



دانشگاه بوعلی سینا

فاز یک پروژه پیشرفته

متین امیرپناه فر

شماره دانشجویی: ۴۰۲۱۲۳۵۸۰۰۳

نیما مخملی

شماره دانشجویی: ۴۰۲۱۲۳۵۸۰۳۵

۱۵ مرداد ۱۴۰۳

فهرست مطالب

۱	اول نحوه بازی	
۳	دوم کلاس ها	
۴	بازی	۱.۰
۶	رابط کاربری	۲.۰
۸	رابط گرافیکی	۳.۰
۱۲	زمین بازی	۴.۰
۱۳	کارت ها	۵.۰
۱۴	کارت های بنفش (ویژه)	۱.۵.۰
۱۶	فصل ها	۲.۵.۰
۱۷	کارت های زرد (سرباز)	۳.۵.۰
۱۹	نشان	۶.۰
۱۹	نشان بازیکن	۱.۶.۰
۲۰	ایالت	۲.۶.۰
۲۰	بازیکن	۷.۰

۲۲

سوم چالش ها

۲۴

چهارم پیوندها

۲۵

گیت هاب

۸۰

۲۶


پنجم منابع



چکیده


در دورانی که قلمرو شاهی به استان های خود مختار تقسیم شده، سردارانی از گوشه ی سرزمین در سودای کشورگشایی، ارتش خود را برای فتح شهرها گسیل داشته اند. تا با مهارت و شجاعت بجنگند و با تدبیر و مکر به حریفان ركب بزنند تا قدرت و افتخار كسب كنند

بازی ركب يك بازی تخته ای است كه توسط ۳ تا ۶ بازیكن انجام میشود این بازی شامل دو نوع كارت بنفش و زرد است كه هر كارت بنفش امتیاز خاص خود را دارد ولی كارت های زرد ویژگی خاصی ندارند كارت های بنفش عبارتند از : مترسك ، طبل زن ، شاهدخت ، شیرزن ، بهار ، زمستان ، ریش سفید بازیكنان باید با كارت هایی كه در دست دارند به جنگ پردازند و سعی كنند در جنگ پیروز شوند و اگر سه استان مجاور یا پنج استان را فتح كنند پیروز كل بازی خواهند شد ما این بازی را به صورت شی گرانوشته ایم كه شرح كلاس ها و قابلیت های آنها را در ادامه برایتان میگوییم

A knight in full plate armor is depicted riding a dark horse. The knight holds a long spear diagonally across the frame. The background is a bright, hazy sky with a strong light source on the right, creating a silhouette effect on the knight and horse. The overall tone is dramatic and historical.

بخش اول نحوه بازی

هنگامی که بازی را آغاز مینماییم در ورودی از کاربر میخواهد که تعداد بازیکنان را وارد کند. سپس هر بازیکن مشخصات خود را وارد میکند و بعد رنگ نشان خود را انتخاب میکنند. بعد از آن سیستم چک میکند و کم سن ترین بازیکن را به عنوان اولین بازی کن که جنگ اول را شروع میکند برمیگزیند. بازیکن مورد نظر استان محل اولین جنگ را انتخاب میکند و سپس سیستم کارت های بازیکنان را به صورت رندوم به آنها میدهد. در خروجی برنامه از بازیکنان میخواهد که به صورت خصوصی کارت های خود را مشاهده کنند و به ترتیب این عمل انجام میشود آنگاه بازی توسط بازیکن کم سن آغاز میشود و به ترتیب سن بازیکن ها کارت های خود را رو میکنند بازیکنان میتوانند کارت هارا رو کنند تا بالاخره فاتح استان مورد جنگ معلوم شود و به همین ترتیب بازی را ادامه میدهند تا یک بازیکن ۳ استان مجاور یا ۵ استان از کل نقشه را فتح کند.



بخش دوم کلاس ها

۱.۰ بازی

این کلاس وظیفه اصلی و مدیریت کلی بازی را بر عهده دارد شرایط بازی را آماده و بازیکنان را مجهز و آماده جنگ میکند این کلاس با کلاس های بازیکن ، زمین بازی ، رابط کاربری ، تمام کلاس های کارت ها و تمام کلاس های نشان رابطه دارد

روابط این کلاس با کلاس های بازیکن ، زمین بازی ، تمام کارت ها ، تمام نشان ها و رابط کاربری رابطه دارد. رابط بین این کلاس ها همه از نوع composition میباشد.

صفات صفات این بازی عبارتند از آرایه ای از تمام کارت های بازی مانند آرایه هایی از کارت های بنفش با تعداد مختلف همچنین اشیاء مورد نیاز این کلاس از کلاس هایی مانند

- ui
- gameboard
- battlemarker
- player

متدها

- play این تابع وظایف شروع و مدیریت بازی را دارد
- gethelp برای ارسال متن راهنمایی است
- check_number_of_player چک کردن

- `set_battleground` ست کردن محل جنگ
- `find_war_winner` پیدا کردن برنده جنگ
- `find_game_winner` پیدا کردن برنده کل بازی
- `getPlayers` ارسال بازیکنان البته اطلاعات بازیکنان
- `shuffle` مخلوط کردن کارت ها
- `distributeCards` پخش کارت ها
- `war` این تابع برای مدیریت جنگ است

۲.۰ رابط کاربری

این کلاس وظیفه برقراری ارتباط سیستم و کاربران را دارد دستورات را از ورودی میگیرد و گزارشات را در خروجی برای رویت کردن کاربران چاپ میکند
این کلاس صفتی که به شکل خصوصی باشد ندارد

روابط

متدها متدها یا رفتارهای این کلاس عبارتند از

- `clearTerminal` پاک کردن تصویر
- `pause` توقف چند لحظه ای
- `spliter` خوشگلاسیون
- `showPlayerPlayedCards` نمایش کارت های بازی شده بازیکنان
- `getCommand` برای انتخاب ویژگی های خاص بازی من جمله پاس دادن
- `get_battleground` مکان جنگ
- `showPlayerStates` نمایش استان های فتح شده بازیکن
- `showPlayerCards` نمایش کارت های در دست بازیکن
- `get_card_name` ارسال نام کارت ها

- `get_players_number` ارسال شماره بازیکن البته به ترتیب سن
- `get_player_name` ارسال نام بازیکن
- `get_player_old` ارسال نام بازیکن
- `get_player_color` ارسال رنگ بازیکن
- `declare_warWinner` برنده جنگ
- `declare_gameWinner` برنده کل بازی

۳.۰ رابط گرافیکی

قسمت گرافیکی این نرم افزار به وسیله محیط QT طراحی شده است.

ما از ورژن QT6_widget استفاده کردیم و برنامه را به صورت شی گرا برای کاربران به نمایش میگذاریم.

کلاس ها قسمت گرافیکی این پروژه از ۱۰ کلاس تشکیل شده است

- Campaign این صفحه اصلی ترین قسمت بازی هست جایی است که در بازی انجام میشود و امپراطور ها بر سر سرزمین با هم میجنگند . در این صفحه نوارهایی برای قرار گیری کارت بازیکنان وجود دارد ، نقشه اصلی بازی وجود دارد و بر روی هر استان یک لیبل گذاشته شده تا بازیکنان بتوانند نشان های مورد نیاز را روی نقشه بگذارند.
- هنگامی که نوبت هر بازیکن میشود کارت های او در پایین صفحه نمایش داده میشوند تا از میان آنها یک کارت را انتخاب کند و یا دکمه پاس دادن را بزند.

- gameWinner در این صفحه یک متن قرار گرفته برای نمایش نام بازیکنی که برنده کل بازی میشود به این معنی که ۵ استان یا ۳ استان مجاور را فتح کرده است. در این صفحه دو دکمه گذاشته شده برای خروج از بازی یا برگشتن به منوی اصلی .

● **Help** همانگونه که قبلاً گفته شد برای هر شخصیت بازی و یا کلیت بازی یک متن آماده شده و بازیکنان میتوانند در فرایند بازی درخواست کمک کنند تا توضیحات برایشان نمایش داده شود پس هنگامی که بازیکن دکمه کمک را میزند به این صفحه می آید تا توضیحات درباره ی هر چیزی که میخواهد برایش نمایش داده شود.

● **locateinfo** در این صفحه نشان جنگ گذاشته میشود یعنی استان محل جنگ توسط کم سن ترین بازیکن انتخاب میشود در این صفحه یک شماتیک از نقشه بازی وجود دارد و بازیکن نام استان مورد نظر را انتخاب میکند.

● **mainwindow** این کلاس اولین کلاسی هست که به نمایش گذاشته میشود و منوی بازی ما محسوب میشود و شامل سه دکمه ی آغاز بازی ، بازی های ذخیره شده و خروج از بازی میشود همچنین امکان قطع صدای بازی را هم داریم .

صفات و ویژگی ها **back_end** این کلاس ما چندین شیء به صورت اشاره گری از کلاس های دیگر داریم که روابط را برای ما ایجاد میکنند این اشیاء عبارتند از :

- Start_*start
- Campaign_*campaign
- Game_*game

- Locateinfo_*battleground_page
- Locateinfo_*favorground_page

در قسمت همچنین از کلاس های پیش فرض خود نرم افزار کیوتی استفاده کرده ایم برای پخش صدای بازی که نام این کلاس ها عبارتند از

QMediaPlayer، QAudioOutput.

توابع فعال این کلاس اول کانستراکتور آن هست که بیشتر فرایند هارا انجام میدهد و دوم تابع کلید صدای آن است که صدای موسیقی در جریان بازی را کم و زیاد میکند.

● options در تمام بازی های دنیا یک صفحه برای تنظیمات تهیه میشود که در فرایند اصلی بازی اگر بازیکن نیاز به تغییرات تنظیمات را داشت بتواند کارش را انجام دهد یا از فرایند خارج شود پس ما هم تصمیم گرفتیم تا صفحه ای طراحی کنیم که این کار را بکند و در آن کلید هایی با عنوان ذخیره بازی ، خروج از بازی ، بازی های لود شده و ... گذاشته ایم هنگامی که بازی کن بر روی چرخ دنده داخل صفحه اصلی بزند به این صفحه آورده میشود.

● setplayer در این صفحه بازیکنان نام سن و رنگ نشان خود را انتخاب وارد میکنند تا اطلاعات آنها ذخیره شود و بازی را شروع کنند.

• showcards در این صفحه کارت هایی که به صورت شانس به بازیکنان داده میشود را به ترتیب سن از کوچک به بزرگ نمایش میدهد.

• start

فرایند اصلی بازی از این صفحه آغاز میشود که در اینجا تعداد نفراتی که بازی میکنند گرفته میشود

صفات و ویژگی ها در این کلاس برای ارتباط با بیرون از خود شی از کلاس های دیگر دارد که به شرح زیر است:

Setplayer_*setplayer

دو متغیر خصوصی نیز با نام های playerID، playersNumber و یک نوع داده ای مپ از رنگ ها داریم برای نشان بازیکنان

std::map<QString, Color>_freeColors

توابع فعال نیز دو تابع هستند که یکی برای دکمه بعدی در داخل صفحه هست و دیگری برای اطلاعات بازیکنان get_player

• warWinner در این صفحه یک متن چاپ میشود که در آن نوشته شده کدام بازیکن برنده آن تک جنگ است و شماره آن بازیکن را نمایش میدهد.

۴.۰ زمین بازی

این کلاس ارتباط بین استان هارا معلوم کرده و در واقع همان صفحه کاغذی موجود در بازی است
این کلاس برای ما استان های بازی را نگهداری کرده و ارتباط بین آنها را ایجاد میکند و مشخص میکند که آیا دو استان با هم مجاورند یا خیر

روابط این کلاس با کلاس ایالت رابطه دارد که این رابطه نوعی رابطه وابستگی را بین آنها شرح میدهد.

صفات صفات خصوصی این کلاس استان های بازی هستند که شامل ۱۴ استان میشوند این استان ها با استفاده از `unordered_map` و کلاس استان گرفته شده اند و صفت بعدی ارتباط بین این استان هاست که برای این کار از کلاس `map` استفاده کرده ایم

متدها متدهای این کلاس عبارتند از :

- `checkAdjacency` برای چک کردن
- `getState` ارسال استان های بازی
- `get_active_states_name` ارسال نام استان فعال و منتخب

۵.۰ کارت ها

این کلاس مسئول نگهداری و سازماندهی کارت هاست این کلاس دو زیر کلاس دارد که بیانگر دو نوع کارت بنفش و زرد ایست که در بازی داریم

روابط این کلاس با کلاس های بازی و بازیکن رابطه دارد این روابط از نوع association و composition است.

صفات صفات کلاس کارت ها عبارتند از

۱. امتیاز که این کلاس چون دو کلاس از آن ارث بری میکنند صفت امتیاز را به صورت protected نوشته ایم

۲. ویژگی اولویت که به صورت خصوصی است

متدها توابع که برای این کلاس تعریف کرده ایم:

- getType ارسال نوع کارت مدنظر
- applyFeature ویژگی کارت های ویژه یا بنفش که برای هر کارت به صورت جداگانه تعریف شده است
- is_season چک کردن اینکه آیا کارت فصل است یا خیر؟
- getPriority ارسال اولویت کارت برای امور بازی
- setPoint قرار دادن امتیاز
- getPoint ارسال امتیاز مربوط به کارت

۱.۵.۰ کارت های بنفش (ویژه)

این کلاس برای کارت های ویژه این بازی طراحی شده است که تمام کلاس های شخصیت های کلاس بنفش از این کلاس ارث بری میکنند کلاس هایی همچون

• bishop

• drummer

• heroine

• scarecrow

• spring

• spy

• turncoat

• winter

صفات این کلاس فقط یک صفت دارد که آن هم برای type کارتهاست

روابط تک تک کارت های بنفش یا شخصیت ها با کلاس کارت بنفش رابطه ارث بری دارند به همین جهت در شرح تک تک کلاس های ویژه دیگر بند روابط را نداریم. همچنین خود کلاس کارت های بنفش هم با کلاس کارت رابطه ارث بری دارد.

متدها همچنین این کلاس فقط یک تابع برای ارسال `type` دارد که اسم آن `getType` است

مترسک

کلاس شخصیت مترسک : این کارت وقتی در بازی به کار برده میشود بازیکن میتواند یک کارت زرد را به بازی برگرداند این کارت روی کارت های بنفش تاثیری ندارد

صفات تنها صفت این کلاس صفت `help` است که موقعی به کار برده میشود که کاربر بخواهد توضیحات مربوط به کارت را ببیند و با کارت آشنا شود

متدها توابع این کلاس عبارتند از `gethelp` که برای ارسال متن توضیحات است و `applyFeature` که مربوط به ویژگی کارت است

طبل زن

کلاس شخصیت طبل زن : این کارت وقتی توسط یک بازیکن رو میشود ارزش تمام کارت های زردی که آن بازیکن در یک درست آورده دو برابر میشود. البته فقط یکبار امکان پذیر است.

صفات تنها صفت این کلاس صفت `help` است که موقعی به کار برده میشود که کاربر بخواهد توضیحات مربوط به کارت را ببیند و با کارت آشنا شود

متدها توابع این کلاس عبارتند از `gethelp` که برای ارسال متن توضیحات است و `applyFeature` که مربوط به ویژگی کارت است

شاهدخت

کلاس شخصیت شاهدخت : کارت شاهدخت یک کارت امتیازیه که هیچ کارت دیگری روی آن اثر ندارد.

صفات این کلاس فقط یک صفت دارد به نام `help` که برای نمایش توضیحات این کارت است

متدها دو متد این کلاس عبارت است از :

- `gethelp` برای ارسال متن توضیحات کارت است که کاربر درخواست کمک کرده است.

- `applyFeature` این تابع نیز برای اعمال ویژگی کارت است

۲.۵.۰ فصل ها

این کلاس برای فصل های بهار و زمستان که دو کارت ویژه هستند طراحی شده است که همان طور که معلوم است دو کلاس بهار و زمستان از آن ارث بری میکنند

صفات این کلاس هیچ عضو داده ای خصوصی ندارد.

متدها تنها تابع عضو داده ای این کلاس تابع `is_season` است

که این تابع نیز برای چک کردن فصل بودن است که قبلا توضیح دادیم

بهار

کلاس شخصیت بهار: تا زمانی که بهار باشد به مجموع کارت های رو شده بازیکن یا بازیکنانی که بالا ترین کارت عددی را رو کرده اند یا در آینده بازی میکنند ۳ واحد اضافه میکند.

صفات این کلاس هم مانند تمام کلاس های کارت بنفش فقط یک صفت help دارد

متدها applyFeature، gethelp
توضیحات این توابع را در کلاس های قبلی گفته ایم.

زمستان

کارت شخصیت زمستان: تا زمانی که زمستان باشد عدد تمام کارت های زرد یا سرباز همه بازیکنان ۱ میشود اگر بهار باشد و زمستان بیاید، زمستان میشود

صفات help

متدها applyFeature، gethelp
مانند دیگر کلاس های کارت های ویژه

۳.۵.۰ کارت های زرد(سرباز)

کارت های زرد یا سرباز موقعی که رو میشوند قدرت ارتش بازیکن را به اندازه عدد روی کارت افزایش میدهد.

صفات هیچ صفتی ندارد

متدها این کلاس هم دو تابع `gethelp` ، `applyFeature` را دارا میباشد

۶.۰ نشان

کلاس نشان یک کلاس پدر است که سه کلاس فرزند دارد این کلاس را از آن جهت طراحی کرده ایم که نشان های بازیکن و جنگ و صلح را پیاده سازی کنیم

صفات صفات این کلاس : ۱- یک شی از کلاس State برای اعمال نشان جنگ ۲- یک شی از کلاس Color برای پیاده سازی نشان بازیکنان با رنگ های متفاوت

متدها توابع عضو این کلاس :

- setState قرار دادن استان مدنظر
- getState ارسال استان مدنظر
- is_set برای چک کردن که آیا استان منتخب است؟

۱.۶.۰ نشان بازیکن

این کلاس را برای نشان بازیکن طراحی کرده ایم تا بازیکنان به هنگام آغاز بازی یک نشان برای خود انتخاب کنند و وقتی یک استان را فتح کردند آن نشان روی استان مورد نظر قرار گیرد

روابط این کلاس با کلاس نشان رابطه ارث بری دارد

روابط این کلاس با کلاس رنگ رابطه دارد برای ست کردن رنگ نشان هر بازیکن.

صفات ندارد

متدها ندارد

۲.۶.۰ ایالت

کلاس ایالت یا استان به این منظور طراحی شده که فرایندی که لازم است مستقیم با استان ها انجام شود راحت تر صورت بگیرند

صفات برای این کلاس دو صفت در نظر گرفته شده است

- name
- set_marker

متدها

۷.۰ بازیکن


این کلاس به منظور مدیریت بازیکنان و تمام رفتارها و ویژگی های آنها طراحی شده است این کلاس با کلاس های کارت استان و نشان بازیکن و رابط کاربری رابطه دارد

صفات ویژگی های خصوصی کلاس بازیکنان شامل نام ، سن ، آی دی ، وکتوری از کارت های در اختیار، وکتوری از کارت های بازی شده ، استان های فتح شده و آرایه ای از نشان بازیکنان.

روابط کلاس بازیکن با کلاس های کارت ، استان ، و نشان و نشان بازیکن رابطه دارد.

متدها توابع عضو این کلاس شامل توابع زیر میشوند :

- getName برای ارسال نام بازیکنان
- getID برای ارسال شماره بازیکنان
- getAge برای ارسال سن بازیکنان
- getPlayedCards برای ارسال کارت های بازی شده بازیکن
- getCards برای ارسال کارت های در دست بازیکنان
- setCards قرار دادن کارت ها
- setState قرار دادن استان مدنظر
- get_states_name ارسال نام استان
- drawn_card کشیدن کارت برای بازی
- push_to_cards افزودن کارت به کارت های در دست بازیکن
- push_to_playedCards افزودن به کارت های مصرف شده بازیکن
- drawn_playedCard کشیدن کارت های بازی شده



بخش سوم چالش ها

چالش هایی که در برنامه نویسی این بازی با آنها مواجه شدیم عبارتند از

۱. پیاده سازی مجاورت بودن استان های بازی که از کلاس map استفاده کردیم


۲. پیاده سازی ماهیت استان ها با استفاده از unordered_map

۳. کامپایل کردن و اجرای برنامه توسط CMAKE (اجرای برنامه ای با چندین فایل)

۴. برقراری ارتباط بین کلاس ها ایجاد بستری برای اجرای برنامه توسط رابط گرافیکی QT

۵. تعیین نوع رابطه موجود بین کلاس ها برای اینکه برنامه از لحاظ شی گرای بی نقص باشد.


۶. رسم نمودار class diagram این برنامه



بخش چهارم پیوندها

۸۰ گیت هاب

لینک شما میتوانید برنامه ما را در این پیوند مشاهده بفرمایید



بخش پنجم

منابع

• کتاب برنامه نویسی به زبان سی پلاس پلاس اثر دایتل و دایتل
ویرایش دهم

• www.geeksforgeeks.org/c-plus-plus