A thick dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the text '2022/2023'. In the bottom-left corner, there are several thin, curved, overlapping lines in shades of blue and grey, resembling stylized grass or reeds.

2022/2023

# Color fun

Aplikácia pre mobilné zariadenia

Dominik Škvarna – 5ZYS21

# OBSAH

Úvod.....	0
1 Stanovenie cieľov práce .....	1
1.1 Hlavné ciele:.....	1
1.2 Vedľajšie ciele: .....	1
2 Simon color.....	1
3 Text/color .....	1
4 Výzor aplikácie.....	2
4.1 Hlavná obrazovka.....	2
4.2 Nastavenia .....	2
4.3 Nastavenia Simon Color .....	3
4.4 Nastavenia Text/Color.....	3
4.5 Simon Color.....	4
4.6 Text/Color.....	4
4.7 Skóre menu.....	5
4.8 Lokálne skóre.....	5
4.9 Finálne skóre.....	6
5 Využívané časti.....	6
5.1 Room databáza .....	6
5.2 ViewModel.....	7
5.3 Data binding a Navigation .....	8
5.4 Live data .....	8
5.5 Toast a Media Player.....	9
5.6 Ukladanie dát do Firebase realtime databázy .....	9
5.7 Vytvorenie vlastných štýlov.....	9
6 Prehľad dostupných aplikácií.....	10
6.1 Komplexné menu .....	10
6.2 Jednoduché menu.....	11
6.3 Výzor hry Simon Color .....	11
6.4 Výzor hry Text/Color .....	12
Zhrnutie.....	12

# ÚVOD

Ako tému svojej semestrálnej práce som si zvolil hru, ktorú som veľmi rád hrával ako malý chlapec. Je to hra známa pod názvom Simon color. Nakoľko mi je táto téma srdcu blízka, bude sa mi na nej pracovať príjemne a s radosťou. Verím, že mi práve to, okrem iného, pomôže k úspešnému vypracovaniu a obhájeniu svojej práce.

# 1 STANOVENIE CIEĽOV PRÁCE

## 1.1 Hlavné ciele:

- Hlavné menu
- Herný mód Simon Color rôznych náročností
- Herný mód text/color rôznych náročností
- Tabuľka nahraných skóre

## 1.2 Vedľajšie ciele:

- Oku milý grafický dizajn aplikácie
- Zvukové efekty
- Intuitívne ovládanie aplikácie

## 2 SIMON COLOR

Hra pozostáva zo štyroch farebných tlačidiel rôznych farieb. Tieto farby sú spravidla: modrá, červená, zelená a žltá.

V prvom kole sa hra sa začína rozsvietením jedného náhodného tlačidla, ktoré po chvíli zhasne. Hráč musí toto tlačidlo stlačiť a tým zopakovať postupnosť farieb. V druhom kole sa následne rozsvieti tlačidlo z prvého kola a po tom ako zhasne sa rozsvieti ďalšie náhodné tlačidlo, ktoré tiež po chvíli zhasne. Úlohou hráča je zopakovať postupnosť farieb zo všetkých predošlých kôl v správnej postupnosti. Hra sa hrá donekonečna, až kým hráč nespraví chybu v postupnosti farieb.

Osobne si myslím, že tento koncept by sa dal krásne preniesť do sveta aplikácií pre mobilné zariadenia.

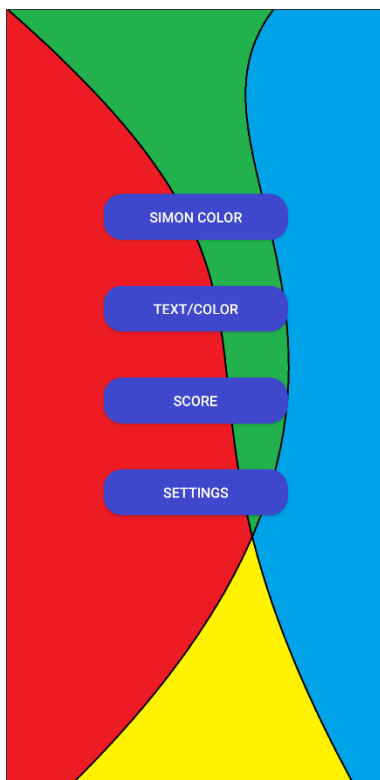
## 3 TEXT/COLOR

Hra pozostáva zo štyroch farebných tlačidiel rôznych farieb, ktoré v sebe majú slovne napísanú náhodnú farbu. Tieto farby sú spravidla: modrá, červená, zelená a žltá. Rovnako tak aj ich slovné obdoby.

Hráč dostane v každom kole inštrukciu, ktoré tlačidlo má stlačiť. Buď tak bude musieť vykonať podľa farby tlačidla, alebo podľa názvu farby, ktorý je v ňom napísaný. Hra končí keď hráč stlačí nesprávne tlačidlo a spraví chybu.

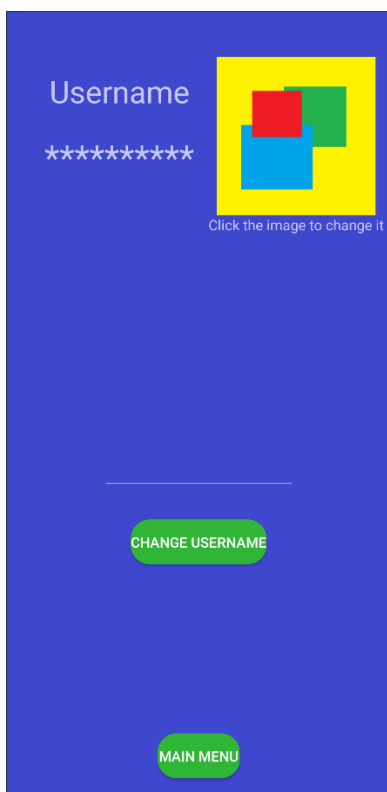
## 4 VÝZOR APLIKÁCIE

### 4.1 Hlavná obrazovka



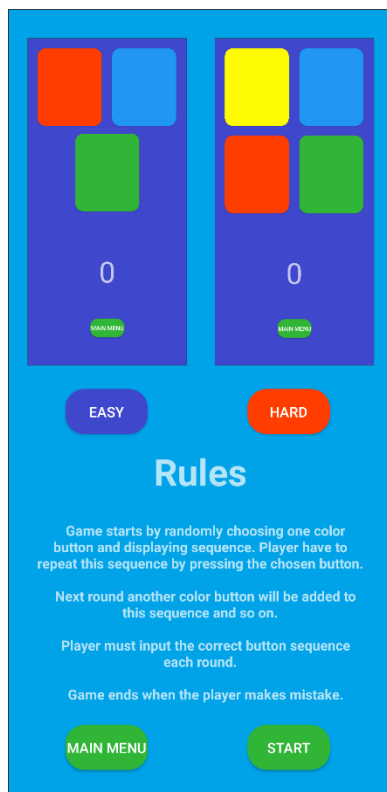
- Po stlačení tlačidla Simon Color sa používateľ dostane na obrazovku nastavení hry Simon Color s krátkym popisom jej pravidiel. Po vybraní a odsúhlasení nastavení sa hra spustí a hráč ju bude môcť hrať.
- Po stlačení tlačidla Text/Color sa používateľ dostane na obrazovku nastavení hry Text/Color s krátkym popisom jej pravidiel. Po vybraní a odsúhlasení nastavení sa hra spustí a hráč ju bude môcť hrať.
- Po stlačení tlačidla Score sa používateľovi zobrazí obrazovka všetkých nahraných skóre na tomto zariadení.
- Po stlačení tlačidla Settings sa používateľovi zobrazí obrazovka nastavení, kde si bude môcť zmeniť používateľské meno a ikonku.

### 4.2 Nastavenia



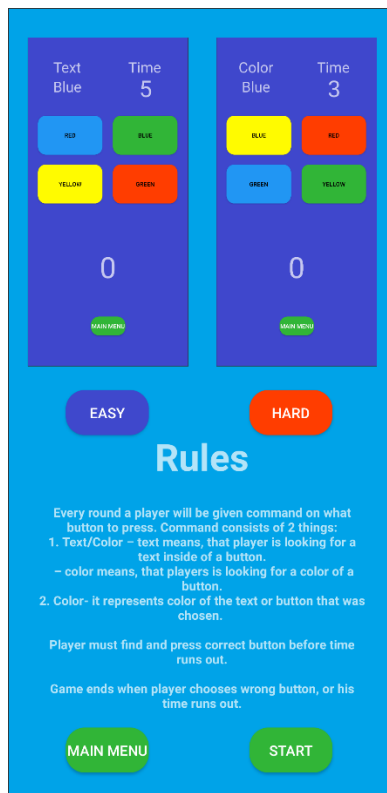
- 
- Po kliknutí na rovnú čiaru môže hráč zadať svoje meno z klávesnice a následne kliknúť na tlačidlo Change username, aby sa zmena používateľského mena potvrdila.
  - Pre zmenu profilového obrázka môže používateľ kliknúť na prednastavený profilový obrázok a tým ho zmeniť na ďalší prednastavený obrázok v poradí.
  - Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

### 4.3 Nastavenia Simon Color



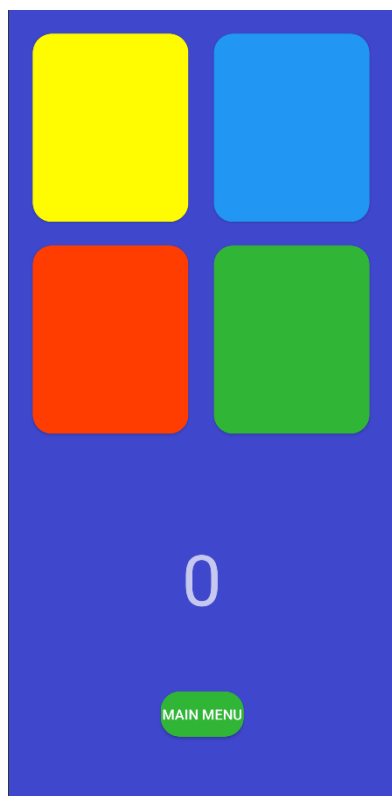
- Na tejto obrazovke si môže hráč prečítať pravidlá hry
- Rovnako má na výber medzi dvoma hernými módmi (EASY/HARD)
- Módy sa líšia v počte tlačidiel, ktoré budú zahrnuté do hry.
- Po výbere obtiažnosti a kliknutí na tlačidlo Start sa zobrazí obrazovka odpočtu začiatku hry a po jeho skončení sa zobrazí obrazovka hry, kde ju už hráč bude môcť hrať.
- Ak hráč nevyberie žiadnu obtiažnosť, nebude môcť hru spustiť.
- Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

### 4.4 Nastavenia Text/Color



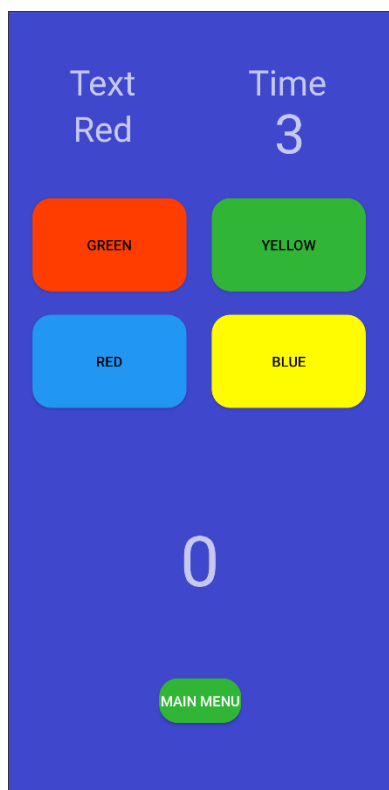
- 
- Na tejto obrazovke si môže hráč prečítať pravidlá hry
  - Rovnako má na výber medzi dvoma hernými módmi (EASY/HARD)
  - Módy sa líšia časom, ktorý bude mať hráč k dispozícii na stlačenie správneho tlačidla v každom kole.
  - Po výbere obtiažnosti a kliknutí na tlačidlo Start sa zobrazí obrazovka odpočtu začiatku hry a po jeho skončení sa zobrazí obrazovka hry, kde ju už hráč bude môcť hrať.
  - Ak hráč nevyberie žiadnu obtiažnosť, nebude môcť hru spustiť.
  - Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

## 4.5 Simon Color



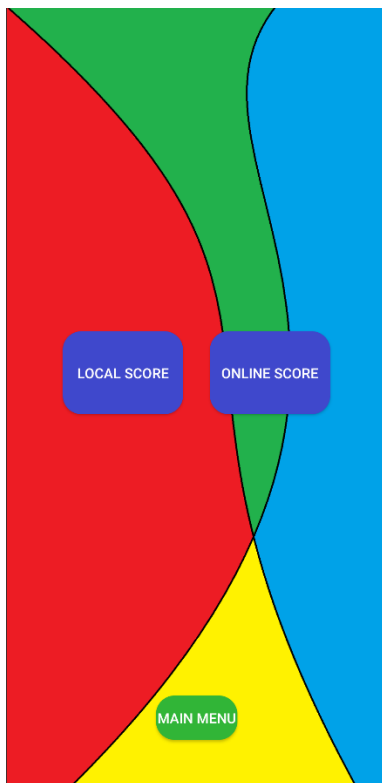
- Hra Simon Color sa hrá na kolá. Každé kolo sa náhodne vyberie jedno tlačidlo, ktoré sa pridá do poradia vybraných tlačidiel. Vybrané tlačidlo sa následne rozsvieti v poradí, v akom boli vybrané.
- To znamená, že v prvom kole sa rozsvieti iba jedno tlačidlo. V druhom kole sa pridá k tlačidlo z prvého kola ďalšie a teda sa v poradí rozsvietia dve. Najskôr prvé vybrané a potom druhé vybrané. Týmto sa hráčovi ukáže poradie tlačidiel, v akom ich má stlačiť.
- Hráč musí tlačidlá stlačiť v rovnakom poradí ako boli vysvietené, inak hra končí. Za každé úspešne zvládnuté kolo dostáva bod. Celkové skóre môže vidieť na obrazovke.
- Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

## 4.6 Text/Color



- 
- Hra Text/Color sa hrá na kolá. V každom kole dostane hráč inštrukciu aké tlačidlo má stlačiť. Na stlačenie tohto tlačidla má určitý časový interval, ktorý sa mu resetuje na začiatku každého kola pokiaľ stlačil správne tlačidlo. Pokiaľ hráč stlačí nesprávne tlačidlo, alebo mu dôjde čas, hra končí.
  - Za každé úspešne zvládnuté kolo dostáva bod. Celkové skóre môže vidieť na obrazovke.
  - Inštrukcia (Text) – hráč musí stlačiť tlačidlo, v ktorom sa nachádza daný text. Daný text sa náhodne generuje a ide o slová: yellow, green, blue a red.
  - Inštrukcia (Button) – hráč musí stlačiť tlačidlo podľa farby, ktorú reprezentuje daný text. Daný text sa náhodne generuje a ide o slová: yellow, green, blue a red.
  - Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

## 4.7 Skóre menu



## 4.8 Lokálne skóre

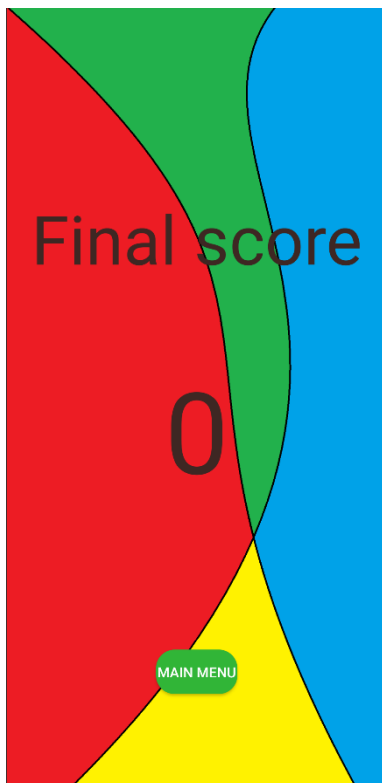


- Po stlačení tlačidla Local score sa zobrazí obrazovka s informáciami o nahraných hrách, ktoré boli odohrané na danom zariadení.
- Funkcionalita Online score nie je naimplementovaná. Dáta z odohraných hier sa síce posielajú do online databázy, ale ich čítanie a zobrazovanie v aplikácii nie je dostupné.
- Dáta, ktoré sa posielajú do online databázy by sa dali využiť na rôzne účely. Napríklad na sledovanie, ktoré herné módy sú viac obľúbené ako iné. Alebo aké priemerné body hráči dosahujú v daných herných módoch. Či napríklad aké profilové fotky si hráči najviac vyberajú.
- Online score by ale mohlo do budúcnosti slúžiť na prezeranie nahraných hier od iných používateľov z iných zariadení, alebo porovnávaní najlepších skóre.
- Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

- 
- Na tejto obrazovke si môžu hráči pozrieť históriu svojich odohraných hier na danom zariadení.
  - Informácie, ktoré sa ukladajú a zobrazujú sú: profilová fotka hráča, meno hráča, hra a jej obťažnosť, dosiahnuté body
  - Hráč môže prechádzať zoznamo hore-dole nakoľko sa dáta zobrazujú v recyclerView.
  - Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.



## 4.9 Finálne skóre



- Táto obrazovka sa zobrazí po prehre v hre.
- Je na nej možné vidieť dosiahnuté skóre, ktoré hráč nahral.
- Po stlačení tlačidla Main menu sa používateľovi zobrazí obrazovka hlavného menu.

## 5 VYUŽITÉ ČASTI

### 5.1 Room databáza

```
private fun addGameRecordToRoomDatabase() { //Metóda pre zápis záznamu o odohranej hre (GameRecord) do Room databázy
    val profilePicture = sharedViewModel.getIndexOfProfilePicture() //Nastavovanie hodnôt premenných zo zdieľaného viewModel-u
    val username = sharedViewModel.getUsername()
    val game = sharedViewModel.getGame()
    val gameDifficulty = sharedViewModel.getDifficulty()
    val score = sharedViewModel.getScore()

    val gameRecord = GameRecord( id: 0, profilePicture, username, game, gameDifficulty, score) //Vytvorenie premennej záznamu hry (GameRecord) s danými vlastnosťami
    viewModel.addGameRecord(gameRecord) //Pridanie záznamu (GameRecord) do Room databázy
}
```

```
@Dao
interface GameRecordDao {

    @Insert(onConflict = OnConflictStrategy.NONE) //Stratégia pokiaľ budú viaceré záznamy rovnaké. V tomto prípade sa to nijak riešiť nebude
    suspend fun addGameRecord(gameRecord: GameRecord) //Suspend (vykonáva sa v inom vlákne ako hlavný beh programu) metóda, ktorá pridá záznam do databázy.

    @Query("SELECT * FROM game_records ORDER BY id ASC") //Načítanie všetkých záznamov z databázy
    fun getAllData(): LiveData<List<GameRecord>>

    @Query("DELETE FROM game_records") //Odstránenie všetkých záznamov z databázy
    fun removeAllData()
}
```

```

Dominik Škvarna
@Database(entities = [GameRecord::class], version = 1, exportSchema = false) //Nastavenie databázy
abstract class GameRecordDatabase : RoomDatabase() {

    Dominik Škvarna
    abstract fun gameRecordDao(): GameRecordDao

    Dominik Škvarna
    companion object { // "Singleton" - Zabezpečí aby v daný moment vždy existovala maximálne 1 inštancia databázy
        @Volatile
        private var INSTANCE: GameRecordDatabase? = null

        Dominik Škvarna
        fun getDatabase(context: Context): GameRecordDatabase {
            val tempInstance = INSTANCE
            if (tempInstance != null) { //Ak už existuje inštancia databázy, vráti danú inštanciu
                return tempInstance
            }
            synchronized(lock: this) { //Ak žiadna inštancia neexistuje, vytvorí ju
                val instance = Room.databaseBuilder(
                    context.applicationContext,
                    GameRecordDatabase::class.java,
                    name: "game_records"
                ).build()
                INSTANCE = instance
                return instance
            }
        }
    }
}

```

```

@Entity(tableName = "game_records") // Dátová trieda, ktorá reprezentuje jeden záznam odohrania hry
data class GameRecord (
    @PrimaryKey(autoGenerate = true)
    val id: Int,
    val profilePicture: Int,
    val username: String,
    val game : String,
    val gameDifficulty: String,
    val score: Int
)

```

## 5.2 ViewModel

```

Dominik Škvarna
class SimonColorViewModel : ViewModel() { //Trieda, ktorá pracuje s dátami
    // a sprístupňuje ich danému/daným fragmentom

    private var round: Int = 1
    Dominik Škvarna
    fun getRound(): Int {
        return round
    }
    Dominik Škvarna
    fun addRound() {
        round += 1
    }

    private var colorButtonsPressed: Int = 0
    Dominik Škvarna
    fun getColorButtonsPressed(): Int {
        return colorButtonsPressed
    }
    Dominik Škvarna
    fun addColorButtonsPressed() {
        colorButtonsPressed += 1
    }
    Dominik Škvarna
    fun setColorButtonsPressed(pNumberPressed: Int) {
        colorButtonsPressed = pNumberPressed
    }
}

```

```

CountDownViewModel
GameRecordViewModel
SharedViewModel
SimonColorViewModel
TextColorViewModel

```

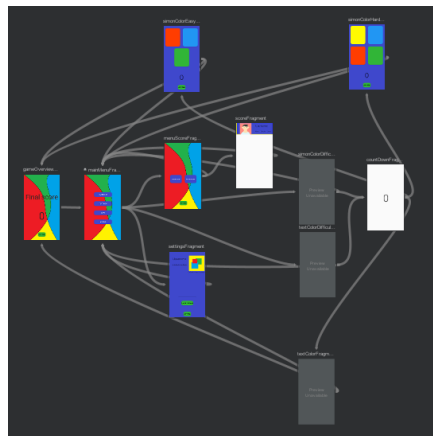
## 5.3 Data binding a Navigation

```
private var _binding: FragmentMainMenuBinding? = null
// Dominik Škvárna
private val binding : FragmentMainMenuBinding
    get() = _binding!! //slúži pre ľahšie pristupovanie k častiam layoutu (TextView, Button, ImageView)
```

```
_binding = FragmentMainMenuBinding.inflate(inflater, container, attachToParent: false) //Nastavenie hodnôt premenným
sharedViewModel.getGameLiveData().observe(viewLifecycleOwner) {}
```

```
return binding.root
```

```
// Dominik Škvárna
private fun setOnClickSimonColor () { //ClickListener pre stlačenie tlačidla
    binding.btnSimonColor.setOnClickListener() { it: View?
        sharedViewModel.setGame("SimonColor") //Nastavenie herného módu na SimonColor
        playSoundButton()
        Navigation.findNavController(binding.root).navigate(R.id.action_mainMenuFragment_to_simonColorDifficultyFragment)
    }
}
```



## 5.4 Live data

```
// Dominik Škvárna
private fun displaySeconds() {
    viewModel.getSeconds().observe(viewLifecycleOwner, Observer { it: Int? //Observer sleduje liveData. Pokiaľ sa dáta zmenia vykoná svoje telo
        playSoundClock() //prehranie zvuku
        if (resources.configuration.orientation == Configuration.ORIENTATION_PORTRAIT) { // Získovanie orientácie zariadenia: Portrait/Landscape
            when (it) {
                2 -> binding.root.setBackgroundResource(R.drawable.green3) //Prepínanie pozadia Fragmentu podľa odrátaných sekúnd
                1 -> binding.root.setBackgroundResource(R.drawable.blue2)
                0 -> binding.root.setBackgroundResource(R.drawable.red1)
            }
        } else {
            when (it) {
                2 -> binding.root.setBackgroundResource(R.drawable.green3_land) //Prepínanie pozadia Fragmentu podľa odrátaných sekúnd
                1 -> binding.root.setBackgroundResource(R.drawable.blue2_land)
                0 -> binding.root.setBackgroundResource(R.drawable.red1_land)
            }
        }
    })

    binding.tvSeconds.text = (it?.inc() ).toString() //Výpis sekúnd do textView tvSeconds
}
```

```
private val seconds = MutableLiveData<Int>()
// Dominik Škvárna
fun getSeconds(): LiveData<Int> {
    return seconds
}
```

## 5.5 Toast a Media Player

```
private fun setOnClickStart () { //ClickListener pre stlačenie tlačidla
    binding.btnStart.setOnClickListener() { //View
        if (sharedViewModel.getIsDifficultyChosen()) { //Podmienka pre zistenie či bola vybraná obtiažnosť
            playSoundButton() //prehratie zvuku
            sharedViewModel.setIsFirstRound(true) //Nastavenie flag-u či sa jedná o prvé kolo na pravdu
            Navigation.findNavController(binding.root).navigate(R.id.action_simonColorDifficultyFragment_to_countDownFragment)
        }
        else {
            playSoundToast() //prehranie zvuku
            Toast.makeText(activity, text: "Choose game difficulty", Toast.LENGTH_SHORT)
                .show() //Výpis krátkej správy na obrazovku
        }
    }
}
```

```
private fun playSoundButton() {
    mediaPlayer = MediaPlayer.create(requireContext(), R.raw.button) //MediaPlayer pre prehratie zvukov
    if (mediaPlayer.isPlaying) { //ak sa zvuk prehrava
        mediaPlayer.pause() //zastavenie zvuku
        mediaPlayer.seekTo(0) //pretocenie na začiatok
    }
    mediaPlayer.start() //zapnutie prehravania
}
```

## 5.6 Ukladanie dát do Firebase realtime databázy

```
private fun addGameRecordToFirebase() { //Metóda pre zápis záznamu o odohranej hre (GameRecord) do Firebase databázy
    val profilePicture = sharedViewModel.getIndexOfProfilePicture() //Nastavovanie hodnôt premenných zo zdieľaného viewModel-u
    val username = sharedViewModel.getUsername()
    val game = sharedViewModel.getGame()
    val gameDifficulty = sharedViewModel.getDifficulty()
    val score = sharedViewModel.getScore()

    val gameRecord = GameRecord(id=0, profilePicture, username, game, gameDifficulty, score) //Vytvorenie premennej záznamu hry (GameRecord) s danými vlastnosťami

    firebaseDatabaseReference.child(pathString: "gameRecord").child(username).push().setValue(gameRecord) //Pridanie záznamu (GameRecord) do Firebase databázy
    // do "podpriechinka" gameRecord
    // do priechinka s názvom meno používateľa
}
```

## 5.7 Vytvorenie vlastných štýlov

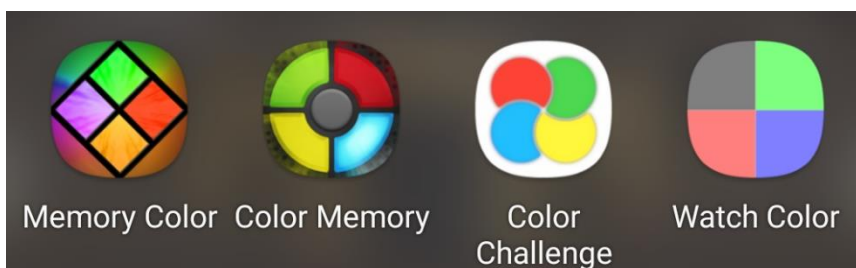
```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="false" android:drawable="@drawable/btn_yellow_normal" />
    <item android:state_pressed="true" android:drawable="@drawable/btn_yellow_pressed" />
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFB00"/>
    <corners android:radius="20dp"/>
</shape>
```

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#AFAD1D"/>
    <corners android:radius="20dp"/>
</shape>
```

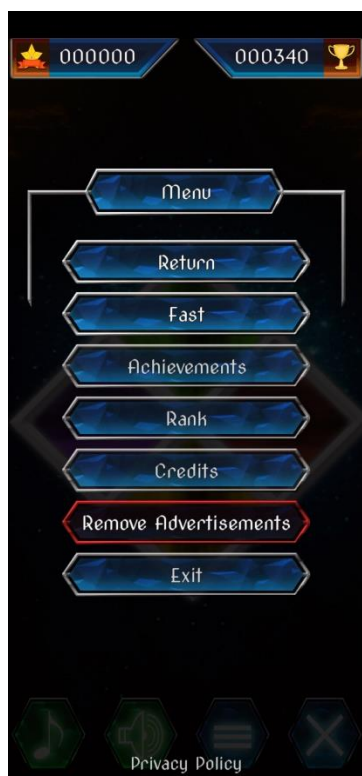
## 6 PREHĽAD DOSTUPNÝCH APLIKÁCIÍ

V Obchod play som našiel 4 solídne aplikácie, ktoré sa podobajú a fungujú podobne ako by som chcel spracovať svoju semestrálnu prácu. Jednalo sa o tieto štyri aplikácie:



Každá z týchto aplikácií riešila a pristupovala k daným problematikám rozličným a svojším spôsobom.

### 6.1 Komplexné menu



- Tento spôsob spracovania menu je vytvorený veľkým množstvom tlačidiel, ktoré slúžia na orientáciu sa v aplikácii.
- Rovnako tak v sebe toto menu implementuje aj určité nastavenia hry. Napríklad tlačidlom "Fast" môžeme nastaviť ako rýchlo bude musieť hráč stlačiť správne tlačidlo, aby mu nevypršal čas
- Tiež si môžeme všimnúť aj to, že menu v sebe implementuje aj ukazovateľ najlepšieho skóre a posledného nahraného skóre

## 6.2 Jednoduché menu



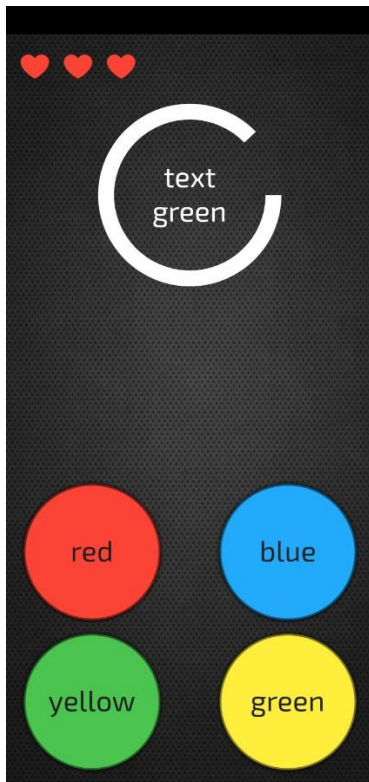
- Toto menu je oproti predošlému oveľa čistejšie čo sa týka dizajnu.
- Implementuje v sebe nastavenie pre vypnutie zvukových efektov, zmenu tmavého režimu na svetlý a rovnako aj tlačidlo na zobrazenie obrazovky skóre, či tlačidlo na zobrazenie úspechov (achievement)
- V pozadí sa nachádza dynamický obrázok, ktorý dopĺňa grafické prostredie
- V strede obrazovky môžeme nájsť tlačidlo na zobrazenie obrazovky hry, ktorej nastavenia sa nedajú meniť a jej obťažnosť je vždy rovnaká.

## 6.3 Výzor hry Simon Color



- 
- V tomto prípade sa farby odlišujú od zaužívanej štvorice farieb. Tlačidlá tiež nie sú usporiadane do štvorca, ale sú v tvare kosoštvorca.
  - Tlačidlá rovnako nemajú jeden statický odtieň danej farby, ale dynamickú textúru, ktorá hre dodáva väčšiu dynamickosť a pôsobí zábavnejšie.
  - Počas hrania vidí hráč svoje priebežné skóre a pri stláčaní tlačidiel sa ozývajú rôzne tóny, podľa farby stlačeného tlačidla.

## 6.4 Výzor hry Text/Color



- Hra sa v tomto prípade ovláda rovnako ako je opísané v sekcii 3
- Hráč ma navyše časovo obmedzený priestor na stlačenie správneho tlačidla v danom kole.
- Implementuje sa tu aj logika životov. Hra končí až keď sa hráč pomýli tri krát.
- Tlačidlá majú kruhový tvar a statickú farbu. To ale vôbec neuberá hre na zábave, nakoľko je podľa môjho úsudku ťažšia ako Simon Color

## ZHRNUTIE

Mojim cieľom je vytvoriť aplikáciu, v ktorej sa budú hrať 2 hry. Jedna z nich bude Simon Color (hra, ktorá cibrí pamäť a využíva pri tom farby) a druhou bude hra Text/Color (hra, pri ktorej je potrebné rýchle rozmýšľanie a postreh).

Najlepších päť skóre z každej hry chcem uchovávať zapísané a sprístupnené pre hráča ako zaujímavosť.

Chcem aby bola aplikácia príjemná na používanie a bola intuitívna. Taktiež aj milá pre oko používateľa.