

نسیم کوهستانی
شماره دانشجویی : ۴۰۱۱۳۴۲۵

کلاس GridOfGame:

کانستراکتور (__init__):

در متغیر grid که در بخش کانستراکتور تشکیل دادیم برای ساخت جدول ۴ در ۴ میباشد
در متغیر score امتیاز حال حاضر بازیکن ذخیره میشود
در متغیر high_score مقدار بیشترین امتیازی که بازیکن گرفته است ذخیره میشود
در متغیر hight ارتفاع صفحه ذخیره میشود
در متغیر width عرض صفحه ذخیره میشود
در متغیر REPEAT وضعیت دوباره شروع بازی ذخیره میشود
سپس در ادامه با استفاده از متود های pygame صفحه مد نظر با مشخصات خواسته شده را میسازیم و
فونت دلخواهمان را مشخص میکنیم و از فایل txt در صورت وجود مقدار high_score را بدست می
آوریم

تابع new_num:

این تابع عدد جدید تولید میکند و به احتمال ۱۰ درصد عدد ۴ است وگرنه ۲ میباشد

تابع check_for_sum:

این تابع چک میکند که بعد از حرکت های مان بر روی جدول اگر لازم بود خانه هایی از سطر یا
ستون داده شده جمع شود یا نه و اگر قابل جمع بود جمع میکند

تابع move:

این تابع نسبت به حرکت ها که L یا R یا U یا D میباشد تغییرات لازمه را بر روی جدول پیاده سازی
میکند

تابع score_board:

این تابع score و high_score را برای بازیکن با استفاده از توابع pygame نمایش میدهد و بعد از
هر حرکت مقدار score آپدیت میشود و اگر high_score را رد کردیم مقدار high_score هم نیز
عوض میشود.

تابع game_page:

این تابع نسبت به ماتریس ۴ در ۴ grid و با استفاده از رنگ هایی که در فایل const.py به ازای هر
عدد تعریف شده است خانه های ماتریس را میسازد و رنگ میکند و هر دفعه بعد از هر حرکت صدا
زده میشود

تابع wait_for_key:

این تابع برای دریافت کلید ورودی میباشد و سپس برای هر کلید کارکتری متناظر با آن برمیگرداند
مانند این که q را Q بازمیگرداند

تابع game_over_page:

این تابع صفحه گیم اور را با استفاده از pygame نمایش میدهد و از بازیکن میپرسد که میخواهد ادامه
بدهد یا خیر

تابع game_over:

این تابع چک میکند که آیا حرکتی باقی مانده است یا نه و اگر باقی نمانده بود بازی را به حالت
game_over_page میرد

تابع play:

این تابع تابعی برای شروع بازی و ادامه آن میباشد و چک میکند که بازی کن میخواهد بازی تمام شود
و اگر نه صفحه بازی را آپدیت نگه میدارد و با استفاده از تابع game_over میبند که بازیکن باخته
است یا نه