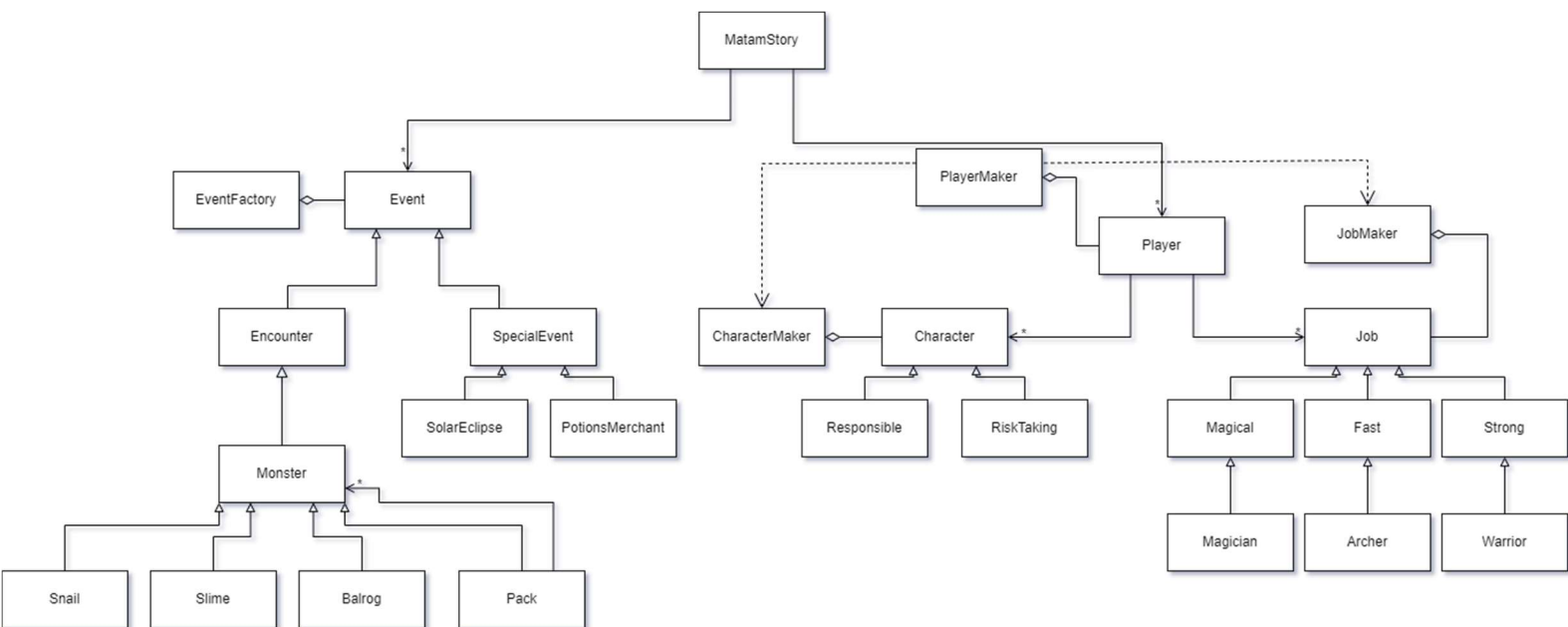


meme:

<https://www.youtube.com/watch?v=jfZOvQnsBq0>

4.1 -

.1



.2

הגדרנו את המחלקות של עבודה ואופי כמחלקות נפרדות, וזאת על מנת ששחקן יוכל לפעול באירועים בהתאם לאופי או עבודתו, בהתאם לאירוע. הגדרה זו תואמת לתבנית תכן זו.	strategy
עבור ייצור אירועים שונים, עבודות שונות, אופי שונה ושחקנים שונים, השתמשנו בתבנית תכן זו, אשר על פי הקלט מייצרת את העצמים השונים הנדרשים. במידה ובעתיד נצטרך להוסיף או להוריד אובייקטים מסוגים אלו, נוכל לעשות זאת בשינוי מינימלי של קוד.	factory
עבור מחלקת האירועים, פיצלנו כל אירוע למחלקה נפרדת וכל מחלקה דורסת את הפונקציה <code>startEvent</code> אשר מתחילה את האירוע בהתאם לאותו אירוע. כעת ב matamStory קוראים לפונקציה זו וכל מחלקה שלנו מטפלת באירוע שלה בהתאם.	command
עבור להקה - להקה היא סוג של מפלצת אך היא גם מחזיקה וקטור של מפלצות, הגדרה זו תואמת לתבנית תכן זו.	composite

3.

נשנה את המתודה

```
3  Monster::Monster(int combatPower, int loot, int damage, int con
4  std::string Monster::startEvent(Player& player){
5      if(CombatPower < player.getCombatPower()){
6          player.winFight(Loot);
7          return getEncounterWonMessage(player, Loot);
8      }
9      }
10     else{
11         player.loseFight(Damage);
12         return getEncounterLostMessage(player, Damage);
13     }
14 }
```

לכך שתתחיל ראשית בבדיקה אל מול הדמות האם היא הפסידה או ניצחה, והדמות תעביר את בשאילתה אל העבודה שלה האם היא באמת הפסידה, נעשה זאת ע"י הוספת פונקציה למחלקה הכללית של העבודות (Job), שבאופן דיפולטיבי תחזיר אמת (כי רק במקרים מיוחדים דמות ש'הפסידה' בקרב לא באמת הפסידה), ובמחלקה המייצגת עבודה של גנב נדרוס את המתודה שתבדוק האם התנאים מתאימים לכך שהיא תתחמק או לא.

במידה והתשובה שהתקבלה היא כן, נמשיך כרגיל כפי שעשינו לפני הוספת הדמות החדשה. במידה והתשובה שהתקבלה היא לא, נפעיל if חדש שנוסיף לכך שהתקבל לנו תיקו, ונוסיף לקובץ היוטיליטיז פונקציה מתאימה שתתאים לכך שהיה לנו מקרה בו דמות הצליחה להתחמק מהקרב ללא פגע.

```
85 void printWinner(const Player& player) {
86     cout << "Congratulations!" << endl;
87     cout << endl;
88 }
89
90 void printNoWinners() {
91     cout << "There are no winners in this game." << endl;
92     cout << "Better luck next time!" << endl;
93 }
94
95 string getEncounterWonMessage(const Player& player, unsigned int loot) {
96     return player.getName() + " won the encounter, gained " + to_string(loot) + " coins and leveled up!";
97 }
98
99 string getEncounterLostMessage(const Player& player, unsigned int damage) {
100     return player.getName() + " lost the encounter and took " + to_string(damage) + " damage!";
101 }
102
103 string getPotionsPurchaseMessage(const Player& player, unsigned int amount) {
104     return player.getName() + " bought " + to_string(amount) + " potions!";
105 }
106
107 string getSolarEclipseMessage(const Player& player, int effect) {
108     string out = player.getName() + " was affected by a solar eclipse!";
109     if (effect != 0) {
110         return out + " their force has changed by " + to_string(effect) + "!";
111     }
112     else {
113         return out + " their force was not changed!";
114     }
115 }
116
117 }
```

4. ע"מ להוסיף את האירוע הנ"ל, נצטרך להוסיף למחלקת player פעולה שמחליפה אקראית עבודה קיימת בעבודה נוכחית, לשם כך נצטרך לבנות ולתחזק מפעל היוצר אקראית עבודות (נוכל לנצל את המפעל הקיים בשביל החלק של יצירת העבודה, אך נצטרך לתחזק חלק חדש של הגרלה שלהן).

בנוסף, על מנת שדמות תוכל לשמור על בונוסים שהיא צברה, נצטרך להוסיף לכל עבודה מקום השומר על הבונוסים שהעבודה צברה, ונצטרך כעת בכל קריאה לפעולה מהשחקן (לקחת נזק, לקבל כסף, להפסיד כסף, להחלים וכדומה) לעבור על כל הבונוסים שצברנו כדי לוודא אם יש לנו בונוסים מסוימים (לדוגמה ליקוי חמה, או במקרה ונרצה להוסיף בונוס על החלמה מהירה או הרווחה כפולה של מטבעות וכדומה) ולהפעיל אותם במידה והם תקפים.

לאחר מכן נצטרך להוסיף בנאי העתקה למחלקה של עבודה, המעתיק את הפרטים של כל עבודה ועבודה, ומעביר אותן בין עבודות (בונוסים, מטבעות, כוח, רמה וכדומה).

לאחר שיהיה לנו את כל הדברים הנ"ל, נבצע השמה של העבודה של השחקן להיות התוצאה של בנאי ההעתקה מהעבודה החדשה על העבודה הקיימת של השחקן.