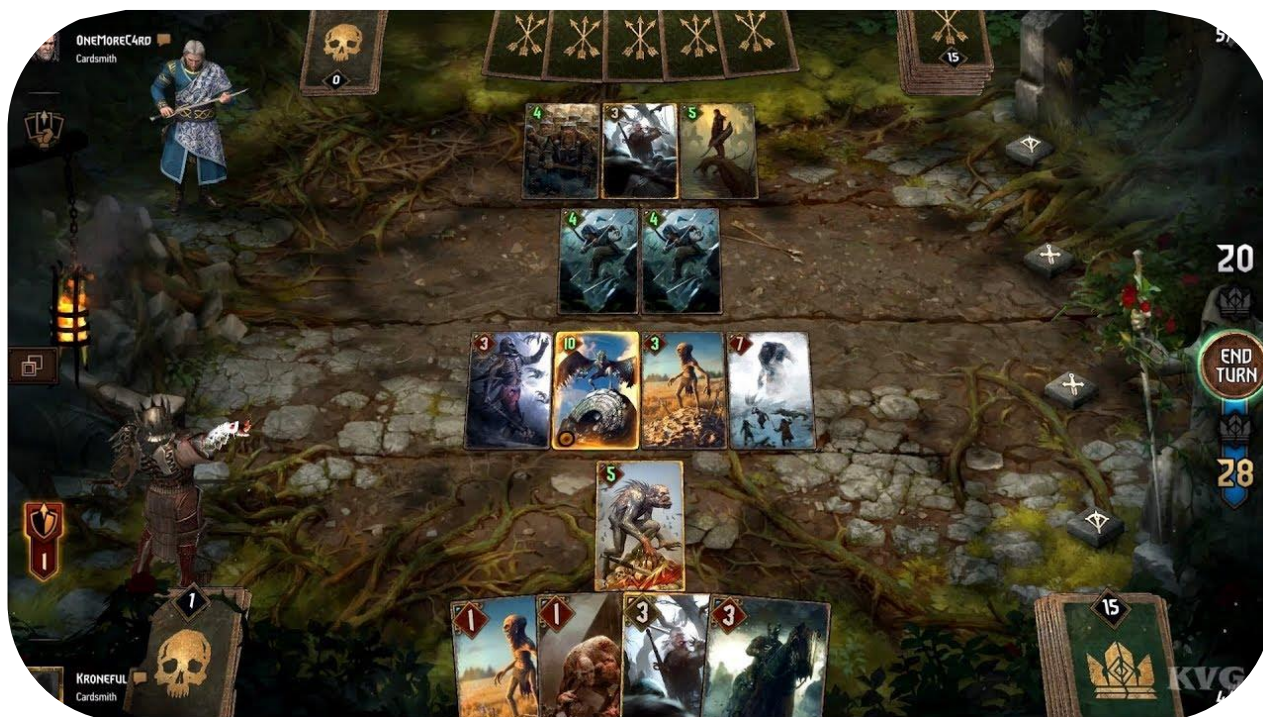


پروژه سوم برنامه سازی پیشرفته

# Cursed Battle



## مقدمه

در این پروژه، قرار است یک بازی توسعه دهیم که بازیکنان با داشتن یک مجموعه از کارتها دارای قدرت و قابلیت‌های خاص، شروع به بازی می‌کنند و هدف آن‌ها کسب امتیاز بیشتر با استراتژی‌ها و تصمیم‌هایی است که در قرار دادن کارتها در زمان و محل‌های مناسب روی میز بازی به کار می‌برند. می‌توانید بازی را از این [لینک](#) دانلود و نمونه گیم پلی را از این [لینک](#) ببینید. پروژه نمونه بسیار ساده شده این بازی بدون رابط کاربری است، هر موردی که در پروژه ذکر نشده مد نظر **نخواهد بود**. پروژه با جزئیات کامل بخش‌های اصلی برای عدم گیج شدن شما طراحی شده، پیاده سازی آن به اندازه ظاهرش پیچیده نبوده و امکان پیاده‌سازی اکثر نمرات امتیازی را خواهید داشت.

# محیط بازی

تصویر نمونه از پیاده سازی محیط بازی در کنسول:

```
1 - PC have summoned the TRISS into the 2S
2
   Cursed Battle
   +-----+ +-----+ +-----+ +-----+
   | 1Q | | 1R | | 1S | | 1T |
   |   | |   | | EITHN | |   |
   |   | |   | | 3 | |   |
   +-----+ +-----+ +-----+ +-----+
   | 2Q | | 2R | | 2S | | 2T |
   |   | |   | | TRISS | | VESSE |
   |   | |   | | 1 | | 9 |
   +-----+ +-----+ +-----+ +-----+

   +-----+ +-----+ +-----+ +-----+
   | 3Q | | 3R | | 3S | | 3T |
   | LAMIA | | ODRIN | | EREDN | |   |
   | 4 | | 2 | | 6 | |   |
   +-----+ +-----+ +-----+ +-----+
   | 4Q | | 4R | | 4S | | 4T |
   |   | |   | |   | |   |
   +-----+ +-----+ +-----+ +-----+

3 # Round 3 / Turn 4|
4 PC: 0 – YOU: 1
5
5 # Score
  PC: 13 – YOU: 12
6
   ----- Choose a card index and place to summon (e.g., 01Q)-----

0. GERAL: Geralt of Rivia - Power: 6
  • The legendary monster slayer
  • Deploy: Destroy the strongest enemy unit.

1. YENNE: Yennefer of Vengerberg - Power: 5
  • A powerful sorceress
  • Deploy: Damage all enemy units by 2.
```

موارد زیر باید مطابق نمونه عکس قرار داده شده (به ترتیب نوار سبز راهنما) چاپ شده و بعد از هر ورودی، کنسول پاک شده و کل محیط بازی دوباره چاپ شود.

۱. آخرین رویداد گذاشتن کارت در محیط بازی (به جز نوبت اول)
۲. میزبازی (به دو نیمه هرکدام شامل ۸ خانه برای جایگذاری کارتها تقسیم شده)
۳. شماره راند و نوبت بازی
۴. تعداد برد راند هر بازیکن
۵. مجموع امتیاز فعلی هر بازیکن
۶. لیست دسته کارت بازیکن

## کلیات نحوه بازی

**شروع:** شما و رایانه یک دست کارت دریافت می‌کنید. هر دست کارت شامل ۱۰ کارت است که به صورت تصادفی از کل کارتهای بازی انتخاب شده اند.

**قرار دادن کارت:** شما و حریفان به ترتیب در هر نوبت یک کارت از دست خود انتخاب و در یکی از خانه‌های بازی قرار می‌دهید، هر کارت دارای یک عدد به عنوان قدرت بوده و ممکن است خصوصیات و توانایی‌های ویژه ای داشته باشند که روی سایر کارتهای بازی تاثیر بگذارند. هدف شما در این مرحله، بازی با کارتها به نحوی است که در پایان مجموع قدرت کارتهای روی صفحه بازی شما بیشتر از حریف باشد.

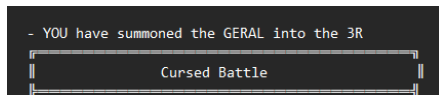
**راند:** بازی از ۳ راند تشکیل شده و تا زمانی ادامه پیدا میکند که کارتهای هر دو بازیکن تمام شود. در پایان هر راند قدرت کارتهای هر بازیکن با هم جمع شده و بازیکنی که قدرت بیشتری داشته باشد، برنده راند خواهد شد.

**پاس:** هر کاربر می‌تواند قبل از پایان راند، پاس بدهد. اینکار او را بدون شمارش امتیاز، بازنده راند می‌کند.

**پایان:** بازیکنی که ۲ راند را ببرد برنده کل بازی می‌شود.

# پیاده‌سازی گیم پلی

1. در مرحله اول محیط بازی بدون هیچ کارتی در میز بازی آغاز می‌شود.
2. هر دست کارت شامل ۱۰ کارت است که به صورت تصادفی انتخاب شده و می‌توانند تکراری باشند.
3. این کارت‌ها در زیر میز بازی همراه با شماره، نام مخفف، نام کامل، قدرت، توضیح کارت و ویژگی‌های خاص (در صورت وجود) چاپ می‌شوند.
4. سپس از بازیکن خواسته می‌شود یک کارت را از دست خود انتخاب و در درون میز بازی قرار دهد. اندیس هر خانه در بالای آن نوشته شده، و در اینجا اگر بازیکن بخواهد کارت 0 را در خانه 1Q قرار دهد باید دستور 01Q را وارد کند



- در بالای میز بازی باید وارد شدن این کارت به میز بازی چاپ شود، مانند:
  - در صورت یافت نشدن ورودی یا پر بودن خانه، دوباره ورودی گرفته شود
  - بعد از قرار دادن کارت، ویژگی deploy کارت با فرخوانی این متد اعمال می‌شود
  - در صورتی که کارتی نداشته باشد، مجدداً نوبت حریف می‌شود
5. کامپیوتر به صورت رندوم یک کارت را درون بازی قرار می‌دهد
    - در بالای میز بازی باید وارد شدن کارت انتخابی به میز بازی چاپ شود
    - بعد از قرار دادن کارت، ویژگی deploy کارت با فرخوانی این متد اعمال می‌شود
    - در صورتی که کارتی نداشته باشد، مجدداً نوبت حریف می‌شود
  6. بازی دوباره به مرحله 4 رفته و تا زمانی ادامه پیدا می‌کند که یکی از بازیکنان با وارد کردن P پاس بدهد، یا کارت هر دو بازیکن تمام شود
  7. بعد از پایان راند، تمام کارت‌های روی میز بازی حذف شده و به هر بازیکن سه کارت جدید داده خواهد شد و بازی به مرحله 3 می‌رود
  8. بازی خاتمه یافته و برنده مشخص می‌شود

# پیاده‌سازی کلاس‌های اصلی پروژه

## Card

این کلاس مشخصات یک کارت را نگهداری می‌کند، سایر کارت‌ها حتما باید از این کلاس ارث بری و متد `deploy` را `override` کنند. سطح دسترسی همه ویژگی‌های این کلاس باید کنترل شود.

### ویژگی‌ها:

- نام مخفف
- نام
- قدرت (در صورتی که به صفر یا منفی برسد از بازی حذف خواهد شد)
- توضیحات
- توضیح ویژگی `deploy` (در صورت وجود)

### متدها:

- `deploy` (این متد وقتی صدا زده می‌شود که کارت درون بازی قرار گیرد)
- `damage` (قدرت را کم و صفر شدن آن را چک میکند)
- `boost` (قدرت را افزایش می‌دهد)

### راهنمایی (اعمال این موارد اجباری نیست)

برای اینکه بتوانید ویژگی‌های کارت را روی سایر کارت‌ها اعمال کنید، در هنگام ساخت `Card`، `Game Board` را پاس داده و ذخیره کنید. و در ادامه به عنوان مثال برای متد `deploy` در `Yennefer`، میتوانید از حلقه `for` برای گرفتن تمام کارت‌های بازیکن مقابل از `Game Board` و فرخوانی تابع `damage` استفاده کنید

```
1. YENNE: Yennefer of Vengerberg - Power: 5
• A powerful sorceress
• Deploy: Damage all enemy units by 2.
```

## Game Board (Interface)

این اینترفیس (به صورت کلاس تعریف نشود) نماینده میزبازی است و حالت فعلی آن را نگهداری می‌کند.

### ویژگی‌ها:

- لیست دست Card های بازیکن ۱ (کامپیوتر)
- لیست دست Card های بازیکن ۲ (شما)
- ماتریس Card های قرار گرفته در صفحه بازی

### راهنمایی (اعمال این موارد اجباری نیست)

می‌توانید ویژگی‌ها را مطابق میل خود شخصی سازی کنید (برای مثال می‌توانید از چهار آرایه مجزا هم به ازای هر ردیف، به جای ماتریس استفاده کنید)

متد هایی برای گرفتن همه کارت های یک بازیکن، یک سطر خاص، کارت با بیشترین و کمترین قدرت و هر متدی که ممکن در deploy نیاز داشته باشید را به این اینترفیس اضافه کنید (صرفاً در اینجا تعریف و در Main پیاده سازی (override) کنید)

## Main

این کلاس اینترفیس **Game Board** را Implement می‌کند. شما می‌توانید منطق این بخش را به شکل دلخواه پیاده سازی کنید.

### راهنمایی (اعمال این مورد اجباری نیست)

چرخه ساده سازی شده برای کمک به درک بهتر نحوه پیاده سازی بازی قرار داده شده، دقت کنید این راهنما، ممکن است کامل نباشد و ملاک نحوه پیاده سازی نیست. توصیه می‌شود **حتما** ابتدا خودتان به نحوه پیاده سازی آن فکر کنید.

۱. دادن ۱۰ کارت رندوم به هر دو بازیکن
۲. چاپ خروجی بازی
۳. در صورتی که کارت هر دو بازیکن تمام شود، تعداد برد راند بازیکنی با بیشترین امتیاز را یکی اضافه کن
  - پایان بازی در صورت برد یک بازیکن
  - اضافه کردن ۳ کارت رندوم به هر دو بازیکن
  - حذف تمام کارت های ماتریس بازی
۴. اگر کارت های این کاربر تمام شده رفتن به مرحله 9
۵. گرفتن دستور جدید از کاربر
۶. در صورتی که بازیکنی پاس داد، تعداد برد راند بازیکن حریف را یکی اضافه کن
  - پایان بازی در صورت برد یک بازیکن
  - حذف تمام کارت های ماتریس بازی
  - اضافه کردن ۳ کارت رندوم به هر دو بازیکن
۷. قرار دادن کارت در قسمت انتخاب شده (یا نمایش خطا)
  - کارت از دست بازیکن حذف می‌شود
  - به اندیس مناسب در ماتریس بازی اضافه می‌شود
۸. فراخوانی متد Deploy کارت
۹. شروع مجدد چرخه برای حریف

## قابلیت ذخیره سازی

بازی باید گزینه‌ای برای خروج و ذخیره بازی داشته باشد، به شکلی که با زدن آن بازی بسته شده و هنگام دوباره بازکردن آن، دقیقا همان بازی از سر گرفته شود.

## کارت‌های بازی

همه کارت (کاراکتر) ها باید از Card ارث بری کنند، و همه مشخصات کارت باید در کلاس ارث بری شده (شامل نام، قدرت و ...) مشخص باشد. همه کاراکترهای Normal و ۲ مورد دلخواه از Silver، و یک کاراکتر دلخواه (باید مخصوص پروژه شما باشد) را پیاده سازی کنید (مجموعا 13 کارت).

پیاده سازی کارت های بیشتر با توجه به نوع آنها، نمره اضافی با ضریب زیر خواهند داشت:

- Normal: 1
- Silver: 5
- Gold: 10

**نکته:** حداکثر 5 کارت اضافی می‌توانید طراحی کنید

**نکته:** کارت‌های نقره‌ای نیاز به تغییر جزئی در منطق اصلی بازی خواهند داشت

**نکته:** می‌توانید تغییرات لازم برای کارت ها برای بالانس کردن گیم پلی را اعمال کنید، نام و مشخصات و قدرت کارت ها را شخصی سازی کنید، ولی نباید Deploy را تغییر دهید.

**نکته:** می‌توانید کارت های دلخواه خود را به بازی کنید. قبل از پیاده سازی **دقیقا** با شکل نوشتاری کارت های معرفی شده زیر، در گروه اعلام کنید (تا سه کارت) تا رده‌بندی شوند. (قدرت‌های با منطق تکراری حتی Gold، به عنوان Normal رده‌بندی خواهد شد)



# Normal

## **SLDRS: Poor Soldiers - Power: 3**

- Weary and impoverished soldiers, bearing the scars of conflict.
- Deploy: Damage a random enemy unit by 3.

## **ARCHRS: Not Poor Archers - Power: 2**

- Skilled marksmen with deadly precision.
- Deploy: Damage the weakest enemy unit by 3.

## **KNIGT: Holly Knights - Power: 4**

- Devout warriors of righteousness and divine protection.
- Deploy: Boost all units in this line by 2.

## **GIANT: Giant - Power: 7**

- Just a colossal creature.
- Deploy: none

## **WIZRD: Dark Wizard - Power: 3**

- A sinister and enigmatic wizard, whispering curses in the shadows.
- Deploy: Damage 5 random enemy unit by 1.

**UNIC: Unicorn - Power: 2**

- A majestic and mythical being that we have no idea WTH it doing here.
- Deploy: Boost all allied units by 2.

**GOLEM: Golem - Power: 5**

- A mighty and formidable creature formed of solid stone.
- Deploy: Boost adjust (right and left) units by 2.

**GRIFF: Griffin - Power: 3**

- A majestic creature with the body of a lion and the wings of an eagle.
- Deploy: Damage most strong enemy unit by 5.

**YGHRN: Yghern - Power: 2**

- An ancient and unstoppable force, master of cautious execution.
- Deploy: Destroy the weakest enemy unit.

**PHOEN: Phoenix - Power: 2**

- A legendary and eternal bird of fire.
- Deploy: Boost the weakest allied unit by 3.

# Silver

## **GOWAR: God of War - Power: 3**

- The indomitable deity of conflict and strife.
- Return to the player's hand and boost itself by 3 if killed in battle.

## **LEGOL: Legolas Greenleaf - Power: 3**

- The Prince of the Woodland Realm and a master archer.
- Deploy: Damage chosen enemy unit by 3.

## **DRACL: Dracula - Power: 3**

- The legendary vampire lord, ruler of the night.
- Deploy: Drain the all life of chosen friendly unit (destroy it) and boost itself by the same amount.

## **ABRAC: Abracadabra**

- A forbidden incantation, Unleash the full force of magic.
- Cast: Destroy chosen enemy unit and the weakest friendly unit.

**Cast:** Works like **Deploy**, but the card will not be moved to the board, and its ability is applied instantly to the chosen card.

### **DUMBL: Dumbledore - Power: 4**

- The venerable and enigmatic Headmaster of Hogwarts
- **Deathwish:** Destroy the card how kill him

**Deathwish:** This ability trigger when the card is destroyed by another unit in the battlefield.

## **Gold**

### **DEMEN: Dementors - Power: 0**

- Soul-draining specters of despairs
- Deploy: Suck half-life of chosen enemy unit and boost their self and damage enemy unit by same amount
- Adjust (right and left) units' power will be decreased by 2, unit dementors removed from the board
- Targeted unit will damage by 1 each turn

## نمره امتیازی (نمره پروژه ۳۰۰ و شرط دریافت نمرات امتیازی است)

- طراحی GUI به نسبت پیاده‌سازی و طراحی سایر بخش‌های امتیازی (40 تا 80)
- صدا گذاری به نسبت پیاده‌سازی (5 تا 10)

### قابلیت Dual-Play (20)

امکان بازی همزمان دو نفر با یک دستگاه را اضافه کنید

۱. منو بازی برای انتخاب این حالت ایجاد شود
۲. بازیکن با وارد کردن N، نوبت خود را تمام کند
۳. باید 5 ثانیه مهلت سوئیچ کردن به بازیکن داده شود تا بازیکن حریف کارت‌ها را نبیند (در این مدت در صفحه تنها پیام سوئیچ کردن نمایش داده شود)
۴. **Player** را با ویژگی‌های:

- لیست دست Card های بازیکن
- مجموع امتیاز و راندهای برده شده
- play (دریافت دستور برای ادامه بازی)

و ارث بری کلاس **PCPlayer** از آن و override کردن play پیاده سازی کنید.

در این حالت منطق بخشی از برنامه را جدا کرده و در کلاس مجزا نوشته اید، باید تغییرات لازم برای پیاده سازی این مورد را در منطق اصلی ایجاد کنید.

### لیدربورد (5)

نام بازیکن (در ابتدای بازی بگیرید)، برنده و امتیاز کل بازیکن ها و تایم بازی را ذخیره کنید

- منو بازی با گزینه‌ای برای وارد شدن به این بخش ایجاد شود
- لیدربورد با بازکردن دوباره بازی حذف نشود (ذخیره در فایل)

## ساخت دک (deck) (10)

بازیکن بتواند قبل از بازی، از کارت های موجود، 13 کارت را انتخاب کند.

- منو بازی با گزینه‌ای برای وارد شدن به این بخش ایجاد شود
- کارت های قبلی دک و کارت های قابل انتخاب را نمایش دهد
- دک با بازکردن دوباره بازی حذف نشود (ذخیره در فایل)

## اعلام رویدادها (10)

چاپ دقیق تمام اتفاقات بازی (شامل دمیج زدن، حذف شدن و ...) در زیر بورد بازی (بالای قسمت 6 در عکس) به جای چاپ شدن کارت وارد شده در بالای بورد بازی.

- منو بازی با گزینه‌ای برای وارد شدن به این بخش ایجاد شود
- تاریخچه رویدادهای آخرین بازی قابل دسترس باشد

## وسطچین کردن میزبازی (25)

در این حالت، میزبازی دقیقاً مطابق بازی اصلی در کنسول پیاده سازی خواهند شد. تمام خانه‌های بازی حذف شده، کارت‌ها همیشه به صورت دینامیک در وسط زمین نمایش داده می‌شوند و با افزودن هر کارت دوباره وسطچین خواهند شد.

- با توجه به حذف خانه ها، از ایندکس‌های جدید برای افزودن کارت‌ها به میزبازی استفاده کنید.
- پیاده سازی درست این ویژگی شرط دریافت نمره این بخش خواهد بود.

## رنگی کردن محیط بازی (10)

از رنگ‌های مناسب برای نمایش محیط بازی در کنسول استفاده کنید. برای مثال قدرت کاراکترهایی که از قدرت پیشفرض (قدرت قبل از damage یا boost) آنها بیشتر است با سبز و کمتر را با قرمز نشان دهید.

## نکات پیاده‌سازی پروژه

**ایده‌ها و خلاقیت:** هر گونه ایده و خلاقیت جدید در پروژه با هماهنگی با دستیاران آموزشی **نمره مثبت** به همراه دارد.

**زبان برنامه‌نویسی:** هیچ توجیهی برای استفاده از زبان‌های برنامه‌نویسی دیگر وجود نداشته و **نمره صفر** به پروژه تعلق می‌گیرد.

**اصول کدنویسی:** اصول شی گرای (وراثت، کپسوله‌سازی، چندریختی، انتزاع) به نسبت پیاده‌سازی رعایت شود. استفاده از نام، کامنت‌های مناسب و رعایت دندانه‌گذاری الزامی است.

**تسلط بر کد:** نمره پروژه به نسبت تسلط، پاسخگویی به سوالات و امکان تغییر کدهای نوشته شده داده خواهد شد. پروژه تنها به صورت انفرادی قابل انجام بوده و هرگونه تشابه بین دو کد یا عدم تسلط و پاسخگویی نسبی به سوالات منجر به **نمره منفی تا پنجاه درصد نمره پروژه** می‌شود. در این مورد سختگیرانه برخورد خواهد شد.

**قابلیت اجرا و استفاده از شی گرای:** پروژه باید قابلیت اجرا داشته باشد و از شی گرای استفاده کند. در غیر این صورت، نمره‌ای بین 0 تا 20 بر اساس کیفیت کد زده شده به پروژه تعلق می‌گیرد.

**کارکرد درست:** پیاده‌سازی اشتباه هر بخش می‌تواند باعث از دست دادن نمره کامل آن بخش شود. در صورت مشاهده باگ در حین ارائه، اگر در زمان ارائه قادر به دیباگ آن باشید، نمره منفی به شما تعلق نمی‌گیرد.

استاد سحر مکرمی – ترم ۱۴۰۲۱

جواد بشارتی‌فر – طرح شده با نظارت مهندس مهدی تروشه