

گزارشکار پیاده سازی بازی tetris:

- برای پیاده سازی، یک کادر قرمز رنگ برای بازی در نظر گرفته ایم که شماره ستون و شماره ردیف این کادر را در قسمت دیتا سگمنت مشخص کرده ایم و با تابع `draw_border` در ابتدای تابع `main` این کادر را رسم می کنیم.
- برای رسم هر شکل، آن را به 4 مربع با شماره های یک، دو، سه و چهار تقسیم می کنیم و برای رسم هر مربع از شماره ردیف و شماره ستون های چهار خط تشکیل دهنده ی آن که در قسمت دیتا سگمنت نیز ذخیره می شود و تابع `draw_shape` استفاده می کنیم. همچنین برای هر شکل ممکن در بازی، یک متغیر `shape_code` و یک متغیر `shape_color` نیز تعریف کرده ایم. با توجه به شکل های موجود در بازی و حالت هایی که برای چرخش هر یک وجود دارد، `shape_code` یک عدد در بازه ی صفر تا 12 می تواند باشد.
- برای رسم هر مهره ی جدید در بازی، تابع `init_shape` را صدا می زنیم که یک مختصات اولیه از قبل مشخص شده برای آن مهره با توجه به کد شکلی که دارد، در نظر گرفته و آن را در آن قسمت از صفحه رسم می کند (کد شکل به صورت رندوم تولید می شود که یک عدد در بازه ی صفر تا چهار است). حال در صورت حرکت شکل (چرخش، پایین، چپ یا راست رفتن آن) این مختصات در طول بازی تغییر می کند.
- یک تابع اصلی و مهم به نام `update_shape` داریم که در تابع `main` در طول بازی به طور مداوم فراخوانی می شود. وظیفه ی این تابع این است که در هر بار اجرا، چک کند که آیا دکمه های کیبورد توسط کاربر، زده شده است یا خیر. و اگر دکمه ای زده شده است، مناسب با آن، شکل موجود در صفحه را به روز رسانی کند. موارد مهم موجود در این تابع از قرار زیر است:

1- **حرکت شکل به سمت چپ:** اگر دکمه ی `a` زده شده باشد، شکل باید در صورت امکان یک بلوک به سمت چپ حرکت کند. پس ابتدا بررسی می کند که آیا امکان حرکت به سمت چپ وجود دارد یا خیر (با توجه به رنگ پیکسل های سمت چپ شکل، که اگر مشکلی باشد نشان از خالی بودن آن خانه ها دارد). اگر امکان حرکت شکل وجود داشت، ابتدا کل شکل را با فراخوانی تابع `delete_shape` حذف می کنیم (در حقیقت برای حذف یک شکل پیکسل های رنگی آن را به رنگ مشکی در می آوریم) و سپس مختصات ستون های هر مربع آن شکل را به اندازه ی یک بلوک (که ما هر بلوک مربع را 8 پیکسل در 8 پیکسل در نظر گرفته ایم) کم می کنیم. حال دوباره با فراخوانی تابع `draw_shape` شکل جدید با مختصات جدید را رسم می کنیم.

2- **حرکت شکل به سمت راست:** منطق پیاده سازی آن مشابه منطق پیاده سازی حرکت شکل به سمت چپ است.

3- **حرکت شکل به سمت پایین:** اگر دکمه ی `s` زده شده باشد، شکل باید در صورت امکان به اندازه ی یک بلوک به سمت پایین حرکت کند و اگر امکانش نباشد، روی سایر شکل ها، متوقف شود و یک شکل جدید در گوشه ی صفحه `init` شود. برای حرکت به سمت پایین ابتدا تابع `delete_shape` را صدا میزنیم. پس از آن مختصات ردیف های مربع های شکل را به اندازه ی یک بلوک (یعنی 8 پیکسل) افزایش می دهیم و تابع `draw_shape` را فراخوانی می کنیم.

4- **حرکت آنی شکل به انتهای صفحه:** منطق پیاده سازی آن مشابه منطق پیاده سازی حرکت شکل به سمت پایین می باشد، با این تفاوت که در حرکت به سمت پایین تنها به اندازه ی یک بلوک جا به جا نمی شود بلکه به اندازه ای به سمت پایین حرکت می کند که دیگر امکان حرکت آن به سمت پایین وجود نداشته باشد و در همان جا متوقف شود.

5- **چرخش شکل به اندازه ی نود درجه به سمت راست:** در صورت زده شدن دکمه ی `w`، ابتدا بررسی می کنیم که امکان چرخش آن شکل (حول یک نقطه ی مشخص از آن) وجود دارد یا خیر. اگر امکانش وجود داشت، ابتدا تابع `delete_shape` را صدا می زنیم. سپس مختصات مربع های هر شکل را با توجه به نوع آن شکل و `shape_code` آن را آپدیت می کنیم.

6- **حذف کردن یک سطر در صورت پر شدن آن:** می دانیم که پر شدن یک سطر، تنها زمانی میتواند اتفاق بیفتد که یک مهره به سمت پایین حرکت کرده و متوقف شده باشد. پس با هر بار حرکت کردن یک شکل به سمت پایین، و در صورت توقف آن، چک می کنیم که آیا آن شکل باعث پر شدن ردیفی شده است یا خیر. در صورتی که ردیفی را پر کرده باشد، آن ردیف را حذف می کنیم (تمامی پیکسل های آن ردیف را به رنگ مشکی در می آوریم) و سپس شکل باقی مانده روی آن ردیف حذف شده را به اندازه ی یک بلوک به سمت پایین حرکت می دهیم (در حقیقت رنگ هر پیکسل از هر ردیف بالای ردیف حذف شده را، به رنگ پیکسل با ستون مشابه اما دارای شماره ردیفی که 8 واحد از شماره ردیف آن کمتر است در می آوریم). همچنین تعداد ردیفی که هر شکل کامل می کند را در یک متغیر به نام `fill_row_counter` نگه داری می کنیم. در صورت بیشتر بودن مقدار این متغیر از عدد یک، بیست واحد به امتیاز کاربر، که در متغیر `my_score` نگه داری می شود اضافه می کنیم و در صورت یک بودن آن ده واحد اضافه می کنیم و در پایان نیز تابع `show_score` را که وظیفه ی نمایش امتیاز کاربر را به عهده دارد، فراخوانی می کنیم تا امتیاز جدید کاربر بر روی صفحه نمایش داده شود.

7- **پایان بازی:** در صورت رسیدن یک شکل به یک ردیف مشخص در صفحه، تابع `show_game_over` نمایش داده می شود. که یک پیام با محتوای `GAME OVER` چاپ می کند و دستور بازگشت به سیستم عامل را صدا می زند.