

Semestrální práce z KIV/MBKZ

**Tetris pro Android**

Ondřej Matura

A19B0135P

maturao@students.zcu.cz

16.05.2022

# Zadání

Jako téma semestrální práce byla zvolena známa hra Tetris pro Android. Samotný Tetris má spoustu různých oficiálních a neoficiálních vydaných verzí, přičemž každá verze může mít jiná pravidla hry a obsahovat jiné herní mechaniky. Základní pravidla však zůstávají stejná – hraje se na mřížkové hrací ploše, na kterou nám postupně po jednom padají bloky skládající se ze čtyř čtverců; úkolem je každý tento blok nasměrovat tak, abychom na hrací ploše měli co nejvíce zaplněných řádek; jakmile se nějaká řádka zaplní, tak zmizí a nám se přičtou body.

Pro jednoduchost budeme volit spíše pravidla a mechaniky starších verzí Tetrisu. Kromě již popsaných pravidel bude mít hráč možnost vidět příští blok, který bude padat a bude také moci urychlit jeho pád a nebo ho posunout úplně dolů. Rychlost pádu začne na jednom řádku za sekundu a bude se zrychlovat každý level. Nový level hráč dosáhne tím, že zaplní 10 řádek. Čím více řádek najednou hráč zaplní, tím více za to dostane bodů. Počet získaných bodů se také násobí aktuálním levelem.

# Programátorská dokumentace

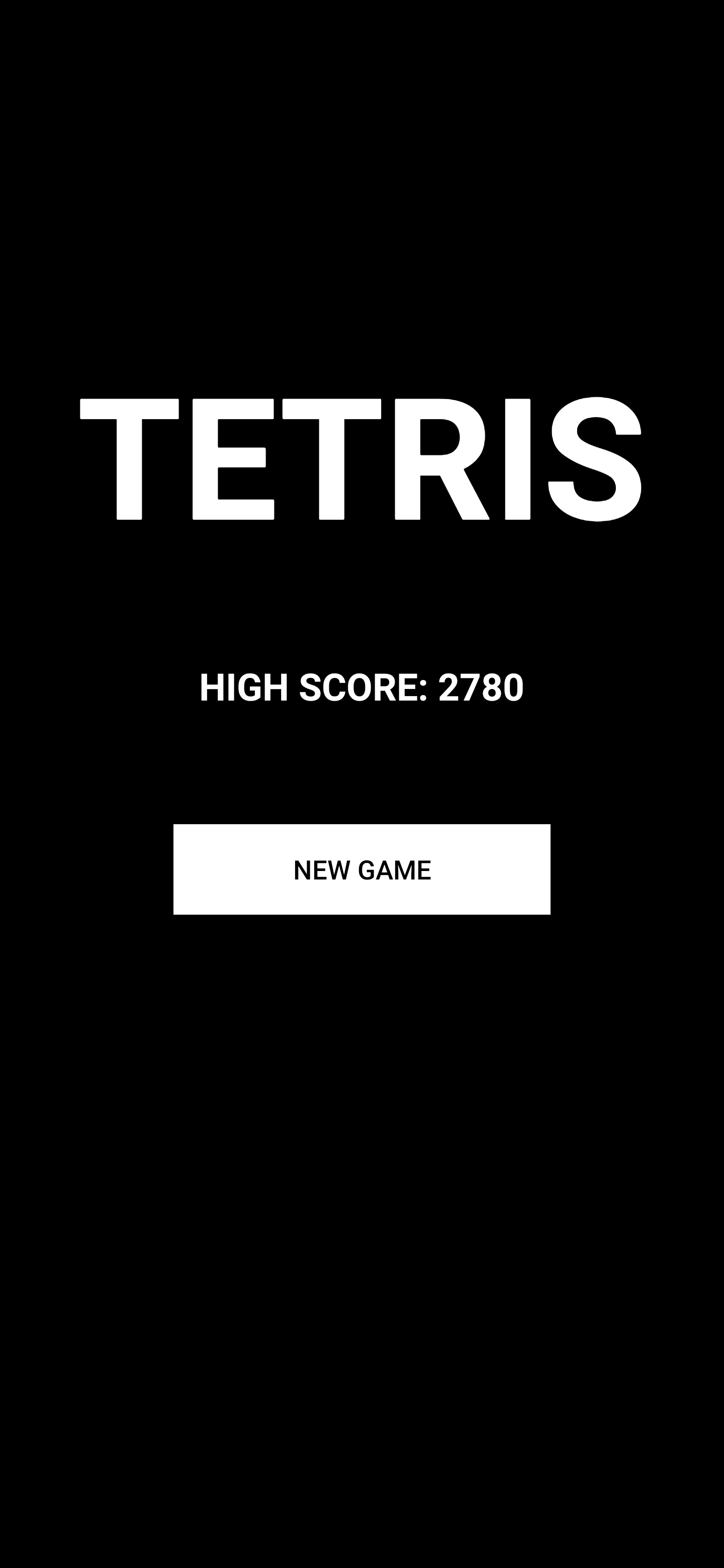
Jako jazyk pro implementaci byl zvolen Kotlin.

Zde jsou popsané třídy a soubory rozdělené dle jejich package (kořenová package projektu je cz.zcu.maturao.tetris):

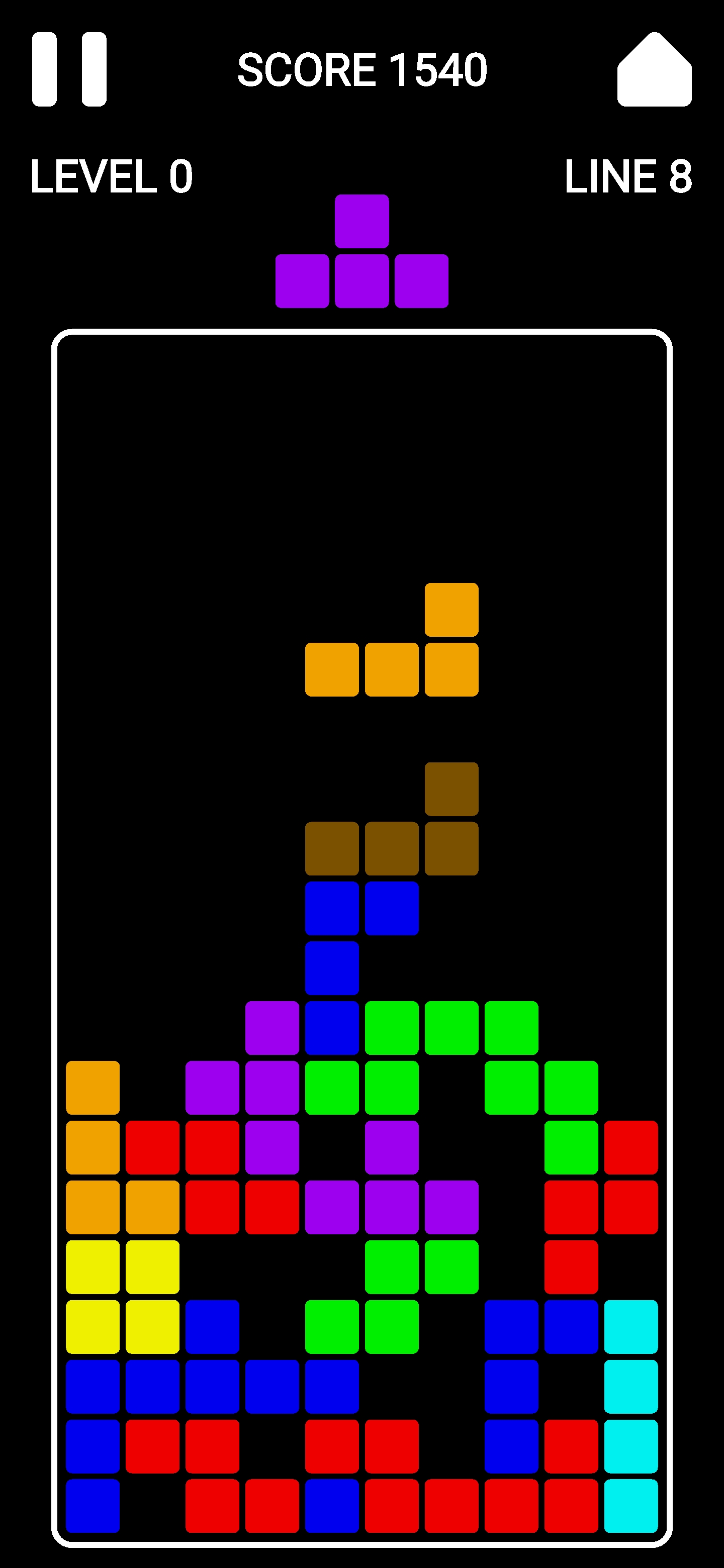
* Package *android*
  + Obsahuje třídy, které jsou nezbytné pro běh aplikace na Androidu
  + Package *activities*
    - Třída *MainActivty*
      * Hlavní menu hry. Stará se o spuštění nové hry, pokračovaní ve hře a zobrazuje hráčovo nejvyšší dosažené skóre
    - Třída *GameActivity*
      * Aktivita, ve které běží samotná hra
  + Třída *GameView*
    - Herní view, které nám umožňuje klasicky vykreslovat hru několikrát za vteřinu (dle FPS)
  + Tř(da *GameLoopThread*
    - Vlákno, které běží paralelně s UI vláknem.
    - Řídí vykreslování hry dle FPS
* Package *game*
  + Zde jsou třídy, které se starají o vykreslování hry a ovládání hry uživatelem
  + Package *ui*
    - Prvky uživatelského rozhraní, které se vykreslují přímo ve hře
  + Třída *Game*
    - Zpracovává vstup od uživatele a vykresluje celou hru
  + Třída *Input*
    - Slouží k načítaní vstupu od uživatele z *ui* vlákna do herního vlákna
  + Třída *StackController*
    - Ovládá a vykresluje herní pole
* Package *logic*
  + Třídy, které řeší pouze herní logiku
  + Třída *Square*
    - Jedna kostka v herním poli
    - Může být buď prázdna, nebo plná s barvou
  + Třída *Matrix*
    - Generická matice
  + Třída *Shape*
    - Tvar, který se skládá z kostek
  + Třída *Shapes*
    - Tvary, která se používají v Tetrisu
  + Třída *Block*
    - Tvar, který má pozici – padající blok
  + Třída *ShapeQueue*
    - Fronta, která podává náhodné nové tvary, které budou padat
  + Třída *TetrisColors* 
    - Konstanty – barvy, které se používají v Tetrisu
  + Třída *Score*
    - Třída, které počítá skóre
  + Třída *Stack*
    - Hrací plocha
* Package *utils*
  + Soubor *code.kt*
    - Obsahuje pouze pomocnou funkci pro vypisování do logu
  + Soubor *draw.kt*
    - Pomocné funkce pro vykreslování

# Uživatelská dokumentace

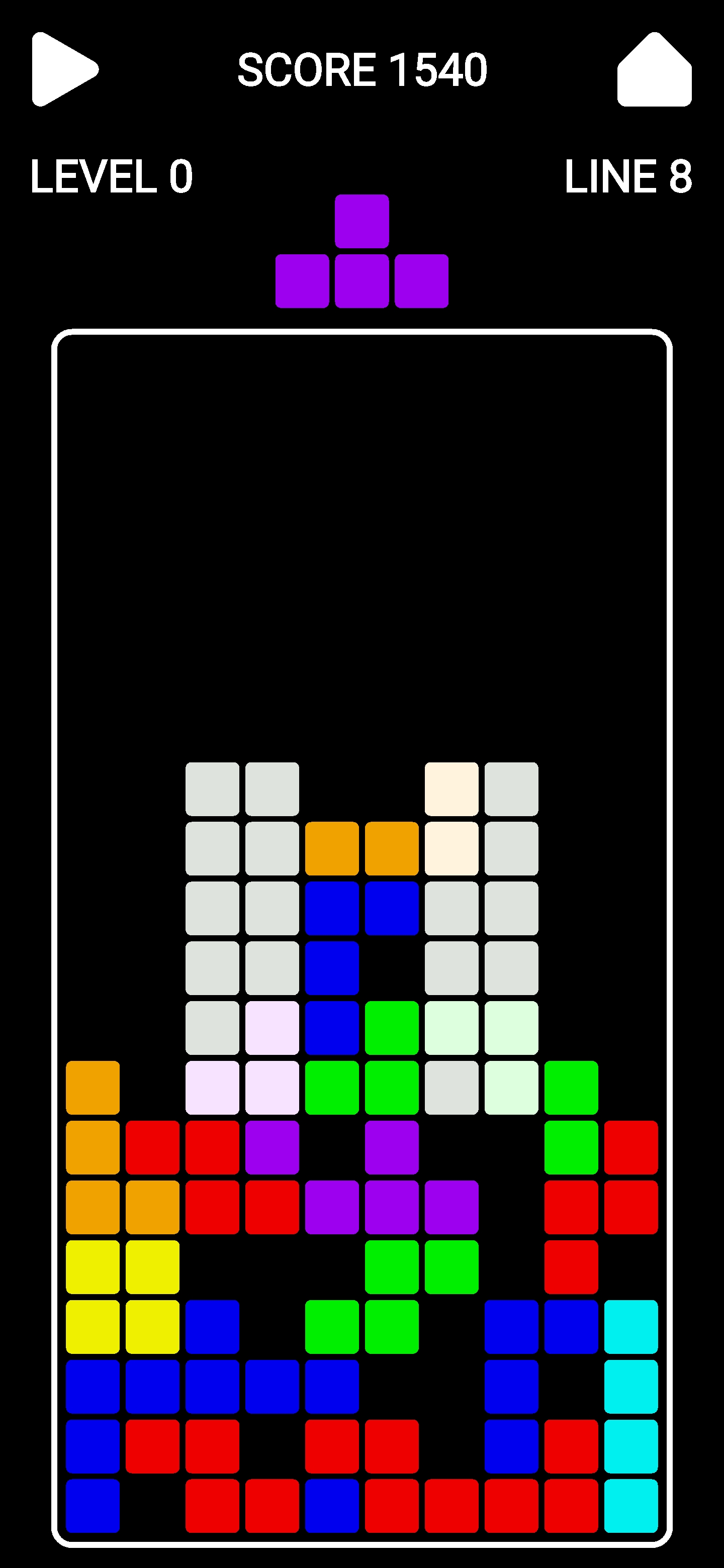
Po spuštění aplikace se uživateli objeví hlavní menu s jedinou možností – tlačítko „NEW GAME“, které spustí novou hru. V hlavním menu je také napsané nejvyšší dosažené skóre uživatele.



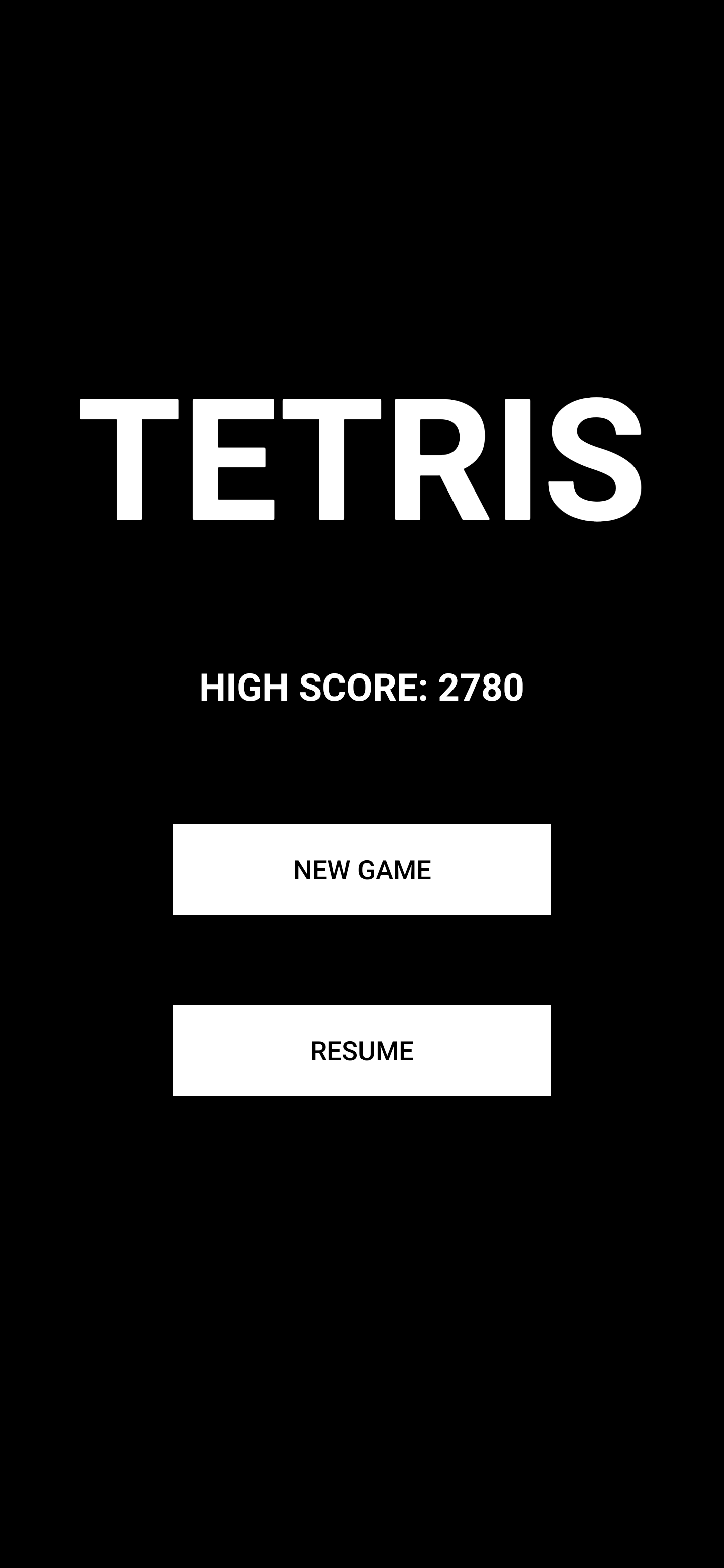
Po spuštění nové hry lze hrát dle pravidel stanovených v zadání. Tahem prstem do stran se posouvá blok do stran. Tahem prstem dolů lze urychlit pád bloku. Ťuknutím na hrací plochu se otočí blok. Ťuknutím na místo, kam blok spadne, se aktuální blok posune až dolů. Nahoře uprostřed se vypisuje aktuální skóre, vlevo je napsaný aktuální level a vpravo je počet vyplněných řádek.



Hra také obsahuje dvě tlačítka – pozastavení/spuštění (vlevo nahoře) a návrat do menu (vpravo nahoře). Hru navíc lze pozastavit stisknutím tlačítka zpět a návrat do menu lze také pomocí tlačítka zpět, pokud je hra pozastavená. Při jakékoliv akci, která hru přeruší (příchozí hovor, odchod na domovskou stránku...), se hra pozastaví.



Po návratu do menu zde přibude nové tlačítko „RESUME“ pro pokračování v započaté hře. Ve hře lze pokračovat do té doby, dokud není dohraná, nebo dokud se aplikace nevypne.



Pokud hráč hru dohraje, vypíše se mu hláška „GAME OVER“ a následně se může jedině vrátit zpátky do menu.



# Řešené problémy

Největší překážky při řešení semestrální práce dělalo zpracování dotykových vstupů od uživatele.

Prvním problémem je, že vstup chceme zpracovávat v herním vlákně, ale od Android API dostáváme vstup z UI vlákna. Proto je potřeba zařídit nějakou komunikaci mezi vlákny. Toto řeší třída game.Input pomocí thread-safe fronty.

Další potíží je, že u normálního Tetrisu, ovládaného např. klávesnici, počítáme s jednoduchými operacemi pro ovládání pozice bloku - posun doprava, posun doleva, posun dolu. Avšak u dotykového ovládání to je složitější. Z podstaty dotykového ovládání potřebujeme obecnější operace pro nastavení pozice bloku – posun na sloupec X, posun na řádek Y. To sebou přináší složitější kontrolu, zda je taková operace validní. Například je třeba dávat si pozor na to, aby blok nemohl projít skrz „zeď“ složenou z jiných bloků.

# Testování

Aplikace byla otestována na těchto zařízení:

* emulátor Pixel 4 s Android 11
* OnePlus 7 Pro s Oxygen OS 11
* Samsung Galaxy A41 s One UI 3.1 (Android 11)

Ne všech zařízení fungovala aplikace správně a uživatelské rozhraní vypadalo tak, jak má.

# Závěr

Výsledná hra funguje docela dobře a je i poměrně hratelná. Oproti jiným nabízeným hrám z obchodu Google Play je sice dost jednoduchá, ale tato vlastnost se dá z jistého hlediska považovat také za výhodu – neobsahuje žádné reklamy, ani mikroinstrukce a je relativně minimalistická.

Hra by se samozřejmě dala vylepšit po různých stránkách. Například přidáním různých animací by hra mohla působit více plynule. Nakonec samotný zdrojový kód by také mohl být o něco čistší a přehlednější.