Spesifikasi Tugas Besar 2
IF2210 Pemrograman Berorientasi Objek

Revisi

27 Maret 2020

Deskripsi

Aang lapar dan mau ke sadikin tapi sambil social distancing

Sebagai pemimpin dunia, Aang tahu kalau *social distancing* sangat penting untuk *flatten the curve*. Tapi ia paham kalau mengurung diri di rumah akan membuat semua orang bosan. Untuk itu, ia ingin menghibur penduduk dunia dengan game yang ia rancang, yang bernama **Avatar Duel**. Pada dasarnya, game ini mirip dengan Yu Gi Oh! dan Magic: The Gathering.

Kartu

Terdapat 3 jenis kartu, yakni karakter, land, dan skill.

Kartu Karakter

Kartu karakter merepresentasikan karakter yang bertanding dengan lawan. Suatu karakter memiliki **nama** dan **deskripsi**. Seorang karakter merupakan satu dari empat **elemen** api, air, tanah, atau angin. Seorang karakter memiliki nilai **attack** dan **defense** yang akan digunakan pada mekanisme bertarung. Untuk mengeluarkan seorang karakter, dibutuhkan **power** dengan jumlah dan elemen tertentu.

Pada saat karakter di arena, karakter bisa berada pada posisi menyerang atau bertahan.

Sebagai contoh: karakter bernama “Zuko” dengan elemen api, power 4, attack 5, dan defense 2. Artinya, dibutuhkan 4 power dari land api untuk mengeluarkan Zuko ke dalam arena.

Kartu Land

Kartu land merupakan sumber power dari pemain. Kartu land memiliki **nama**, **elemen**, dan **deskripsi**. Tiap giliran pemain, satu kartu land akan memberikan **tepat satu** power sesuai elemennya.

Kartu Skill

Kartu skill merepresentasikan kartu yang memengaruhi satu kartu karakter (baik karakter sendiri maupun lawan, elemen tidak harus sama dengan kartu skill). Suatu skill memiliki **nama**, **elemen**, dan **deskripsi**. Sama seperti karakter, untuk mengeluarkan skill, dibutuhkan **power** dengan jumlah dan elemen tertentu. Setelah dikeluarkan, kartu akan tetap di arena hingga karakter yang ditargetkan mati atau kartu dibuang oleh pemain.

Setiap skill memiliki **effect.** Effect yang mungkin:

1. Aura, karakter yang ditargetkan mendapat penambahan attack dan defense. Penambahan ini dapat bernilai 0 bahkan negatif. Kartu ini merupakan kartu paling banyak muncul.
2. Destroy, karakter yang ditargetkan mati.
3. Power Up, jika kartu yang ditargetkan menyerang karakter yang sedang dalam posisi bertahan, HP lawan juga akan berkurang seperti saat menyerang karakter dalam posisi menyerang.

Gameplay

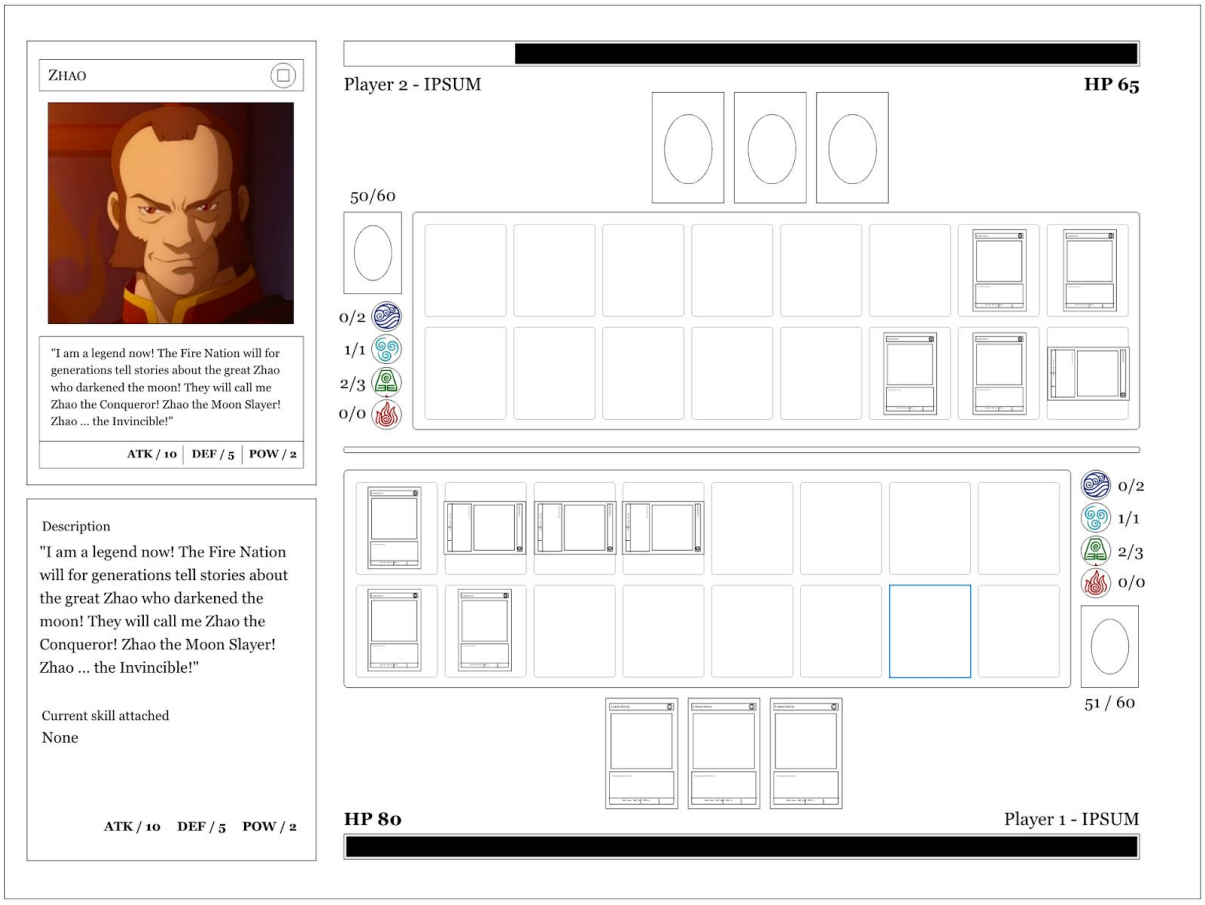
1. Permainan dimainkan oleh dua orang secara bergantian.
2. Setiap pemain akan diberikan sebuah *deck* berisi maksimal 60 kartu dan minimal 40 kartu.
3. Setiap permainan akan dimulai dengan kedua pemain memiliki 80 HP (Health Points).

4. Pada awal game, tiap pemain mengambil 7 kartu dari deck ke tangan.
5. Setiap giliran dibagi menjadi beberapa phase
    a. **Draw Phase**
        i. Pemain yang sedang bermain akan mengambil **satu** kartu dari *deck* milik pemain tersebut dan menaruhnya di tangan.
        ii. Power pemain di-reset sama dengan jumlah kartu land yang ia miliki. Power yang tersisa dari giliran sebelumnya tidak lagi dipakai.
    b. **Main Phase 1**. Pemain dapat melakukan beberapa aksi (atau tidak sama sekali) dalam Main Phase 1, yaitu:
        i. Meletakkan 0 atau lebih **kartu karakter**. Kartu karakter dapat ditaruh dalam posisi bertarung atau bertahan. Karakter yang baru saja diletakkan tidak dapat bertarung di battle phase.
        ii. Mengubah posisi dari kartu karakter yang ada pada field. Semua kartu karakter bisa diubah posisinya berkali-kali.
        iii. Meletakkan **maksimal satu kartu land**. Dalam 1 giliran, hanya 1 kartu land yang boleh diletakkan.
        iv. Meletakkan 0 atau lebih **kartu skill**, lalu memilih karakter yang ditargetkan.
        v. Membuang 0 atau lebih **kartu skill**.
    c. **Battle Phase**
        i. Pada phase ini, pemain dapat menggunakan karakternya untuk menyerang karakter lawan atau HP lawan.
        ii. Apabila ada karakter di arena lawan, pemain tidak bisa menyerang HP lawan secara langsung.
        iii. Setiap karakter hanya dapat menyerang maksimal satu kali.
        iv. Mekanisme pertarungan dapat dilihat di poin 6.
    d. **Main Phase 2**
        i. Semua aktivitas yang dapat dilakukan di Main Phase 1 dapat dilakukan juga di fase ini.
        ii. Karakter yang baru saja menyerang di battle phase tidak dapat diubah posisinya.
    e. **End Phase**
        i. Pemain mengakhiri giliran pada phase ini. Lawan akan memulai giliran-nya, dimulai dari draw phase, dan seterusnya.
6. Mekanisme Bertarung
    a. Karakter yang menyerang haruslah dalam posisi menyerang
    b. Jika karakter lawan sedang dalam posisi menyerang:
        i. **Attack** karakter lawan tidak boleh lebih dari attack karakter pemain.
        ii. Setelah menyerang, karakter lawan mati.
        iii. Selisih attack akan dikenakan ke HP lawan.
    c. Jika karakter lawan sedang dalam posisi bertahan:
        i. **Defense** karakter lawan tidak boleh lebih dari attack karakter pemain.
        ii. Setelah menyerang, karakter lawan mati.

iii. Tidak ada HP yang berkurang.

7. Pemain dinyatakan menang jika salah satu hal berikut terjadi:
    a. HP lawan mencapai 0 atau kurang
    b. Pada draw phase lawan, deck lawan kosong

Tampilan



Gambar di atas merupakan contoh kerangka dari tampilan dasar untuk game ini. Perhatikan bahwa:

- **Tampilan di atas merupakan contoh untuk membantu Anda saja. Anda tidak diwajibkan persis mengikuti. Selama semua aturan permainan terpenuhi, Anda boleh berkreasi sekreatif mungkin.**
- Layout tempat menaruh kartu terdiri dari 2 baris. Baris bagian atas untuk tempat karakter, baris bagian bawah untuk kartu skill. Kartu land yang diletakkan di arena tidak ditampilkan, melain hanya dituliskan di samping dengan format “x / y” untuk setiap elemen, dengan x merupakan power yang masih bisa dipakai di giliran ini dan y merupakan banyaknya kartu land yang ada.
- Anda tidak harus membuat ilustrasi untuk semua kartu.
- Asisten sudah memberikan beberapa kartu dan gambar (lihat bagian Panduan Pengerjaan), namun Anda boleh menambahkan kartu sendiri.
- Pada setiap giliran pemain, kartu tangan pemain tersebut akan dibuka dan kartu tangan pemain lawan akan ditutup, vice versa. Posisi layout tidak berubah.
- Setiap kartu bisa di-**hover** dan akan menampilkan info lengkap tentang kartu tersebut. Ketentuannya adalah sebagai berikut:
    - Semua kartu di arena dapat di-hover.
    - Kartu tangan hanya bisa di-hover oleh pemain bersangkutan pada saat gilirannya.
- Untuk memilih kartu (misal mau menyerang / mengaktifkan skill card), digunakan klik pada kartu tersebut. Terdapat indikator yang menandakan kartu tersebut sedang aktif / diklik oleh pemain, misal kartu diberi outline kuning.
- **Bonus:** game ini memiliki animasi

Tambahan

Berikut penjelasan tambahan untuk mudah membedakan game Avatar dengan Yu Gi Oh! dan Magic: The Gathering.

- Kartu karakter tidak pernah dalam kondisi tertutup.
- Tidak ada trap card (Yu-Gi-Oh!) atau Instant (Magic: The Gathering).
- Kartu land tidak sama dengan Field Spell Cards di Yu Gi Oh. Kartu Land di arena bisa ada lebih dari satu.
- Berbeda dengan game duel kartu lainnya, game ini dimainkan di satu laptop / komputer. Pada saat ganti pemain, layout permainan tidak perlu diputar, jadi memang ada pemain yang bermain di bagian atas dan ada pemain yang bermain di bagian bawah.

Panduan Pengerjaan

Dalam pengerjaan Tugas Besar 2, peserta mata kuliah diwajibkan menggunakan *framework* GUI **JavaFX** dengan menggunakan **JDK 8/Java 8**. Dalam Tugas Besar ini, peserta wajib menggunakan **gradle** sebagai *build tools*, dan dapat dijalankan menggunakan *command*

```
./gradlew
```

Gradle adalah sebuah alat yang memudahkan kita menyatakan *library-library* yang diperlukan sekaligus membantu kita melakukan *building* aplikasi. Gradlew merupakan sebuah *executable* gradle yang membuat sebuah mesin tidak perlu menginstall gradle terlebih dahulu untuk menjalankan aplikasi kita.

Berikut adalah repo yang dapat Anda *clone*. Di dalamnya, Anda dapat menemukan setup sederhana dari gradle dan file csv untuk kartu, lengkap dengan gambar ilustrasinya. Anda boleh mengubah kode atau struktur yang ada. Repo ini hanya contoh, bukan panduan.

Anda diwajibkan memanfaatkan prinsip-prinsip OOP berikut ini dalam aplikasi Anda :

- Inheritance
- Composition
- Interface
- Polymorphism
- Method Overriding
- Java API Collection
- Prinsip SOLID
- Design Pattern

Dalam pengerjaan, buatlah kode yang baik dengan menerapkan prinsip:

- DRY (Don't Repeat Yourself), tidak memiliki kode duplikat
- Memiliki struktur kelas yang mudah dipahami
- Dekomposisi yang baik dan implementasi yang tidak terlalu kompleks (sebuah method tidak terlalu panjang)
- Mengikuti konvensi penamaan yang baik

Panduan Laporan dan Dokumentasi

Sebagai programmer yang baik, Anda dilatih tidak hanya untuk membuat kode, tapi juga membuat dokumentasi dan tes.

Anda wajib membuat sebuah README (disarankan dalam markdown) berisikan setidak-tidaknya struktur kode, cara *compile*, cara *run*, dan *screenshot* aplikasi.

Anda juga perlu membuat tes untuk kode Anda. Lebih baik lagi, bila Anda dapat membagi tugas dan setiap kode diuji oleh orang yang berbeda. Tes yang baik akan mencoba semua kasus yang mungkin terjadi. Anda dapat membaca juga mengenai Unit Testing dan Integration Testing. Gunakanlah junit untuk memudahkan testing.

Revisi Avatar Duel

Dalam programming, program yang dibuat haruslah *maintainable*. Untuk menjaga *maintainability* kode, programmer dapat menggunakan prinsip SOLID, DRY, dan lainnya.

Karena kami percaya mahasiswa IF ITB mampu membuat kode yang *maintainable*, kami ingin membuat beberapa perubahan dalam spesifikasi tugas besar ini. Selama Anda mengubah kode Anda, tanyakan hal ini ke diri Anda:

- Berapa banyak kode yang saya ubah?
- Berapa banyak file yang berubah?
- Jika yang mengubah kode adalah orang lain, apakah ia akan tahu mana saja kode yang perlu diubah?
- Apakah kode saya sudah *maintainable* dan *readable*?
- Jika ada orang lain yang membaca kode saya, apakah ia akan langsung paham, atau perlu bertanya pada saya?
- Jika ada perubahan spesifikasi lagi, apakah kode saya sudah mudah berubah?

Element Energy

Nampaknya, di akhir Avatar: The Last Airbender, terdapat elemen baru, yakni *energybending*. Karena itu, tambahkanlah elemen Energy ke game Avatar Duel. Buatlah beberapa contoh karakter dengan elemen Energy.

Banyaknya karakter dan skill di arena

Pada spesifikasi sebelumnya, tidak dispesifikasikan banyaknya karakter dan skill maksimal di arena. Setelah berdiskusi dengan Product Manager game ini, diputuskan maksimal karakter dan skill di arena adalah 6.

Untuk kartu di tangan, tetap tidak ada spesifikasi jumlah kartu maksimal. Artinya, Anda boleh mendefinisikan jumlah kartu maksimal Anda sendiri.

Phase

Setelah diujikan ke beberapa *alpha tester*, banyak pengguna tidak pernah menggunakan Main Phase 2. Karena itu, phase ini dihilangkan. Setelah battle phase, game langsung lanjut ke End phase.

Dokumentasi

Berikanlah dokumentasi dari kode Anda. Dengan dokumentasi yang baik, programmer lain dapat membaca dokumentasi Anda dan paham cara melakukan modifikasi terhadap kode Anda, melakukan *debugging*, dan menggunakan kode Anda. Java menyediakan javadoc sebagai konvensi dokumentasi dan generator dokumentasi.

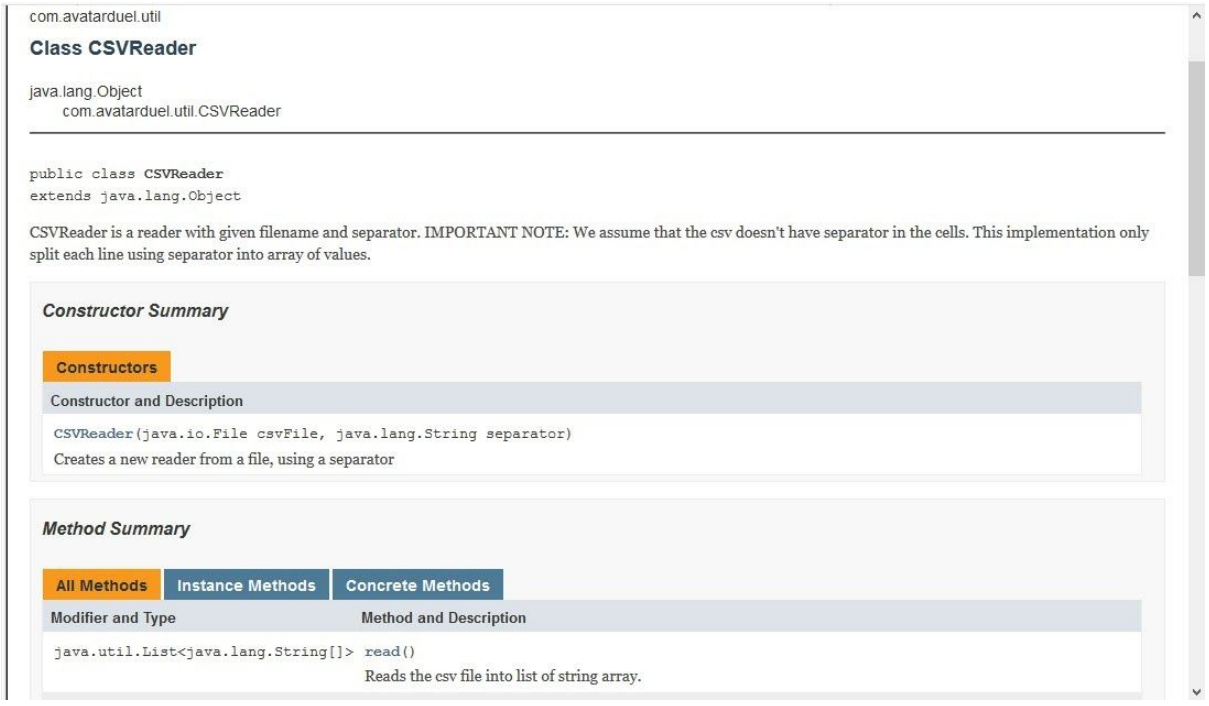
Tambahkan kode berikut di build.gradle:

```
task avatarDocs(type: Javadoc) {
  source = sourceSets.main.allJava
}
```

Untuk membuat dokumentasi, gunakan:

```
./gradlew avatarDocs
```

Hasil dokumentasi dapat diakses di ./build/docs/javadoc/index.html. Kelas CSVReader yang dibuat asisten sudah dibuat menggunakan konvensi java doc. Silakan coba buka com.avatarduel.util, lalu buka CSVReader. Anda harusnya dapat menemukan seperti ini:



Perhatikan kalau semua deskripsi diambil dari komentar yang ada di file CSVReader.java.

Tambahan

- Perhatikan bahwa paradigma prosedural pada Algoritma & Struktur Data tidak bisa dipakai lagi.
    - Anda tidak dapat membuat sekuens (meminta pilihan karakter -> meminta pilihan karakter lawan -> hitung pertarungan).
    - Dalam OOP, tiap objek berperilaku seperti objek. Mereka berkomunikasi dalam dunia OOP.
- Untuk pembuatan testing dan dokumentasi, tidak perlu sangat lengkap. Untuk diskusi mengenai seberapa dalam testing dan dokumentasi yang dibutuhkan, silakan diskusi dengan asisten pembimbing masing-masing.
- Berhati-hatilah dalam membuat kelas, jangan sampai membuat God Object.
- Sangat disarankan untuk menggunakan IDE, misalnya IntelliJ IDEA. Dengan email student, Anda bisa mendapat license versi Ultimate.
- Buatlah struktur yang baik dalam package program Anda. Ada banyak cara untuk menstrukturisasi package. Berikut 2 contoh cara menstrukturkan kode:
    - Berdasarkan role, sehingga ada package view, conroller, model, dll
    - Berdasarkan fungsionalitas / context, sehingga ada package card, player, dll
- Update file .gitignore di project kalian, folder seperti `out`, `build`, `bin`, `.gradle`, `.settings`, `.classpath` tidak perlu dimasukkan. Jika sudah terlanjur dicommit, hapus dengan command `git rm`.
- Jangan sampai penggunaan fxml membuat keterbatasan dalam kode Anda. Tetap buatlah kelas yang modular dan Single Responsible. Anda bisa juga mengextend beberapa komponen seperti HBox, Pane, dan lainnya. Controller yang Anda buat seharusnya tidak perlu sampai 200+ baris.
- Sebisa mungkin gunakan standar penulisan kode di java. Google sudah membuat guide yang baik untuk diikuti. Beberapa yang dasar:
    - Nama package hanya terdiri dari huruf kecil, tanpa simbol
        - Contoh baik: com.avatarduel.card, com.avatarduel.controller
        - Contoh jelek: com.avatarduel.GUI, com.avatarduel.Fxml
    - Nama class ditulis dengan UpperCamelCase, merupakan kata benda
        - Contoh baik: Card, Character
        - Contoh buruk: SKILL, Attack
    - Nama method ditulis dengan lowerCamelCase, dimulai dengan kata kerja
        - Contoh baik: getSize(), draw()
        - Contoh buruk: GetAttack(), position()
    - Gunakan pilihan bahasa yang konsisten. Jangan mencampuradukkan pilihan bahasa.
- Deadline pengerjaan tugas besar diundur menjadi 20 April 2020 pukul 08.00 WIB.