שמות המגישים :

עוז נשיא - 308484443 [oznasi1@gmail.com](mailto:oznasi1@gmail.com)

עמית דרור – 303042667 amitdr@mta.ac.il

תחילה חילקנו למחלקות אשר לכל אחת מהן תחום אחריות אחת.

בעקבות המעבר ל REACTjs חילקנו את כל מה שמוצג למסך לקומפוננטות כמה שיותר קטנות אשר יודעות לצייר את רצוננו. לדוגמא, קופננטה של קלף מיממשנו באמצעות פונקציה ריאקטית שכל שיודעת לקבל את ה ATTRIBUTE הרלוונטי ולצייר למסך.

דוגמא נוספת, עבור מחלקה ריאקטית הינה ה – GAME הכללי שלנו אשר מכיל בתוכו בתוך ה STATE את כל המידע שבעקבות שינוי ולו באחד מהמידע ייתבצע רינדור מחדש של הקומפוננטות הרלוונטיות.

כל הקומפונטטות נכתבו בקובץ INDEX.js.

כמו כן ישנה מחלקה נוספת חשובה, Controller.

זו אחראית על התקשורת בין המנוע לבין עדכון ה STATE. בכל פעם שיש שינוי במנוע המשפיע על רכיב UI פונקציה בקונטורלר נקראת ומבצעת SETSTATE. כך בעצם הצלחנו לחבר את המנוע שנכתב בתרגיל 1 ולהשמישו בתרגיל הנוכחי.

2 מחלקות עיקריות נוסף על מחלקת הUI כמובן, האחת gameEngine

והאחרת. actionManager.

המחלקה הראשונה אחראית "לעבד" את המידע מהפרונט לשלוח אל המחלקה השנייה - actionManager

ולקבל תשובה האם הפעולה שהתרחשה כרגע במערכת חוקית או לא חוקית.

gameEngine מתחלקת לשתיים , בתחילתה , מקבלת את הקלף הנוכחי שנלחץ (ע"י השחקן או ע"י מחשב) לאחר מכן שולחת אל ה – actionManager וזו עונה לה , האם הפעולה חוקית -מוסיפה , אחרת מחזירה Enum מתאים.

ובחלקה השני , אחראית "לתפוס" את תשובה של actionManager ולפעול בהתאם.

לא הספקנו לממש בונוסים לצערנו אך ניסינו ליצור חווית משחק חמודה ככל האפשר בזמן המוקצב. (הפעם הוספנו השהייה מינימלית כדי שנראה קצת את הבוט משחק 😊)