



Design Patterns .2

Composite: עבור הבנייה של Monster והוספה של וקטור של Monster לתוך Pack)Pack מכיל ווקטור של Monster).

Strategy: ב-Character וב-Job למתודות השונות התלויות בתכונות הדמות, כמו סוחר שיקויים, יכולת קרב או solarEclipse השינויים בערכים של השחקן יקבעו לכל פעולה לפי התכונות של אותו שחקן (למשל כשנקרא לפונקציה solarEclipseEffect אז לכל השחקנים תרד נקודת כוח, ואצל הקוסם המתודה הזאת נדרסת ומעלה נקודת כוח).

Command: במתודה applyEvent של המחלקה Event, כאשר מופעלת המתודה על שחקן כלשהו היא מתבצעת שונה לכל אירוע, בהתאם לסוג האירוע, כאשר כל האירועים יורשים מ-Event (לדוגמה אם האירוע הוא Slime, אז תופעל המתודה applyEvent של Slime ויתבצע קרב של השחקן עם Slime).

Factory: עבור הבנייה של עצמים מסוג character, monster, player, event, job לצורך בנייה של וקטורי Events ו-Players בתוך MatamStory, factory class של כל אחד מהסוגים הנ"ל יוצר פוינטר מתאים לקלט שהתקבל באמצעות שימוש ב-map שהמפתחות הם הסוגים השונים של העצמים string. ב MatamStory נבנה את event באמצעות EventFactory (שתשתמש גם ב-MonsterFactory ואת players נבנה באמצעות PlayerFactory (שתשתמש גם ב-JobFactory וב-CharacterFactory).

3. במידה ונצטרך להוסיף Rogue בתור עבודה חדשה למשחק, נוסיף מחלקה בשם Rogue שתירש מ-Job. נשנה את מתודת lostBattle של המחלקה Player כך שתקבל Monster & monster (או רק את ציון הקרב של המפלצת בנוסף לנזק, כי נעדיף לא להשתמש במחלקה Monster אלא אם נהיה חייבים) וממנה תוציא את שדה ה-Damage (כמו במימוש הנוכחי) אבל עבור Rogue נוסיף בדיקה של שדה ה-combatPower, נבדוק את התנאי המבוקש (האם לשחקן יש חצי או פחות מציון הקרב של המפלצת) ונפעל בהתאם (נקיים את סיום הקרב ללא נזק או שלל).

4. במידה ונרצה להוסיף אירוע חדש בשם השראה שמימית שבו דמות מחליפה את עבודתה לעבודה אקראית חדשה, נצטרך להוסיף מחלקה שיורשת מ-Event ולדרוש את applyEvent במחלקה החדשה ולהחליף את השדה job בעבודה החדשה שקיבלנו, כלומר ניצור מצביע חדש בעזרת JobFactory ונחליף את המצביע הישן בחדש, כל שאר המימוש ישאר כשהיה.

6.3 תחרות מימז:

תרגיל בית 4

אנחנו

