

در `board_color` رنگ بلاک های صفحه نگهداری میشوند.

در `board_row` پیکسل سمت چپ-بالای آن ردیف نگه داشته میشود. کلا ۱۱ ردیف وجود دارد (ردیف ۰ تا ردیف ۱۰)

در `board_col` پیکسل سمت چپ-بالای آن ستون نگه داشته میشود. کلا ۸ ستون وجود دارد (ستون ۰ تا ستون ۷)

تا قبل از اینکه شکل سر جایش قرار بگیرد (کلید f فشرده شود) بلاکی را برای نشان دادن بلاک اولیه شکل حاضر در نظر میگیریم و محاسبات بر اساس جایگاه این بلاک و درجه چرخش شکل انجام میشود. بلاک نشاندهنده اشکال در هر حالت (`curr_block`):

درجه ۰	درجه ۹۰	درجه ۱۸۰	درجه ۲۷۰

شکل های جدید در صورت امکان در گوشه بالا سمت چپ صفحه ظاهر میشوند. برای هر شکل باید خالی بودن بلاک هایی که در آن قرار میگیرد را چک کرد و در صورت نیاز، شکل را در ردیف اول آنقدر به راست شیفت داد تا تمام بلاک های مورد نیازش خالی باشند (در صورت نبود این امکان، بازی تمام میشود).

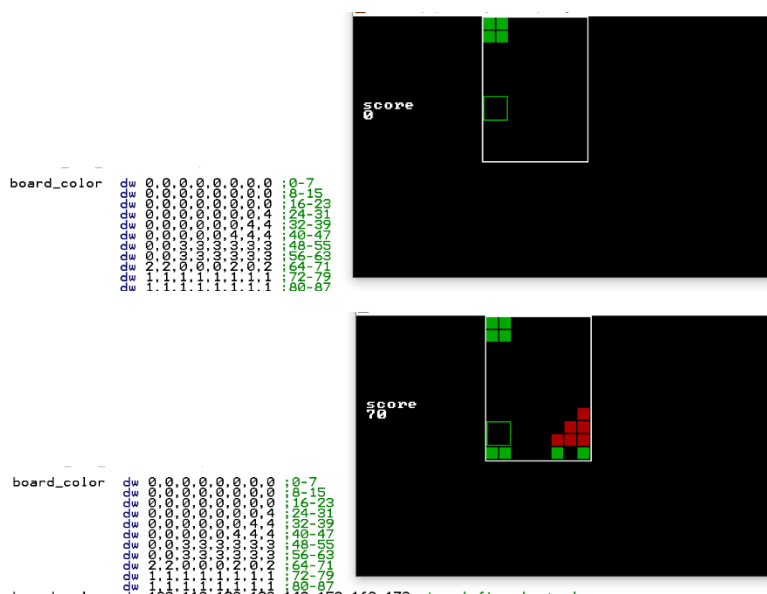
مثلا شکل اول نیاز به چک کردن چیزی ندارد زیرا اگر سطر اول خالی نباشد بازی تمام شده است. برای شکل دوم اگر بلاک اولیه که در جدول بالا مشخص شده است را در نظر بگیریم، با توجه به اینکه سطر اول خالیست، شکل دوم باید بلاک+۸ (فاصله ۲ ردیف برابر ۸ بلاک است) و بلاک+۹ چک شوند. البته چون در مراحل بعد نیز باید برای هر حرکت، امکان قرار گرفتن شکل در بلاک مقصد را چک کنیم، این کار در پراسسی به نام `can_place_here` انجام میشود که مثلا برای شکل دوم بلاک، بلاک+۱، بلاک+۸ و بلاک+۹ چک میشوند و در صورتی که شکل میتواند در آن بلاک قرار بگیرد، پراسس عدد ۱ که نشاندهنده `True` است را در `bx` میگذارد.

برای هر شکل جدید که هنوز سر جایش قرار نگرفته، محلی که با فشردن کلید f در آنجا قرار میگیرد، به صورت کادر دور شکل، نشان داده میشود. برای محاسبه محل قرار گیری شکل، فرض میکنیم شکل در حال حاضر در صفحه نیست (رنگهای آن در board_color را موقتاً به سیاه تغییر میدهیم) و از curr_block شروع میکنیم و یکی یکی به خانه های پایین تر میرویم تا به بلاکی برسیم که نمیتوان شکل را در آن قرار داد. بلاک بالایی آن، جواب است (curr_f_block) که نشان میدهد اگر کلید f را بزیم شکل از چه بلاکی شروع میشود (curr_block به curr_f_block منتقل میشود).

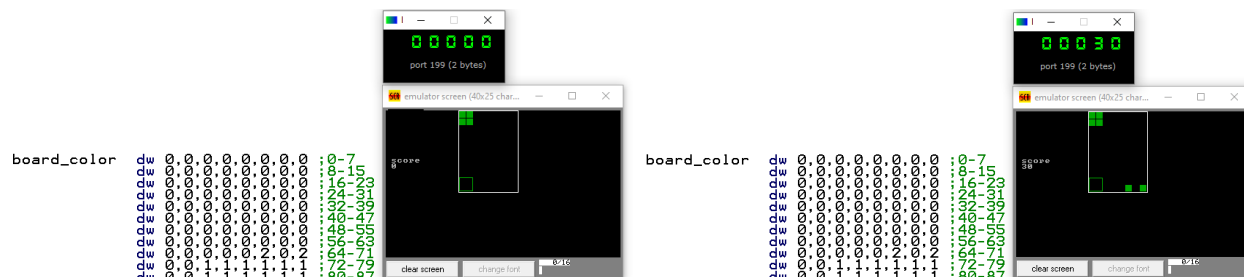
با فشردن کلید f شکل در سر جایش در پایین صفحه قرار میگیرد و چک کردن ردیف ها شروع میشود. به گونه ای که از بلاک سمت راست پایین (بلاک شماره ۸۷) شروع میکنیم و اگر تمام بلاک های ردیفی رنگی بودند، خانه ای آن ردیف در full_row را به ۱ تغییر میدهیم.

سپس از پایین ترین ردیفی که پر است شروع میکنیم و محاسبه میکنیم که کدام ردیف را باید به این ردیف منتقل کنیم (طبیعتاً ردیف هایی که پر هستند باید از بین بروند). برای این کار از skip_rows و bx در پراسس move_rows_down استفاده میشود. به طوری که در مرحله اول اگر ۳ ردیف بالاتر را باید پایین بیاوریم: skip_rows=3 و bx=3*8*2 (تعداد ردیف، ۸=تعداد بلاک بین ۲ ردیف، ۲=چون block_colors آرایه ای از جنس word است) و با حرکت هر ردیف دوباره چک میشود که میتوان از ۳ ردیف بالاتر آورد پایین یا اینکه ۳ ردیف بالاتر پر است و باید از ردیف بالاتری آورد و skip_rows هم در هر مرحله کم میشود تا هر موقع صفر شد موارد گفته شده باز چک شوند. اگر هم به جایی رسیدیم که بلاک ها را از ردیف اول به ردیف فعلی میبریم، از bx در هر مرحله ۱۶ واحد کم میکنیم تا همچنان از ردیف اول استفاده شود و خارج از کادر صفحه نرود (معادل این است که از یه جایی به بعد کل سطر را سیاه میکنیم)

نمونه: (رنگ بلاک ها رو فقط درون کد مشخص کردم و در شکل اولیه کشیده نشده اند ولی در شکل دوم که ردیف ها جابجا شدند معلوم است بلاک های کدام ردیف به کجا جابجا شدند)



برای نشان دادن امتیاز در صفحه گرافیکی باید cursor به محل موردنظر جابجا شود و کاراکترهای موردنظر چاپ شوند. برای نوشتن “score” از پراسس print_score_string استفاده میشود. برای نشان دادن امتیاز هم از show_score استفاده میشود که در آن عدد هگز به صورت دسیمال و یک رشته درمیآید و چاپ میشود. همچنین با استفاده از port 199 در emu8086 به LED_Display.exe وصل شد و امتیاز را در آن نیز نمایش داد.



همچنین در هر مرحله برای کشیدن شکل جدید از رنگ‌های موجود در color_arr استفاده میشود به طوری که color_arr_ptr به خانه شامل رنگ فعلی اشاره میکند و پراسس new_shape از آن استفاده میکند تا شکل جدید را در صفحه بکشد. پراسس show_next_two نیز با استفاده از این آرایه و اشاره‌گر، دو شکل بعدی را در سمت چپ صفحه نشان میدهند.

