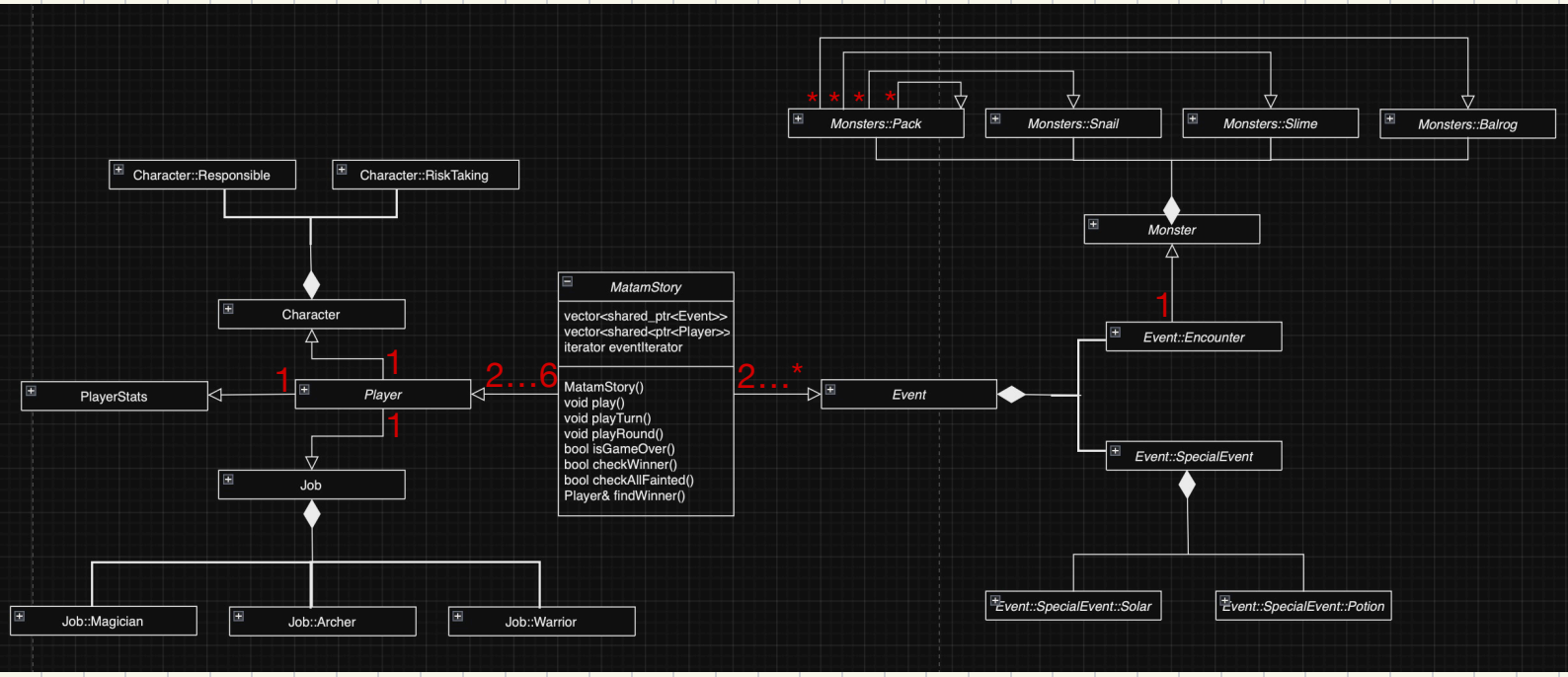

```
Turn: 3
Player: Baraa, Archer with RiskTaking character (level 1, force 5)
Event: Balrog (power 15, loot 100, damage 9001)
Baraa lost the encounter and took 9001 damage!
```

מצטערים אם לא רואים!!!
ושהחצים המלאים לא באמת חוצים....
ושבעץ הסטנדרטי העלים מתחת לענף ואצלנו לא...
ניסינו להכניס הכל שיכנס בתמונה
נצרף בעמוד האחרון UML יותר גדול אבל מסובב...



.2

Iterator

השתמשנו באיטרטור על מנת לשלוט בשחקנים ובאיוונטים כך שבעזרת האיטרטורים בכל תור נרכיב את המשחק והפעולות על השחקנים הנכונים. בסוף כל תור פשוט קידמנו את האיטרטור.

Composite

המשחק שלנו יש מפלצת Pack שמחזיקה תחתיה מספר מפלצות נוספות. בקוד שלנו כאשר נגמר התור של הPack צריכים להפעיל עליו ועל המפלצות תחתיו שינוי (Balrog

Strategy

לדוגמא לכל שחקן יש Character שמשפיע על פעולותיו במשחק

Factory

בקוד שלנו בחרנו להוציא את הלוגיקה של יצירת Players\Events למודולה נפרדת, כך בעתיד אם נרצה להוסיף עוד אפשרויות כמו Jobs, הלוגיקה של יצירת שחקן לפי חלקיו תהיה מרוכזת במקום אחד.

.3

במידה והיה נוסף Rogue כJob למשחק אזי היינו צריכים להוסיף ברמת המשחק MatamStory בפונקציה של playTurn קריאה לפני ביצוע התור ל PreFightDecision.

כלומר להוסיף לJob שהוא קלאס אבסטראקטי עוד פונקציה וירטואלית אותה כל הJobs יירשו.

כך בדומה לPostFightConsequences שאנו מפעילים על השחקן בסוף תורו שנקבע לפי הJob שלו.

מבחינת שינויים בקוד, רק קריאה לפונקציה נוספת ברמת המשחק שבודקת אם השחקן "ברח". בנוסף הוספת הפונקציה הוירטואלית PreFightDecision לכל שאר העבודות.

.4

בכדי להוסיף את האירוע החדש היינו צריכים לבצע מספר דברים:

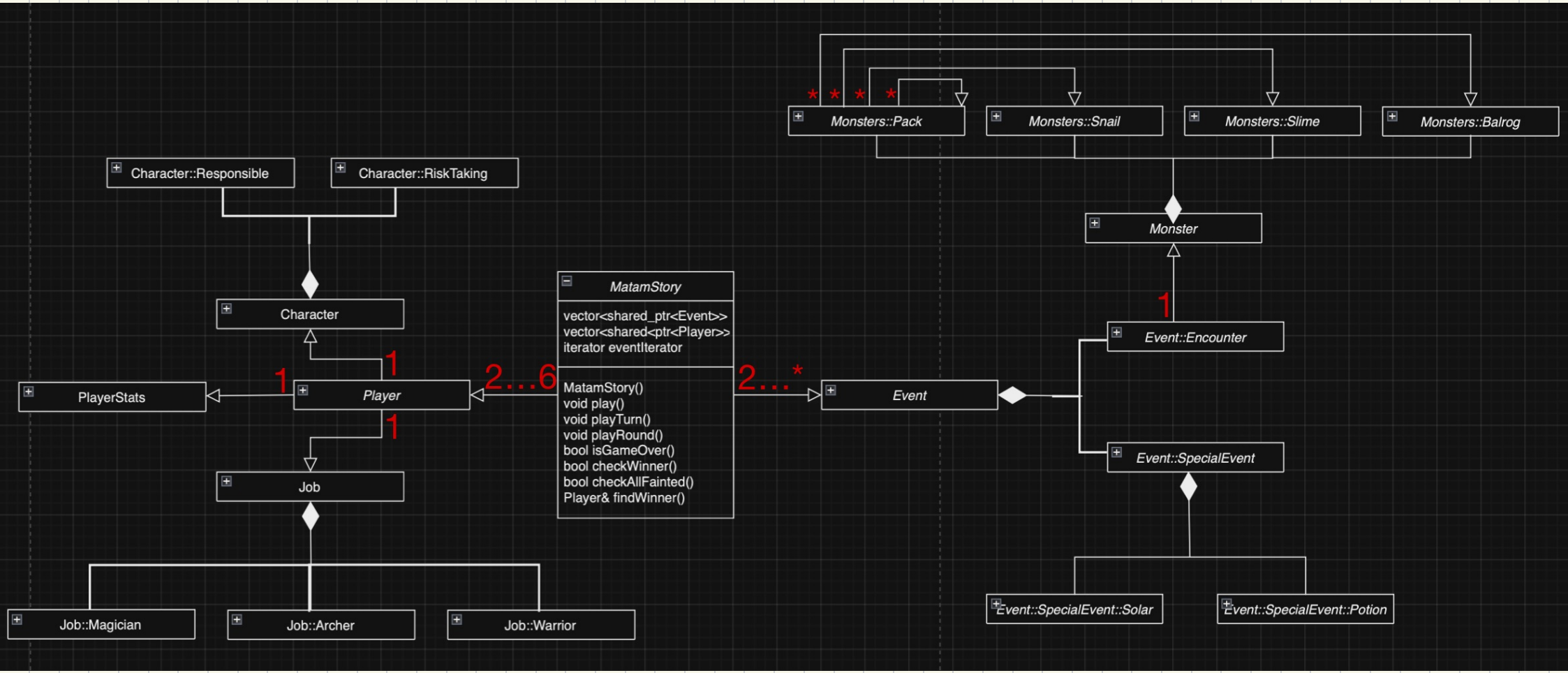
*

השתמשנו בפוינטרים חכמים אזי לנהל אותם - הכוונה היא שבזמן השינוי תחילה לגרום לשדה בתוך המחלקה של השחקן המחזיקה את המצביע החכם לעבודה שלו תחילה להצביע ל nullptr

*

בPlayer ליצור מתודה המקבלת עבודה ומחליפה את העבודה הנוכחית שלו בחדשה.

בקוד שלנו ניתן לממש דרישה זאת בקלות ע"י ירושה ממחלקת האב Event::SpecialEvent.


Character::Responsible
Character::RiskTaking
Character
PlayerStats
Player
Job
Job::Magician
Job::Archer
Job::Warrior
MatamStory
vector<shared_ptr<Event>>
vector<shared<ptr<Player>>
iterator eventIterator
MatamStory()
void play()
void playTurn()
void playRound()
bool isGameOver()
bool checkWinner()
bool checkAllFainted()
Player& findWinner()
Event
Monsters::Pack
Monsters::Snail
Monsters::Slime
Monsters::Balrog
Monster
Event::Encounter
Event::SpecialEvent
Event::SpecialEvent::Solar
Event::SpecialEvent::Potion
1
1
1
1
2...6
2...*
*
*
*
*