טכניות? האם אתה מנסה ליצור תהליך ייצור אפקטיבי? או שאתה מנסה להעביר את החזון שלך לצוות שלך או למוציא לאור? הסעיף הבא יסייע לך לעצב את אב הטיפוס הדיגיטלי שלך עבור מצבים שונים מאוד אלה.

סוגים של אבות-טיפוס דיגיטליים

ניתן לחלק את תהליך האב-טיפוס לארבעה תחומי חקירה נפרדים: מכניקת משחק, אסתטיקה (נראות, סאונד), קינאסטיקה (הרגשה), וטכנולוגיה.

1. אב-טיפוס עבור מכניקת משחק

מכניקת המשחק הם, כפי שהסברנו בעבר, התכונות של הפורמליים של המשחק. אם כבר יצרת אב טיפוס פיזי, יש לך התחלה טובה לגבי חלק זה של העיצוב. אבל יש שאלות-עיצוב שאי אפשר לענות עליהן באב-טיפוס פיזי. במקרה זה, אתה יכול להתחיל עם אב טיפוס דיגיטלי. הדבר החשוב ביותר לזכור כאשר אתה עושה את זה הוא לעשות את זה פשוט וממוקד על שאלה מסוימת - אל תנסו לשלב את כל השאלות שלכם על המשחק לתוך אב טיפוס אחד, לפחות לא בהתחלה. מאוחר יותר אתה יכול ליצור אב טיפוס מסוג שילוב של תכונות, אבל כאשר אתה מתחיל, אתה רוצה להתחיל עם הליבה שלך, בדיוק כמו שעשיתי כאשר בנתי את אב טיפוס FPS. דוגמה של שימוש באב טיפוס דיגיטליים עבור שאלות במשחקים ניתן לראות בעבודה של מעצב משחק עצמאי יונתן בלו, שהיה אחד המארגנים של סדנת משחקים ניסיוניים בכנס פיתוח משחקים. בלו דיבר בוועידה על עבודתו על המשחקים הניסיוניים הקשורים לזמן. הוא יצר משחק פעולה חדשני המאפשרת לשחקן "להריץ אחורה" זמן בדרכים יוצאות דופן, מה שהופך את התכונה הזאת לחלק אינטגרלי של המשחק. שאלה אחת שגרמה לאב טיפוס מעניין היה משהו שהוא קרא "אורקל ביליארד". בלו שאל את עצמו איך המשחק של הביליארד ישתנה אם השחקן יכול לראות את העתיד. בנוסף לראיית הכדור על השולחן, המשחק הראה את המיקומים הסופיים של הכדורים לאחר שפגע. כאשר הוא בדק את האב טיפוס, הוא הבין שזה לא כיף, אבל זה היה אינפורמטיבי. "זה לא עשה מה שרציתי", הוא אומר, "אבל זה נתן לי הרגשה שלא קיבלתי בשום משחק ששיחקתי בעבר". הלמידה מתוך אב-טיפוס זה ואחרים הובילו בסופו של דבר לעיצוב של המשחק Braid (יוטיוב: <https://youtu.be/QxXkcg-stLE> אתר רשמי: <http://braid-game.com> ויקיפדיה: [https://en.wikipedia.org/wiki/Braid_\(video_game\)](https://en.wikipedia.org/wiki/Braid_(video_game))) - משחק שהצליח מאד גם בביקורות וגם בקופות.

דוגמה נוספת מגיעה מתהליך אב-טיפוס המתואר על ידי אריק טוד במצגת שלו על המשחק Spore (אתר רשמי: <http://www.spore.com> / ויקיפדיה: [https://en.wikipedia.org/wiki/Spore_\(2008_video_game\)](https://en.wikipedia.org/wiki/Spore_(2008_video_game))). השאלה העיצובית היתה: איך לעשות את ממשק יצירת היצורים פשוט ואינטואיטיבי לשימוש, אבל עדיין מורכב מספיק כדי לתת מגוון רחב של תוצאות כך שחקנים ללא כל ניסיון עיצוב תלת-מימד יוכלו לעשות יצורים ייחודיים באמת. הוא נותן דוגמה שלחבר צוות אחד היה רעיון לתכונה, אבל כשניסה להסביר את זה לשאר הקבוצה, הם לא הבינו איך זה יעבוד. אז כדי להסביר את הנקודה שלו, חבר הצוות הרכיב אב טיפוס דו-ממדי. ההדגמה של הרעיון שלו של האב-טיפוס הזה עזר לשאר הצוות להבין בדיוק על מה הוא מדבר. התקשורת המוצלחת של הרעיון גרם לצוות להחליט ללכת קצת יותר עם אב טיפוס ולחדד את זה בתלת-ממד. בסופו של דבר, אב-טיפוס התלת-ממדי עזר להגדיר חלק גדול של המשחק.

סוג זה של אב טיפוס במשחקים דיגיטליים הוא לא רק משהו שנעשה על ידי מפתחים מקצועיים; הוא גם שימושי מאוד ופרקטי לסטודנטים ומעצבים מתחילים. במעבדת החדשנות באוניברסיטת קליפורניה, מספר אבות טיפוס נוצרו כדי לענות על שאלות רבות של הצוות על איך להשתמש במכניקה משחק כדי לעורר תחושה של הרפיה וחופש במשחק. אבי טיפוס במשחקים לא צריכים להיות תוכניות עצמאיות.

לעתים קרובות, השאלות שיהיו לך על המכניקה שלך יהיו רבות וייתכן ותצטרך להשתמש בתוכנת אקסל לצורך מעקב.

2. אב-טיפוס עבור אסתטיקה (חזותית וקולית):

הרכיבים האסתטיים של המשחק הם הרכיבים החזותיים והקוליים הדרמטיים של המשחק שלך. נסה להוסיף קצת עיצוב חזותי וקולי לאב טיפוס שלך. זה יכול לעתים קרובות לעזור לבטא את מכניקת המשחק. החוכמה היא לדעת מתי אתה מוסיף את מה שצריך ומתי אתה מבזבז זמן יקר.

בנוסף, לפעמים יש לך שאלות על סוגיות אסתטיות במשחק שלך שאתה צריך לבדוק מוקדם. לדוגמה, איך אנימציות הדמויות תעבוד עם מערכת הלחיצה? או איך פתרון ממשק חדש יעבוד עם הסביבה? כמה דרכים פשוטות לעשות זאת הן בעזרת לוחות סיפור, אמנות קונספט, אנימטיקה, טיפוס ממשק ושרטוטי שמע.

- לוחות סיפור (storyboard) הם סדרת רישומים המציגה סקיצה גסה של רצף חזותי. אלה משמשים לעיתים קרובות בקולנוע לקביעת אופן הצילום של סצנות, אך הן שימושיות גם לסצנות קטנות במשחקים ולמיפוי ממשק פוטנציאלי בתוך רמה.
- אמנות קונספט (concept art) מורכבת מציורים או רישומים של דמויות וסביבות, תוך בחינת מבטים, פלטות וסגנונות פוטנציאליים לאסתטיקה חזותית.
- אנימציה (animatic) היא מודל מונפש של המשחק בפעולה. אנימציה לא משתמשת בטכנולוגיית המשחק האמיתית והיא לא נותנת תחושה של קינסטטיקה, אבל היא יכולה לעזור לתקשר הן את האסתטיקה של המשחק וגם כמה חלקים מהמשחק.
- אב-טיפוס לממשק (interface) הוא מודל של ממשק-המשתמש החזותי. ניתן לעשות זאת בלוח סטטי או באמצעות אנימציה. ניתן אפילו לעשות זאת תחילה כאב-טיפוס של נייר ולבחון אותו לפני שעוברים לפורמט דיגיטלי.
- רישומי שמע (audio sketches) הם טיוטות מוקדמות של המוזיקה ואפקטים הקוליים שיכולים באמת לעזור לקבוע את הטון של המשחק והם מועילים להכניס חיים באנימציה ואבות-טיפוס אחרים.

הצוות ב- Insomniac Games תיאר את תהליך האבות של טיפוס אנימציה שלהם עבור המשחק Ratchet & Clank כחיסכון בזמן לא רק עבור האנימטורים אלא גם לצוות העיצוב והתכנות. "ככלל", אומר המנהל הטכני לאנימציה, ג'ון ללי, "האבטיפוס שלנו הדגיש את הפונקציה על פני סגנון עבור האנימטורים המשמעות היא שדמויות באב-טיפוס צריכות לקפוץ לגבהים הנכונים שלהם, לתקוף בהתאם למפרטי העיצוב שלהם, ולרוץ במהירויות הנכונות. "דמויות הפרוטו" נבנו עם חפצים פרימיטיביים שהיו דומים אך לא זהים לגלגוליהם העתידיים.

טיפוסי אנימציה אלה אפשרו לאמנים לבדוק תכונות כמו תזמון, מדידה ואינטראקציה עם דמויות אחרות, שלכולן יש השפעה ישירה על משחק המשחק.

באופן דומה, הצוות ב-Naughty Dog, יוצרי סדרת המשחקים של Jak, כשהם מתמודדים עם האתגר המסובך של עיצוב ממשק התאמה אישית עבור Jak X: Combat Racing, השתמשו במספר אבות טיפוס אסתטיים כדי להוכיח את רעיונות העיצוב שלהם. מנהל המשחק ריצ'רד למרצ'נד אמר: "מערכת הממשק של Jak X: Combat Racing הייתה מורכבת יותר מהממשק של כל משחק ש-Naughty Dog יצרה לפני כן, מכיוון שהשחקן היה צריך להיות מסוגל להתאים אישית את מכוניותיו ולבחור משימות מרובות משתתפים מקוונים, כמו גם לעבור דרך משחק יחיד של שחקן. עיצבנו את הממשק תחילה בתרשים זרימה המלווה ברישומי עיפרון של המסכים ואחר כך ייצרנו אב טיפוס ב-Macromedia Flash בכדי לתת לנו תחושה מהירה כיצד תורגש הזרימה בין המרכיבים השונים. כאשר יישמנו את הממשק הסופי, הצלחנו לחסוך זמן רב בגלל התגליות שעשינו על ידי פתרון בעיות בעיצוב המקורי באב-טיפוס פלאש.

3. אב-טיפוס קינאסטיקה (הרגשה):

הקינאסטיקה היא ה"תחושה" של המשחק, איך מרגישים הפקדים, עד כמה הממשק מגיב וכו', בניגוד למכניקה ואסתטיקה, שכל אחד מהם יכול להיבדק בשיטות פיזיות או אנלוגיות לפני המעבר לאב-טיפוס דיגיטלי, קינאסטיקה חייבת להיבדק על אב-טיפוס דיגיטלי. לתחושה של משחק דיגיטלי יש קשר רב לסוגי הבקרים העומדים לרשותכם. למשחק המיועד למקלדת ועכבר תהיה תחושה שונה מאוד ממשחק המיועד למסך מגע. חשוב! כשאתה מגדיר את המשחק שלך, זכור את הפקדים שיהיו זמינים בפלטפורמה הסופית כדי שתוכל לעצב אותם בראש.

דוגמה לשימושיותו של אב-טיפוס קינאסטי היא בסיפור שמאחורי התפתחות המשחק של Katamary Damacy. לקייטה טקהאשי, מעצב המשחק של קטמארי, היה רעיון לגבי משחק בו תוכלו לגלגל כדור דביק ולהרים אותו דברים. הוא היה סטודנט באוניברסיטה באותה תקופה, והאבטיפוס לקטמארי נעשה כתרגיל לתזה. המשחק, שהוסבר בעל פה או אפילו בעזרת לוחות סיפור, נשמע מוזר ואינו משתלב באף ז'אנר משחק מסוים. עם זאת, כאשר המנהלים בנמקו שיחקו את אב-הטיפוס, הם, כמו כל השחקנים מאז, התלהבו מהפשטות והקסם של המשחק.

דוגמא נוספת לאב-טיפוס קינאסטי מוצלח הייתה למעשה אנימציה עם מערך חכם. מעצב המשחק, ינו, קיבל השראה ליצור את המשחק "Osu! Tatakae!" כשראה לראשונה הדגמה של ה-Nintendo DS (אז) החדשה. הליבה של המשחק כוללת הקשה על מסך ה-DS בתיאום עם הסמנים והמוזיקה הוויזואלית כדי "לעודד" את דמויות המשחק. לדברי ינו, צוות הפיתוח הרכיב הדגמה בפלאש לרעיון החדש, המחקר את תחושת הממשק והבקרות. "כשהצגנו את זה בפני נינטנדו, שיחקתי את זה במחשב נייד. הם נגעו במסך המחשב האישי שלי כדי לקבל מושג כיצד המשחק ישחק" האב-הטיפוס של המשחק האנימטי / קינאסטי הזה נראה והרגיש די קרוב לגרסה הסופית, והקונספט נמכר למנהלים ב-Nintendo במהירות.

בקורות של אב-טיפוס יכולות לעיתים להביא להתפתחות רעיון משחק שלם. לדוגמא, כותר המוזיקה האינטראקטיבית PixelJunk 4AM החל כחקירה של Sony Move SDK. כפי שמתאר המעצב הראשי רואן פייקר, "עם צוות של שני מתכנתים ומעצב אחד, ערכנו אבות-טיפוס של לפחות 12 שיטות בקרה שונות ל-AM4, וכל זאת תוך שימוש במעבר לשליטה במוזיקה בחלל. לא משנה כמה רעיון נראה מגוון, עם זאת, המבחן היחיד שהיה חשוב היה האם נוכל לשאול, "האם אוכל לעשות מוזיקה, והאם זה כיף?" די בכדי למצוא את מערך הבקרה החדשני עליו מבוסס המשחק. "היה הרבה יותר קל ליישם תפריט ומצביעים פשוטים, אבל זה לא היה נותן את אותה חוויה כמו AM4!"

4. אב טיפוס טכנולוגי

אבות-טיפוס טכנולוגיים, כשם כן הם: דגמים של כל התוכנות שיידרש כדי שהמשחק יעבוד טכנית. זה יכול לכלול אבות טיפוס של יכולות הגרפיקה למשחק, מערכות ה-AI, הפיזיקה, או כל מספר בעיות ספציפיות למשחק שלך. זה יכול לכלול גם אב-טיפוס של תהליך ייצור המשחק. אב-טיפוס בתחום זה עוסק בבדיקה וביצוע ניפוי של כלים בזרימת העבודה להכנסת תוכן למשחק. הזדמנות לנסות רעיונות בצורה מהירה וגסה. זה עדיין לא הקוד "האמיתי".

לאחר שלמדת את מה שאתה צריך ללמוד, אתה יכול לחזור מאוחר יותר ולכתוב את הקוד האמיתי הרבה יותר נקי ומהיר. המפתח כאן הוא בעצם לעשות זאת כדי לא לקחת את קוד האב-טיפוס ולנסות להפוך אותו לקוד-המשחק הסופי. המסקנות מאב-טיפוס צריכות להיות רעיונות מופשטים, כמו אלגוריתמים או מושגי משחק.

דרך אחת טובה להימנע מהמלכודת של הפיכת האב-טיפוס שלך למוצר הסופי היא אב-טיפוס בשפת-תיכנות אחרת - משהו כמו Processing או פלאש. אם המשחק הסופי שלך יכתב ב-C++ או C#, לא תוכל להשתמש בקוד האבטיפוס ישירות. עם זאת, ישנם יוצאים מן הכלל לטכניקה זו. הרבה הפקות משחק קטנות יותר אכן מפתחות את אבות-הטיפוס שלהן ישירות לקוד המשחק הסופי שלהם. אמנם אינו אופטימלי, אך זהו תהליך ייצור מעשי עבור צוות קטן העובד בשפה יחידה.

עד כה התרכזנו באב-טיפוס יחיד למשחק שלך, האב-טיפוס הפיזי. אבל אבטיפוס דיגיטלי אפקטיבי הוא לרוב יעיל יותר כאשר הוא נעשה בפרויקטים קטנים וניתנים לשליפה. זה נקרא "טיפוס מהיר של אבות-טיפוס", וזה אומר שאתה מציב שאלה באיזה היבט של המשחק שלך, תמצא פיתרון פוטנציאלי, ואז תבנה מודל מהיר ופשוט של הפיתרון הזה, כך שתוכל לראות אם הרעיון שלך יצליח. כפי שמציינים הקר וגיגולד, אבות-טיפוס אינם מייצרים רעיונות; הם פשוט מאמתים רעיונות טובים או מפריכים רעיונות רעים. אבטיפוס מהיר טוען טענה ניתנת לבחינה ומספק למידה ניתנת לפעולה על אותה טענה.

תרגיל 8.1: מה אתה צריך לאב-טיפוס?

אילו חששות יש לך לגבי המשחק המקורי שלך – מנגנון-המשחק, האסתטיקה, הקינסטטיקה או הטכנולוגיה? מהחששות האלה, איזו נמצאת בעדיפות הגבוהה ביותר? מה יהרג את המשחק אם הוא לא יעבוד? בתשובתך, החלט היכן למקד את המאמצים שלך עבור אב הטיפוס הדיגיטלי הראשון שלך.

תכנון תוכניות בקרה

אחת ממשימות המפתח בעיצוב כל משחק דיגיטלי היא פיתוח בקורות טובות ואינטואיטיביות. במובן הטכני, משחקים דיגיטליים עומדים על שלושה דברים: קלט, פלט ו-AI. פקדים הם חלק הקלט של משוואה זו. כאשר הומצא משחקי וידאו לראשונה, הם היו מוגבלים מבחינת בקורות. סטיב ראסל וכמה סטודנטים נוספים ב-MIT תיכנתו את Spacewar בשנת 1962, שלרוב נחשב כמשחק הדיגיטלי הראשון, ובעשותם הם מצאו את מתגי המיתוג המובנים בחזית ה-DEC PDP-1 שלהם מסורבלים מדי, ולכן הם בנו בקר מיוחד משלהם כדי לעבוד עם המשחק. ל Spacewar היו רק ארבעה פקדים: סובב שמאלה, סובב ימינה, דחף ואש. בקורות עברו דרך ארוכה מאז שנות השישים. הפקדים של ימינו כוללים את המקלדת, העכבר, הג'ויסטיק, גלגלים הגה, אקדחי פלסטיק, גיטרות, תופי בונגו, מסכי מגע, חיישני תנועה, כפפות נתונים, אוזניות מציאות מדומה, זיהוי קולי ועוד. לאחרונה חלה התפתחות רבה של בקרים חלופיים, כולל בקרי VR הניתנים להעברה ותנועה וזיהוי קולי. הסמארטפונים והטאבלטים הפכו את המולטי-טאץ' לממשק בכל מקום שחוצה פלחי שוק מרובים למשחקים ומחוצה לה. ההתקדמות הזו בטכנולוגיית הבקרה מסוגלת למשוך קהלים חדשים למשחקים, כמו למשל שחקנים המעוניינים במשחק האולטימטיבי הפעיל של ה-Oculus, Vive או Sony PlayStation VR או ההצגה הפשוטה והאינטואיטיבית הזמינה בטאבלטים ובסמארטפונים שלהם. כמעצב אתה צריך לוודא שאתה מבין את יכולות הבקר עבור הפלטפורמה שאליה אתה מעצב. משמעות הדבר היא יצירת אב-טיפוס קינסטטי ובדיקת הפקדים עד שהם ישולבו בצורה מושלמת במשחק שלך.

מאמר של המעצב אריק צימרמן מ-Gamelab (זמין באתר gamedesignworkshop.com) מתאר כיצד הרצון של הצוות שלו ליצור מערך רעיונות בקרה חדש ומעניין הוביל לרעיון למשחק **Loop** שלהם. במקרה זה, הם יצרו אב

טיפוס דיגיטלי של מנגנון הליבה – הבקרה ב"looping" - ובדקו זאת ביסודיות כדי לוודא שהוא אינטואיטיבי ומהנה לפני שהם התקדמו הלאה עם הרעיון.

לאחר שהבנתם את מכשיר הקלט, עליכם לחשוב כיצד המשחק שלכם יכול לנצל אותו בצורה הטובה ביותר. דרך טובה להתחיל היא להסתכל ברשימת ההליכים לאב-הטיפוס הפיזי שלך. יש לתרגם נהלים אלה למערך בקורות דיגיטליות. לדוגמה, באב-טיפוס ה-FPS, היו נהלים להתקדם קדימה, אחורה, פנייה שמאלה, פנייה ימינה וכו'. היו גם נהלים לירי בנשק, להחלפת נשק וכו'. את כל אלה תצטרכו למפות לבקרים. אם יש לך מערכת פקדים מפורטת מאוד, סביר להניח שתסיים לקבץ אותם תחת מערכת תפריטים או מכשירים חזותיים אחרים שאפשר לגשת אליהם באמצעות בקרת יחיד או מערכת בקורות. לאחר שהחלטת כיצד הפקדים יפעלו, צור טבלת בקרה כדי לוודא שחשבת על הכל. בעמודה אחת, רשמו את הפקדים, ובעמודה הבאה, רשמו את נהלי המשחק שנגקט כאשר הפקד הזה מופעל. אם המשחק שלך מורכב, ייתכן שיהיה עליך להכין מספר טבלאות, שכל אחת מהן מייצגת מצב משחק ספציפי. לצורך בקורות, מצב משחק חדש קיים בכל פעם שהפקדים משתנים. לדוגמה, אם מדובר במשחק בו תוכלו לנהוג במכונית, להטיס מטוס או לרכב על אופניים, יהיו שלושה מצבי משחק. במקרה זה, על המעצב לנסות ולהשאיר את הפקדים דומים ככל האפשר בין שלוש המצבים הללו, כדי שלא לבלבל את השחקן.

תרגיל 8.2: פקדים מקוריים

הגדר סכימת בקרה עבור המשחק המקורי שלך. לדוגמה, אם המשחק שלך מיועד למסוף משחקים, כמו ה-Xbox, הקפד לתייג כל כפתור בבקר. אם לכפתור אין שום פונקציה, אז תייג אותו כלא פונקציונאלי. אם השליטה כוללת חיישני תנועה, תאר את תנועת הבקר עבור כל פעולת משחק.

עיצוב בקורות, כמו עיצוב משחקיות, הוא תהליך איטרטיבי. ייתכן שהניסיון הראשון שלך לא יהיה אינטואיטיבי כמו שהאמנת. הדרך היחידה שתדעו באמת אם הפקדים עובדים היא לבדוק אותם. המטרה שלך היא להפוך את הבקורות לנטולות-מאמץ ככל האפשר. שחקנים לא רוצים לחשוב על הפקדים כשהם שקועים במשחק. הם רוצים שהפקדים יהיו אינטואיטיביים. במקרה זה, תגלו שהוספת אפשרויות שליטה רבות מדי מתסכלות את המשתמש הממוצע. עבור שחקנים מומחים, בקורות מפורטות אלה עשויות להיות רצויות, כמו גם תוכניות בקרה בהתאמה אישית, אך יהיה עליכם לבצע הרבה בדיקות משחק כדי לוודא שלא תקשו על שחקנים פחות מנוסים.

בחירת נקודת מבט

הממשק למשחק דיגיטלי הוא שילוב של נקודת המבט של המצלמה של סביבת המשחק והתצוגה הוויזואלית של מצב המשחק והבקורות המאפשרים למשתמש לתקשר עם המערכת. הפקדים, נקודת המבט והממשק כולם פועלים יחדיו בסימביוטיות כדי ליצור את חווית המשחק. בדומה למערכות בקרה, נקודות המבט של משחקי הווידיאו הראשונים היו מוגבלות, והן היו מורכבות בעיקר מתיאורי טקסט של הסביבה. משחקים מבוססי טקסטים חוו התחדשות של עניין במרחב העצמאי עם עליית כלים כמו Twine (<https://twinery.org>) כלי קוד פתוח ליצירת סיפורים אינטראקטיביים.

1. נקודת מבט עליונה (Overhead)

התבוננות ישירות כלפי מטה על אובייקט היא זווית לא טבעית למדי, אך מעניקה מבט ברור על השטח וכך יכולה להיות שימושית עבור סוגים מסוימים של משחקים. משחקי דיגיטל מוקדמים השתמשו בנקודת מבט זו לא מעט, בכל דבר, החל ממשחקי ספורט והרפתקאות וכלה במשחקים כמו פק-מן.

2. נקודת מבט צדדית (Side view)

נקודת מבט צדדית נפוצה במשחקים כמו דונקי קונג, טטריס ו-Angry Birds. במקרה זה השחקן יכול לשלוט בקלות במשחק, ומשפרת את מיקוד השחקן במשחק בקצב וזמן התגובה למכשולים.

3. נקודת מבט איזומטרית

מבט זה פופולרי במשחקי אסטרטגיה, הדמיות בנייה ומשחקי תפקידים. נקודת מבט זו היא חלל תלת-ממדי ללא פרספקטיבה לינארית. זה אפשר מבט כמו מ"עין האל". המאפיין המיוחד של נקודת מבט זו הוא כמות המידע הנגישה לשחקן. משחקים כמו Myth ו-WarCraft III משתמשים בתצוגה האיזומטרית בסביבה תלת ממדית לחלוטין, ומאפשרים לשחקנים להרחיק את נקודת המבט שלהם או רחוק יותר מהפעולה. משחקי חברה רבים, כמו FarmVille 2, משתמשים בתצוגה איזומטרית.

4. מבט גוף-ראשון

זוהי נקודת המבט המועדפת בקרב שחקנים ומפתחים ונתמך על ידי תשתית טכנולוגית עמוקה של מנועי משחק וכלים הזמינים באופן נרחב לכל הרמות של מפתחי המשחק. מבחינה יצירתית, ההשקפה בגוף ראשון יוצרת מיידידות וקשר עם הדמות הראשית, ומכניסה את השחקן ממש לנעלי הדמות. השקפה זו מגבילה גם את הידע הכולל של השחקן, ומאפשרת רגעים דרמטיים של מתח והפתעה כאשר אויבים עלולים להיות בכל פינה או אפילו להתקרב מאחור.

5. מבט גוף-שלישי

התצוגה של האדם השלישי בדרך כלל עוקבת אחר דמות מקרוב, אך היא אינה מכניסה את השחקן ישירות לתצוגת הדמות. משחקי הרפתקאות, משחקי ספורט ומשחקים אחרים התלויים בשליטה מפורטת יותר על פעולות הדמות נוטים להשתמש בנקודת מבט זו. משחקים המשתמשים בגוף שלישי מאפשרים ליצור תחושת אמפתיה עם השחקן.

מרבית שאנחנו רגילים לנקודות-המבט השונות האלו, אנחנו לא תמיד חושבים מה נקודת-המבט המתאימה ביותר למשחק שלנו: מה מטרת הממשק? מה מצב המשחק, וכמה מידע צריך השחקן לדעת עליו? אלו שאלות מהותיות שתוצאות לשאלן במהלך תהליך האב-טיפוס הדיגיטלי שלכם. מומלץ לשקול ליצור אבות-טיפוס מנקודות מבט חלופיות, ובעיצובי ממשק חלופיים, כדי לראות כיצד הם משנים את חוויית המשחק.

נקודות המבט שתיארתי מספקות דרגות גישה למצב העולם, כמו גם מציבות את השחקן במערכת יחסים משתנה עם הדמות או חפצי משחק אחרים שעמם עליהם להתמודד. זה הופך את הבחירה בתצוגת הממשק לאלמנט עיצובי רשמי וגם דרמטי. האם על השחקן להרגיש קרוב מאוד לדמות המשחק, לחלוק את תחושת התנועה שלהם בנוסף לחוסר הידע שלהם לפעמים? או שמא השחקן צריך להישאר קרוב, אך מעט מחוץ לדמות, ולהיות מסוגל לראות יותר מהסביבה, אולי להרים רמזים או כלים שאולי לא נמצאים בשדה-הראיה הישיר של הדמות? איזו ראייה של המשחק הכי מתאימה לעיצוב שלך?

תרגיל 8.3 בחירת נקודת מבט

איזו נקודת מבט היא הבחירה הטובה ביותר למשחק המקורי שלך? למה? תאר כיצד בחירה זו משפיעה על האלמנטים הפורמליים והדרמטיים של המשחק שלך.

תכנון ממשק יעיל

בנוסף לנקודות המבט של המשחק, תרצו גם ליצור אב-טיפוס על האופן שבו מידע על מצב המשחק ואפשרויות הפעולה של השחקן מועברות לשחקן. לדוגמה: מספר נקודות של השחקן, מצב ההתקדמות שלו במשחק, סטטוס של יחידות אחרות, תקשורת עם שחקנים אחרים, אפשרויות שתמיד פתוחות בפניהם, או הזדמנויות מיוחדות לנקוט בפעולה. כיצד תשלב מידע זה בתצוגה הראשית שלך? הממשק למשחק, כאמור, עובד יחד עם הפקדים ונקודות המבט כדי ליצור את חווית המשחק, והוא צריך להיות מובן ביותר. בדיוק כמו בעיצוב בקורות, המטרה שלך צריכה להיות להפוך את הממשק לקל להבנה ככל האפשר. הממשק האידיאלי הוא מצד אחד חדשני ומקורי, ומצד שני מרגיש מוכר וקל לשימוש – וזו אכן בעיית עיצוב קשה מאוד. טכניקות העיצוב שלהלן הן דרכים להתקרב לתהליך שלך שיכולות לעזור למשחק שלך לשקף בו-זמנית גם חשיבה מקורית וגם רגישות לציפיות המשתמשים.

1. צורה בעקבות פונקציה

יתכן ששמעת את הביטוי "צורה בעקבות פונקציה". לואי אנרי סאליבן, האדריכל שהמציא את הביטוי הזה, הצהיר כי העיצוב של אובייקט חייב להגיע ממטרתו. אם אתם מתכוונים לבנות בניין, שאלו את עצמכם לגבי מטרת הבניין לפני שאתם מעצבים את הדלתות. אם אתם מתכוונים לבנות משחק, שאל את עצמך מהם הרכיבים הרשימים של המשחק לפני שאתה מעצב את הממשק או הפקדים שלו. אחרת תמצא את עצמך עם משחק שנראה ומתנהג כמו כל משחק אחר. כיום מעצבים רבים פשוט חוזרים ואומרים דברים כמו "המשחק שלי הוא Call of Duty, אבל הוא שוכן בכלא מאובטח במיוחד, משם אתה צריך לברוח." ברוב המקרים, המעצב ישאיל את הממשק ותוכנית הבקרה מהמשחק הנ"ל, ואז יתכן את התוכן כך שיתאים לפרמטרים האלה, עם אולי תכונה חדשה או שתיים שנזרקות פנימה. זה בסדר, וזה אולי משחק מהנה, אבל זה לעולם לא יהיה חדשני. כדי להימנע משיבוט של משחקים קיימים, עליך לחזור לרעיון המקורי שלך ולשאול את עצמך "מה מיוחד ברעיון הזה?"

הרעיון היה לברוח מהכלא. הקונפליקט ברור: על האסיר להערים על האבטחה. עכשיו איך תוכלו לעשות זאת בדרך חדשה? מה אסיר צריך לעשות כדי לברוח מהכלא? אילו סוגים של כלים וכלי נשק ומכשולים קיימים? כמעצב, עליך לתכנן כיצד לייצג את המתח של הסיטואציה הספציפית הזו ואת ההתרגשות שבה, הן בבקורות והן בממשק. התנסו בדרכים חדשות לדמיין אלמנטים אלה, הקצו להם מאפיינים ואפשרו להם ליצור אינטראקציה זה עם זה. כפי שאתה יכול לראות, הממשק מגיע מהמשחק, ולא להפך. הגישה הטובה ביותר היא לעולם לא לעצב את הממשק קודם, אלא לאפשר לו להתפתח מהצרכים הנדרשים מתפקוד המשחק. במילים אחרות, הצורה עוקבת אחר הפונקציה.

2. מטאפורות

ממשקים חזותיים הם, בשורשיהם, מטפוריים. הם סמלים גרפיים שעוזרים לנו לנווט בעולם המוזר של המחשב. אתם ודאי מכירים היטב את המטפורה של "שולחן עבודה", הנפוצה גם במערכות ההפעלה של מיקרוסופט וגם במקינטוש. תיקיות קבצים, מסמכים, קופסאות ופחי אשפה, כולם מטאפורות עבור תכונות ואובייקטים שונים של המערכת. מטאפורה זו הצליחה מכיוון שהיא מסייעת למשתמשים לקשר את חוויית העבודה במחשב עם חוויית עבודה מוכרת על שולחן-עבודה פיסי. כשאתה מעצב את ממשק המשחק שלך, עליך לקחת בחשבון את המטאפורה הבסיסית שלו. איזו מטאפורה חזותית תעביר בצורה הטובה ביותר את כל ההליכים, הכללים, הגבולות האפשריים שהמשחק שלך מכיל?

משחקים רבים משתמשים במטאפורות פיזיות המקושרות לנושאים הכלליים שלהם. כך, למשל, החפצים שהדמויות יכולות לשאת עליהן במשחק-תפקידים מונחים ב"תרמיל". מטאפורות ממשק לוקחות את העובדות היבשות והסטטיסטיקות המקושרות בזיכרון המחשב ומציגות אותן באופן שמתאים לחוויית המשחק. בעת יצירת מטפורה, חשוב לזכור את "המודל הנפשי" שהשחקנים יביאו איתם למשחק. המודל הנפשי הזה יכול לעזור לשחקנים להבין את המשחק שלך, אך הוא יכול גם לבלבל אותם. מודלים נפשיים כוללים את כל מגוון הרעיונות והמושגים שאנו מקשרים להקשר מסוים. לדוגמה, אם הייתי מכין רשימה של מושגים שעולים לי בראש כשאני חושב על קרקס, הייתי יכול לחשוב על: אמן-הטבעות, טבעות, ליצנים, חבל גבוה, מופעים צדדיים, בעלי חיים, פופקורן, צמר-סוכר, מנהל טקסים וכו'. אם הייתי עושה משחק שמשתמש במטפורה של קרקס לממשק שלו, הייתי יכול להחליט שאמן-הטבעות יהיה

המארח או מערכת העזרה. הטבעות עשויות להיות אזורי משחק שונים, ופופקורן וסוכריות עשויים להיות שיפורים (power-ups). השימוש במטאפורה זו עוזר לדמיין מידע זה בצורה משעשעת. עם זאת, אם אינך נזהר, המטאפורה שלך יכולה גם לבלבל ולהקשות על הניווט במשחק. לכל אחד מהמושגים שרשמתי יש גם מגוון אסוציאציות משלו, ולעיתים המודלים הנפשיים שאנו מביאים למטפורה יכולים לגרום לבלבול רב יותר מאשר בהירות.

תרגיל 8.4 מטפורות

צור רשימת מטפורות פוטנציאליות לממשק המשחק המקורי שלך. הם יכולים להיות כל דבר: חווה, מפת דרכים, קניון, רכבת - תבחר. עכשיו חבר קשר חופשי לכל מטאפורה למשך חמש דקות. פרט מושגים שעולים בראשך.

3. ויזואליות

במהלך משחק, השחקנים צריכים לעבד סוגים רבים של מידע כמותי במהירות רבה. דרך טובה לעזור להם לעשות זאת היא להציג את המידע בצורה חזותית, כך שיאפשר להם לדעת בקלות מה מצבם הכללי. כולנו מכירים טכניקות הדמיה: מד הדלק במכוניתך זז בקשת ממלא לריק, סרגל המדחום עולה ככל שהטמפרטורה עולה. הדוגמאות הללו משתמשות בציפיות תרבותיות כדי להצביע על משמעותן - הקשת הזזה שמאלה או מטה פירושה שכמות הדלק יורדת; פס עולה פירושו שמשו עולה. זה נקרא "מיפוי טבעי", וממשקי משחק יכולים לעשות בהם שימוש טוב. ממשק ה-Quake הוא למעשה דוגמא קלאסית לשימוש במיפוי טבעי כדי להמחיש היבט של מצב המשחק. הפנים במרכז מסמנות את הבריאות שלי. בתחילת המשחק הפנים כועסות, וככל שהדמות שלי מקבלת פגיעות הפנים נעשות חבולות ומדממות, ומאפשרים לי להבחין במעמדי במבט חטוף.

ממשק האווטאר למשחק כמו Dance Central 3 הוא דוגמה טובה נוספת למיפוי טבעי. במשחק זה התנועה משופרת על ידי אנימציות הגורמות לשחקנים להרגיש מועשרים ביכולות הריקוד שלהם.

תרגיל 8.5 מיפוי טבעי

האם יש הזדמנויות להשתמש במיפוי טבעי בממשק המשחק המקורי שלך? אם כן, רשום את הרעיונות הללו כדי להבהיר כיצד העיצובים עשויים לתפקד. אתה יכול להשתמש ברעיונות אלה מאוחר יותר כאשר אתה מתווה את עיצובי הממשק המלא שלך.

4. ארגון בקבוצות (grouping)

כשאתם מארגנים את השולחן, אתם בטח ממיינים דברים לקבוצות דומות - כל השטרות הולכים זה לזה, כל כרטיסי הביקור יחד, כל העטים והעפרונות יחד, וכו'. עיצוב ממשק דורש חשיבה מאותו סוג. עדיף לרוב לקבץ חזותית תכונות דומות כך שהשחקן תמיד יידע היכן לחפש אותן. אם יש לכם כמה סוגים של מדדי בריאות, למשל, אל תכניסו אותם לפינות שונות של המסך, תקבצו אותם זה עם זה. אם יש לך מספר תכונות לחימה, אתה יכול להקל על הגישה על ידי הצבתן על לוח בקרה יחיד. או אם יש לך תכונות תקשורת במשחק שלך, יהיה זה הגיוני לקבץ גם אלה.

תרגיל 8.6

קח ערימה של כרטיסי אינדקס ורשום פקד אחד מהמשחק שלך בכל כרטיס. מניין את הקלפים לקבוצות הגיוניות עבורך. נסה את אותה התרגיל עם שלושה או ארבעה אנשים אחרים. שימו לב לקווי הדמיון והשוני בין ההחלטות של כל אחד ואחד. האם תרגיל זה נותן לך רעיונות כיצד לקבץ את הפקד של המשחק שלך בצורה הטובה ביותר?

5. עקביות

אל תעביר את התכונות שלך מאזור לאזור כשאתה משנה מסכים או אזורי משחק. האם אי פעם שיחקת משחק בו כפתור היציאה עבר מהחלק הימני העליון במסך אחד לחלק הימני התחתון במסך אחר? אם כן, בוודאי חווית את התסכול של חוסר עקביות.

6. משוב

חשוב לאפשר לשחקן לדעת, דרך משוב חזותי או קולי, כי הפעולה שלהם נקלטה ע"י המשחק. מעצב טוב תמיד מספק משוב כלשהו לכל פעולה שהשחקן מבצע. מערכות משוב הם משהו שתרצה לשלב באב-טיפוס הדיגיטלי שלך מוקדם כדי לבדוק אם הוא מתקשר בצורה אפקטיבית לשחקנים.

משוב קולי מידע את השחקן שהקלט שלו התקבל, או שמהו חדש עומד לקרות. מצד שני, הוא אינו יעיל בדרך כלל למתן נתונים מדויקים כמו המצב המדויק של משאבי השחקן או כדי לאפשר לשחקן לדעת היכן יחידות הן. במקרה זה, סביר להניח שתצטרך להשתמש במשוב חזותי.

תרגיל 8.7

קבעו אילו סוגי משוב דרושים למשחק שלך כדי לתקשר ביעילות עם השחקן. החליטו כיצד להציג משוב זה בצורה הטובה ביותר: מבחינה קולית, חזותית וכו'.

כלים לאב-טיפוס

אולי שמתם לב שזה הפרק הראשון בספר שעוסק ישירות בתכנות המשחק שלכם. זאת מכיוון שלדעתי צריך קודם כל לעצב את חוויית השחקן, ואז לחפש פתרונות טכנולוגיים. לא צריך לתת לטכנולוגיה לשלוט בתהליך העיצוב. כל מעצב-משחק צריך לדעת תוכנות בסיסי כדי שיוכל לתקשר עם המתכנתים.

מנועי משחק

השימוש במנוע משחק לאב-טיפוס יכול לחסוך לכם הרבה זמן ומשאבים. עם זאת, זה יכול גם לדחוף אתכם להחלטות עיצוב על סמך מה שהמנוע יכול לעשות, כך שיש לו יתרון וחסרון. חלק ממנועי המשחק הם קוד פתוח, כלומר שאם יש לך את היכולת, אתה יכול להיכנס לקוד המנוע ולשנות אותו כך שיתמוך ברעיונות המשחק המקוריים שלך. אחרים רק מאפשרים לבצע פעולות משחק באמצעות התכונות הקיימות של המנוע.

Unity 3D מציע את מנוע המשחק הפופולרי ביותר עבור אב-טיפוס וגם לפיתוח אינדי. העורך שלו מאפשר אפילו למפתחים מתחילים ליצור מגוון של משחקים מתוחכמים, החל מ-D2 לתלת מימד, ולפרוס אותו בפלטפורמות מגוונות, כולל מערכות PC, Mac, Online, iOS, Android, קונסולות ומערכות VR. יש לו רשיון חינם לסטודנטים ואינדידים, כך שתוכלו להתחיל במשחק שלכם ללא תקציב גדול לרכישת כלים. לאחר שתתחיל להרוויח כסף, תוכל לשדרג למנויים ברמה הגבוהה יותר.

מנועי משחק קלים אחרים ללימוד כוללים את GameSalad, GameMaker: Studio, RPG Maker VX Ace, Studio Game Studio, ו-The Factory Factory 2. לכל אחד מאלה יש מגבלות, אך הם יכולים להציע למתכנן / מתכנת המשחקים ההתחלתי הזדמנות ליצור אב-טיפוס במהירות וביעילות.

כלי פיתוח שאינם מנועי-משחק מלאים אך עם זאת מועילים מאוד לאב טיפוס הם Flash וכפי שכבר ציינתי Processing. ולמעצבי משחק צעירים יותר, ישנם כלים כמו Scratch מבית MIT, Kodu של מיקרוסופט, Gamestar Mechanic ו-LittleBigPlanet, כולם כלים ליצירת משחקים כמו משחק או מבוססי משחק לשחקנים / מעצבים מגיל 9 או 10.

כנראה מנוע המשחק המסחרי החזק והמשתמש ביותר הוא מנוע Unreal. מנוע זה שימש למשחקים מתקדמים רבים, ביניהם BioShock Infinite, Mass Effect 3, וכמובן, למשחקי Unreal. אם תסתכל בקפידה על צילום המסך של העורך של מנוע האמיתי באיור 8.17, תבחין שהוא כולל את אותם האלמנטים הרשמיים שנמצאו באב-טיפוס של ה-FPS שלי - רשת מפה, חדרים, יחידות, חפצים וכו'. למעשה, לבלות זמן בעורך כזה זו דרך טובה להרגיש תחושה של ז'אנר משחק ספציפי. לפני מספר שנים עבר Unreal למודל תמחור ברמת כניסה ידידותית לתלמידים ומפתחים עצמאיים, המייצג בחירה מעניינת לפרויקטים עצמאיים שאפתניים. עם זאת, אם אתה רק מתחיל בפיתוח משחקים, סביר להניח שתצליח יותר עם מנוע פשוט יותר עד שתשיג יותר ניסיון.

עורכי רמות (Level Editor)

דרך שימושית ומהנה נוספת ללמוד על תכנות גם אם אין לך רקע במדעי המחשב היא להתנסות בעורכי רמה. עורכי רמה הם תוכניות המשמשות ליצירת רמות מותאמות אישית של משחקי מחשב וקונסולה. בדרך כלל הם מתבססים על גרירה ושחרור, כך שלא צריך להיות מתכנת כדי להשתמש בהם. יצירת רמה מותאמת אישית תחשוף אתכם לרכיבים הרשמיים של המשחק, ויעזרו לכם ללמוד כיצד ליצור אבטיפוס משלכם. חלק מעורכי-הרמות מגיעים עם המשחקים, וחלקם נוצרים על ידי צדדים שלישיים. רבים ניתן להוריד מהאינטרנט בחינם לאחר שרכשתם את המשחק. עורך זה מאפשר לך לערוך בלוקים, מצבים ושחקנים כדי ליצור תוכן חדש למשחק. קהילת Minecraft יצרה מאות כלים התומכים בפיתוח תוכן בתוך המשחק. עיצוב בתוך משחק כזה הוא דרך נהדרת להתחיל פיתוח דיגיטלי. ל-LittleBigPlanet (ויקיפדיה: <https://en.wikipedia.org/wiki/LittleBigPlanet>) יש גם עורך-רמה מתוחכם כחלק ממצב ה-Create שלה. ועורך-הרמות של המשחק Portal 2 שוחרר בפומבי ועם כלים חינוכיים לשימוש בבתי ספר. ניתן להשתמש בכלי זה ליצירת שיעורים המתמקדים בהעצמת חשיבה ביקורתית, חשיבה מרחבית וכישורי פתרון בעיות.

ז'אנר האסטרטגיה בזמן אמת מאפשר גם עריכה מעמיקה של רמות. מפתח המשחקים Blizzard Entertainment מכנה את עורך הרמה שלהם עבור WarCraft III כ"עורך עולמות (world editor)". זה מאפשר לך ליצור מפות WarCraft III משלך ולתמרן כמעט את כל היבטי המשחק. זה אותו עורך שמעצבי הרמה ב-Blizzard השתמשו בו כדי ליצור את רמות-ההדרכה למשחק המסחרי. היכרות עם עורכי רמה כאלה היא דרך טובה להבין עיצוב משחקי RTS בסיסיים.

אחד הדברים שאפשר לקבוע בעורך-הרמות הוא את גודל הרמה. רמה גדולה ומורכבת יותר מאריכה לעיתים את זמן המשחק, ואילו רמה קטנה ופשוטה יותר מאפשרת חוויה קצרה ולעיתים יותר אינטנסיבית. מספרי ברירת המחדל הם מספרים שנקבעו על ידי מעצבי המשחק ב-Blizzard. כשאתה מתחיל להשתעשע עם המספרים, אולי תתהה כיצד המעצבים בחרו ערכים מסוימים עבור כל יחידה. התשובה היא באמצעות אבות-טיפוס ובדיקות-משחק (playtest). ליחידות חזקות יותר יש עלויות גבוהות יותר מבחינת משאבים וזמן בנייה. לדוגמה, יחידת האבירים מגיעה עם 800 נקודות בריאות ועוצמת התקפה קרקעית של 25. היא כמעט פי שניים חזקה יותר מיחידת הרגלי, שמגיעה עם 420 נקודות בריאות ועוצמת התקפה קרקעית של 12.5. לאביר יש עלות מתואמת של 245 זהב פלוס 60 עץ לעומת עלות הרגלי שהיא 135 זהב ו-0 עץ. כמו כן, לאביר זמן בניה ארוך, 45, לעומת זמן הבנייה הקצר של הרגלי - 20, כך שתמיד יש איזון במשחק. היה צורך לערוך בדיקה מדוקדקת של כל יחידה ב-WarCraft III ולהתאים אותו למטרה על בסיס יתרונות מול עלות עד שמערכת המשחק התאזנה. אם מספר כלשהו היה מחוץ לפרופורציות, שחקנים מנוסים היו מייצרים בייצור המוני את היחידה הזו, מה שהופך את כל היחידות האחרות ללא רלוונטיות.

תרגיל 8.8 צור אב טיפוס דיגיטלי:

קח את השאלה או החשש לגבי רעיון המשחק המקורי שלך שיצרת בתרגיל 8.1, וצור פתרון פוטנציאלי לשאלה זו. התחל בפשטות, עם שאלה לגבי הבקורות או הממשק שלך, למשל. ואז פתח אבטיפוס דיגיטלי של הפתרון שלך.

מסקנות

כעת עבדת על המשחק שלך כאב-טיפוס פיזי והתחלת לעבוד על אב-טיפוס דיגיטלי. כשתמשיך, בוודאי תגלה שאלות נוספות לגבי העיצוב שלך שיובילו לרעיונות חדשים ושונים שיזדקקו לאבות-טיפוס. בדיקת משחק ואיתור מושגים אלה הם תהליך מרגש ויצירתי. הפרקים הבאים עוסקים באופן שבו אתם לוקחים את הרעיונות הראשוניים שלכם, ובאמצעות תהליך בדיקות משחק קפדני, מפתחים אותם למשחקים פועלים ומוכנים לייצור.

מקורות

• Game Design Workshop, Tracy Fullerton, Chapter 8

סיכום: גיא אנקרי. ערך: אראל סגל-הלוי.