با سلام و خسته نباشید...
در این پروژه من یک کلاس به اسم Game درست کردم که در آن با توابع initwindow و inisupportedkeys و initstates به ترتیب window که یک renderwindow است و supportedkeys که مپی است و stackای از کلاس مجازی state است. که هر سه این توابع در سازنده این کلاس فراخوانی میشوند.سپس در تابع اصلی برنامه یک شئ از این کلاس درست کرده و تابع run ان رو فراخوانی میکنیم که در ان در یک while که همیشه درست است تازمانی که پنجره باز باشد این ادامه دارد...در این while سه تابع همین کلاس فراخوانی میشود که عبارتند از updatedt و update و rende که به ترتیب شئ ای از کلاس Clock را resetart میکند و در تابع دومی update بالای استک مورد نظر را فراخوانی میکند و در قسمت تابع سومی render بالای استک مورد نظر را فراخوانی میکند(توجه داشته باشید که در کلاس state که یک کلاس مجازی است دو تابع render&update هر دو pure virtual هستند و در این قسمت یعنی ما از رفتار چند ریختی برای نمیاش صفحات بازی استفاده کردیم.
از کلاس مجازی state دو کلاس mainmenu و gamestate ارث برده است به ترتیب منو بازیست و صفحه چیدن هیروهای بازیست.
درکلاس state یک تابع به اسم updatemouseposition داریم با سه متغیر که هر کدام رو get کرده و در داخل ان ها میریزیم...
طبق گفته ارث بریمان این تابع با سه متغیرشان برای آن دو کلاس هم ارث برده میشوند و دائم فراخوانی میشود با متغیر clock ای که در کلاس game تعریف کردیم.
در کلاس mainmenu که از state ارث برده است من ابتدا دو تابع render و update بازنویسی کردم چون در آنجا مساوی با صفر بود.از طرفی یک کلاس به اسم button من درست کردم که در سازنده خود نه متغیر میگیرد که عبارتند از موقعیت طولی و عرضی و سایز آن و رنگ های آن.در کلاس mainmenu من سه شئ از این کلاس ساخته ام یعنی سه تا دکمه برای صفحه منوی بازیم که به ترتیب NewGame-About-Quit

هستند.در کلاس صفحه منوام یه background درست کردم دو در سازنده خود این کلاس با تابع initbackground ان را مقدار دهی کرده و از دایرکتوری مورد نظر load اش میکنم.
در تابع update این کلاس من سه تابع را فراخوانی میکنم که به ترتیب عبارتند از updatemousposition و updatebutton.
در این کلاس منوی بازی من یه شئ از فونت ساخته و با تابع initFont ان را مقدار دهی کرده و این تابع را در سازنده این کلاس فراخوانی میکنم.
در قسمت render این کلاس دو خط کد وجود دارد که عبارتند از نشان دادن این background و فراخوانی تابع renderbuttons که در آن دکمه هارا در همین صفحه نمایش میدهد.
دقت کنین که من سایز صفحه ی باز شده(پنجره)را به background میدهم پس این عکس دقیق اندازه پنجره است.
در سازنده این کلاس به جز دو تابعی که برای مقدار دهی پس زمینه و فونت فراخوانی میشود یک تابع دیگه به اسم initbuttons هم فراخوانی میشود که این دکمه هارا new میکند در یک مپی از جنس *Button.
در مخرب این کلاس هم چون ما این دکمه هارا در یک مپی new کردیم پس موظفیم که در این قسمت آن هارا delete کنیم.پس با یک for و iteretor این مپ را خالی میکنیم.
در قسمت updatebutton این کلاس سه تا شرط وجود دارد که به این صورت هستند که اگه روی newgame کلیک شود یک gamestate در استک state کلاس اصلیمان new میکنیم.اگر روی Quit کلیک شود تایع endstate همین کلاس که باز نویسی شده از کلاس مجازی state فراخوانی میشود.
حال در قسمتی که کاربر روی دکمه newgame کلیک میکند یک gamestate نیو میشود و push میشود در استک مورد نظر ما.
ازآن پس update این gamestate فراخوانی میشود چون الان در بالاترین قسمت این استک این gamestate قرار دارد و الان استک ما اندازه اش دو میشود.

قبل از این که وارد کلاس gamestate بشویم ابتدا کلاس entity را توضیح بدهم.این کلاس کلاسی است که آیکون های هیرو از ان ارث میبرند توجه کنین که این کلاس خلاف کلاس state مجازی نیست یعنی میشود از این شئ ساخت.

در این کلاس دو متغیر enum داریم یکی از ان ها برای رنگ آیکون هاست نسبت به موقعیت صفحه کلیک تغییر رنگ بدهد و دیگری برای اینه که چه هیروییست که از این کلاس ارث برده است یعنی در سازنده تمام اون ده آیکون ها نسبت به هرکدام یکی از ان ها در این متغیر ریخته میشود که اولیه ان giant است.

در قسمت private این کلاس دو شی از این دو enum میسازیم و در سازنده این کلاس مقدار دهی میکنیم.

سازنده این کلاس دو عدد اعشاری برای موقعیت sprite و یک رشته میگیرد برای اینکه چه عکسی را از فایل load کند...یعنی اگر giant بود عکس ان را بارگیری کند.

در ده کلاسی که از این کلاس ارث برده شده فقط یک چیز اضافه میکنیم آن هم این است که در سازنده این ده کلاس متناظر با اسمشان مقدار درست enum را در آن متغیری که بالا عرض کردیم میریزیم.

حال برمیگردیم به کلاس gamestate که در ان یک vector از جنس * Entity تعریف میکنیم و در تابعی به اسم initEntities ان را پر میکنیم به تعدا هیرو ها یعنی ده تا هیرو دز ان new کرده و این تابع را در سازنده این کلاس فراخوانی کرده و جون new کردیم باید در مخرب این کلاس همه را delete کنیم که همین کاررا هم کردیم در gamestat~.

در سازنده کلاس gamestate یک تابع هم فراخوانی میکنیم آن هم این کار را میکند که زمین بازی را درست میکند یعنی دو زمین ۹×۹ درست میکند که هرخانه‌ی آن شئ ای از کلاس Tileهستش که این کلاس هم دست نویس خودم است مثل بقیه کلاس هایی که تا الان اسم انهارا بردم...

این کلاس در خود یک Texture و Sprite دارد که در سازنده این کلاس تکسچر مورد نظر را از فایل لود میکند و روی اسپریت میگذار(همان خونه های سبز هستند که در صفحه بازی میبینید)
ما در کلاس gamestate از این کلاس دوتا زمین ۹×۹ ساخته و بهشون موقعیت میدهیم و تابع نمایش ان هارا فراخوانی میکنیم...
برای موقعیت دادن از متغیر های static استفاده کردم که بقل هم چیده شوند.
در این کلاس (gamestate) دو تا وکتور داریم از جنس *
heroabstractclass که همان پنج هیرو ایست که دو بازیکن انتخاب میکنن...
پس من چهار تا تابع در update این کلاس نوشتم که به ترتیب هرکدام صدا زده میشوند...
اولی ان ها تابعی است که بازیکن اول پنج تا هیروی خودش را انتخاب میکند سپس تابع دوم این است که پنج تا از خانه های زمین را انتخاب میکند سپس همین دو تابع برای بازیکن دوم هم پیاده سازی شده و یکی پس از دیگری فرخوانی میشود...
پس از اتمام انتخاب هیرو های متناسب با هرکدام از ان هیرو در ان دو وکتوری که ذکر کردم new شده و سپس push میشود.
پس از چیدن هیروهای بازیکن دوم در زمین توسط کاربر دوم بازی شروع میشود و تابع play در فراخوانی میشود.
الان یه توضیحی درباره heroabstractclass میدهم...
این کلاس سه متغیر دارد که یکی از ان ها بولین است و دوتای دیگر خون و قدرت هیرو است که از جنس عدد صحیح است...
دوتابع attack و setattribute هردو مساوی صفر است پس از این کلاس نمیشود شئ ساخت سر همین قضیه من از این کلاس دو تا وکتور اشاره گر میسازم چون میخواهم رفتار چند ریختی عین استک استیت ها اجرا کنم...
سر همین این دوتا تابع را بازنویسی کردم در تمام ده هیرو و متناسب با هرکدام ان ها در تابع setattribute که در سازنده فراخوانی میشود ان سه متغیر را مقدار دهی میکنم متناسب با چه هیرویی بودنش...

و در تابع attack که هنوز پیاده سازی نکردم باید چگونگی attack هر هیرو رو پیاده سازی کنم
درکلاس هیرو مجازی چند تابع داریم که هرکدام یک کار خاصی را انجام میدهند...
دوتا از ان ها قدرت و خون هر هیرو را برمیگرداند...
یکی از ان ها چک میکند که آیا خون هیرو از صفر کوچک تر است یا نه (برای زنده بودن یا نبودن)
یکی از ان ها دو عدد اعشاری برای موقعیت اسپریت میگیرد و سازنده به جز این دو عدد اعشاری و یک رشته میگیرد برای عکس مورد متناظر با ان برای لود کردنش...
در کلاس gamestate بعد از چیدن هیرو ها توسط بازیکن دوم تابع play فراخوانی میشود و باید یک هیرو را انتخاب کند و شروع به بازی کند...
در قسمت aboutstate که از کلاس مجازی state ارث برده است با زدن دکمه about یک صفحه ساخته میشود و آیکون هیرو ها با اسمهایشان را میبینیم در صفحه ی about.
برای هیرو ها تابع attack را پیاده سازی کردم فقط این تابع شلیک معمولی است یعنی همانی که یک هیرو و شماره ی آن خانه را گرفته و بعد به اندازه قدرت ان هیرو از اون هیروی مفعول کم میکند...