

תרגיל בית 4 – מבוא לתכנות מערכות

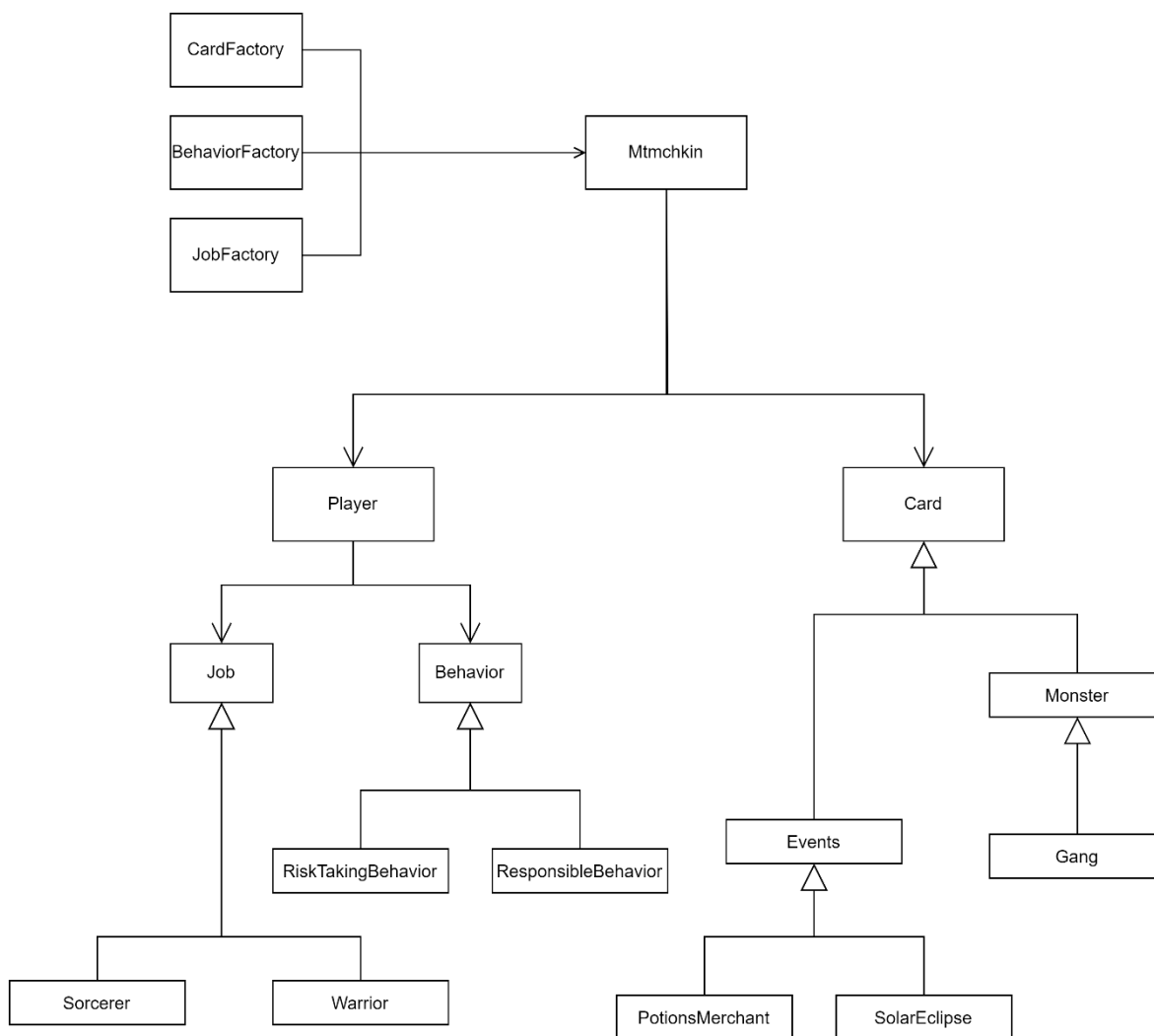
מגישות:

ליאל עזרן 205961691

ג'ינאן מחאמיד 212969372

חלק יבש

1.



2. כן, השתמשנו בתבניות תכן הבאות:

- **strategy design pattern** בחלק של Player – עבור Behavior וגם עבור Job. בתוך כל מחלקה שיושט Job \ Behavior הגדרנו את האסטרגיה שעל השחקן לבצע אם הוא נתקל בקלף של אירוע מסויים או קרב על פי האופי \ העבודה שלו. בנוסף, עבור Warrior יש אסטרטגיה אחרת לחישוב כוח ההתקפה מאשר לשחקן שהוא לא warrior.
Behavior – הגדרנו מתודה handlePotionMerchant שהיא אחראית על בניית אסטרגיה של שחקן להתמודד מול קלף שכזה בהתאם להתנהגות שלו, והחזרת כמות שיקויים לקנייה. בכל

handlePotionMerchant של מחלקה עם התנהגות מסויימת יש אסטרגיה אחרת ולכן תוחזר כמות שיקויים שונה.

Job – הגדרנו מתודה handleSolarEclipse שהיא אחראית על בניית אסטרגיה של שחקן להתמודד מול קלף שכזה בהתאם עבודה שלו. בכל handleSolarEclipse של מחלקה עם עבודה מסויימת יש אסטרגיה אחרת ולכן יוחזר ערך אחר שיתאר את הסימן של עדכון הכוח (קוסם יחזיר +1 על מנת לעדכן את הכוח ב1, ועבור כל שחקן עם עבודה אחרת נחזיר -1). בנוסף, עבור Job הכנו גם מתודה handleBattle, ששם מוחזר תנאי אם השחקן ניצח או לא (אם כוח ההתקפה שלו גדול משל המפלצת). הקריאה לכוח ההתקפה גם נדרסת ע"י מתודה של לוחם – שעבורו מחושב כוח התקפה שונה משל שחקנים אחרים. החלטנו לשים את handleBattle בתוך Job ולא כמתודה רגילה מתוך הקלפים\השחקן מאחר וצפינו בעזרת החלק היבש שיכול להיות שיהיו בעתיד עוד דמויות שינהלו את הקרב מול המפלצות בצורה שונה, ולא בהכרח רק על סמך כוח ההתקפה שלהם, אלא גם עפ"י עבודתם.

- **Command pattern** בחלק של Card – לכל קלף תפקיד אחד ויחיד, קלף מפלצת – קרב, קלף שיקויים – קניית שיקויים וריפוי השחקן, קלף ליקוי חמה – עדכון כוח (הורדה\הוספה של כוח).

לכן, כאשר אנחנו קוראים למתודה של Card שהיא applyCard, היא תדרס ע"י כל applyCard של הקלפים – מפלצת \ ליקוי חמה\ שיקויים, בהתאם למה שהקלף עושה. בתוך כל מתודה applyCard של המחלקות היורשות מ-card, נקרא לאסטרגיה של השחקן להתמודדות עם אותו קלף ספציפי, בהתאם לעבודה\אופי שלו.

- **Factory design pattern** בחלק של Mtmchkin – הגדרנו 3 מחלקות:

- CardFactory: אחראית על יצירת הקלף הנוכחי בהתאם לשם הקלף שהתקבל בפלט.
- BehaviorFactory: אחראית על קביעת ההתנהגות של השחקן בהתאם לשם שהתקבל בפלט.
- JobFactory: אחראית על קביעת העבודה של השחקן בהתאם לשם שהתקבל בפלט.

3. ניתן להוסיף מחלקה נוספת בשם Ninja שתורש מהמחלקה Job. במקרה הזה, נצטרך לדרוס את הפונקציה handleBattle במחלקה Job, וליצור פונקציה handleBattle חדשה עבור המחלקה Ninja. שם נבדוק את התנאי שביקשו בשאלה (בשונה מהתנאי הקיים כרגע) ונשלח חזרה applyCard לtrue/false של Monster במידה והשחקן ניצח או הפסיד בקרב, ונמשיך לבצע את כל הדרישות שקורות לאחר הקרב (קבלת כסף \ גרימת נזק וכו'). בנוסף, נוסיף לוקטור העבודות שלנו את השם Ninja ונגדיר את המחלקה בצורה דומה לאיך שהגדרנו את העבודות האחרות (מלבד handleBattle שצריך לדרוס כפי שהזכרנו קודם).

4. כן, ניתן לממש את הדרישה עבור האירוע החדש. נוסיף למחלקה Card מחלקה יורשת חדשה מסוג DivineInspiration. נוסיף מתודה למחלקה החדשה בשם applyCard שתדרוס את המתודה applyCard של המחלקה האבסטרקטית Card. מתוך המתודה הזאת נקים עבודה חדשה בעזרת <Job>unique_ptr ונקרא למתודה של השחקן setJob שמקבלת <Job>unique_ptr ומעדכנת את m_job.