



SEMESTRÁLNA PRÁCA

Hra 2048



2020/2021

Jakub Rapsik

DOSTUPNÉ APLIKÁCIE PODOBNÉHO ZAMERANIA

2048



Veľmi populárna hra, ktorej cieľom je spájanie dvoch rovnakých čísiel, ktoré nám dajú jeho hodnotu na druhu. Cieľom je, aby sme sa dostali ku číslu 2048. Toto je hra, podľa ktorej chcem spraviť svoju semestrálnu prácu.

Fives



Toto je hra, ktorá sa líši od 2048. Je to hra, ktorú hrávam vo svojom voľnom čase a myslím si, že je to zaujímavá téma. Cieľom tejto hry je získať, čo najväčšie možné skóre. Hra sa po čase stáva ťažšia, že generuje čím ďalej vyššie čísla podľa toho aké máte skóre.

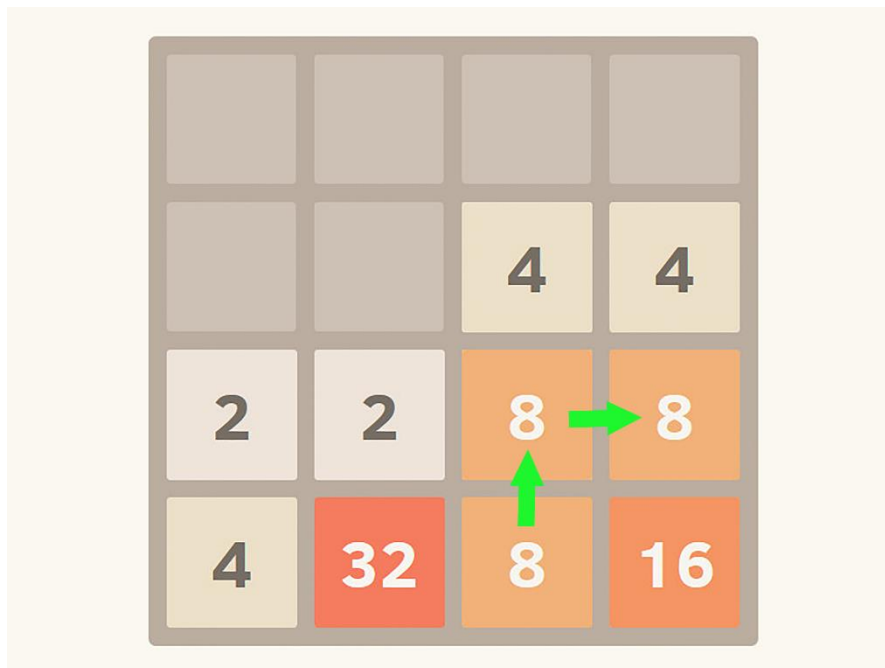
TÉMA: 2048

1.) Pravidlá

Spájame dve rovnaké čísla a vznikne nám z nich ich sčítanec. Po každom ťahu sa náhodne vygeneruje číslo dva alebo štyri.

3.) Pravidlá spájania

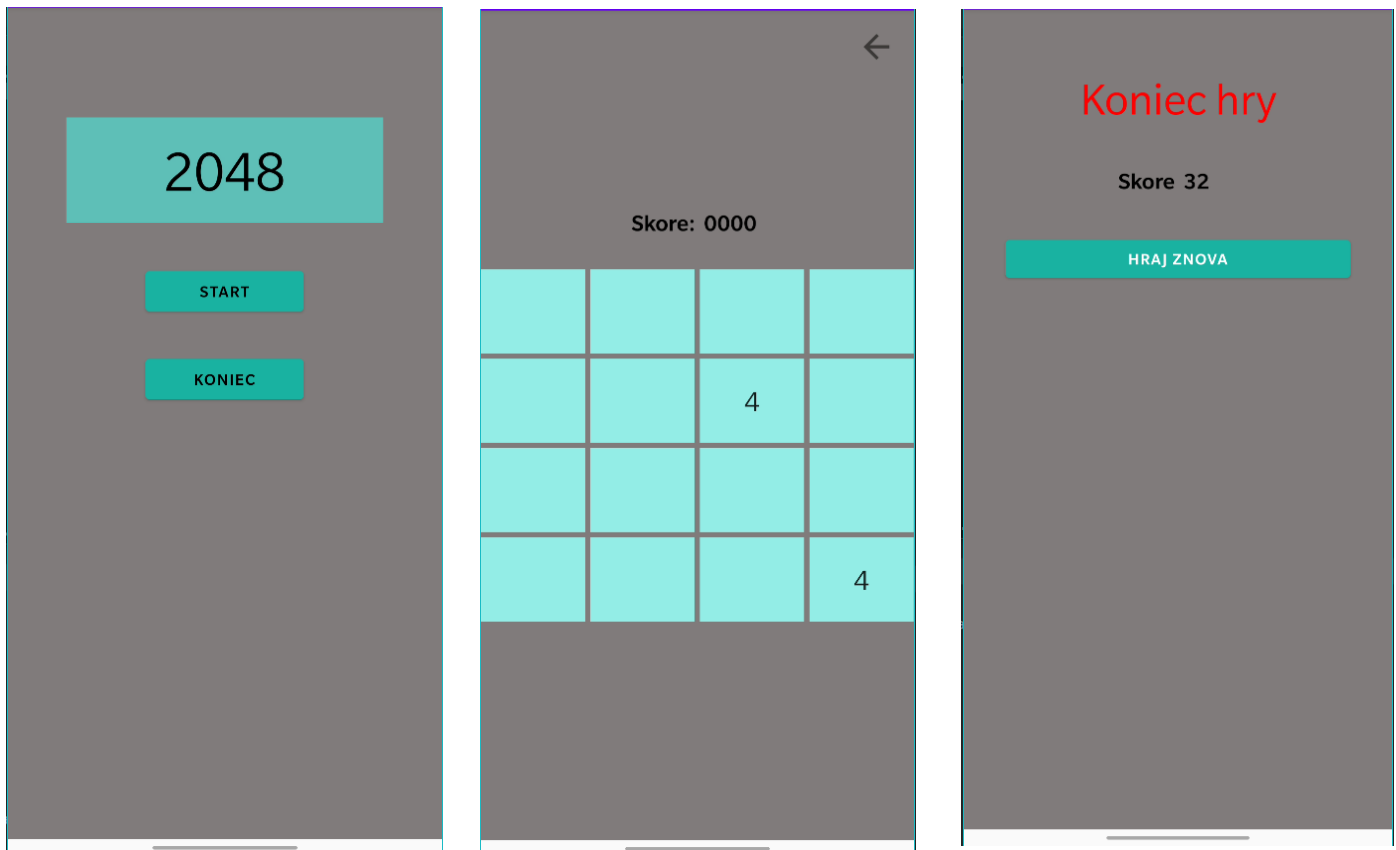
Na začiatku sa začína spájaním čísel 2 a 2 alebo 4 a 4, z nich nám vznikne číslo 4 alebo 8 a to potom spájame s rovnakým ďalším číslom.



2.) Priebeh hry

Po každom posune sa náhodne vygeneruje číslo 2 alebo 4. Cieľom hry je dosiahnuť hodnotu 2048 na jednom z políček. Hra skončí ak sa zaplnia všetky políčka.

ANALÝZA NAVRHOVANEJ APLIKACIE + UKÁŽKA NAVRHU OBRAZOVIEK



Po kliknutí na ikonu aplikácie sa otvorí menu hry, kde budeme mať na výber z dvoch tlačidiel štart game, ktorý nám spusti už samotnú hru a tlačidlo exit, ktorá nám zavrie hru. Po spustenej hry sa náhodne vygeneruje pár základných čísiel na gride 4x4. Hracia obrazovka tak tiež obsahuje aktuálne skóre hráča a šípku, ktorá bude slúžiť na návrat do menu. Hra pokračuje až do bodu kedy v gride nie je žiadne voľné políčko v takomto prípade hra konci. Keď hra skončí na hracej obrazovke sa zobrazí finálne skóre a tlačidlo hraj znova.

NÁVRH ARCHITEKTÚRY APLIKÁCIE

MainActivity

Slúži na spustenie celej aplikácie. A vytvorenie hlavného menu.

KoniecHry

V tejto triede ktorá je zavolaná po vyhratí alebo prehre kde sa vytvory finálna obrazovka s finálnym skóre, tlačidlom a textom. Ak používateľ vyhral vypísaný text bude „Vyhral si“ inak bude „Koniec hry“

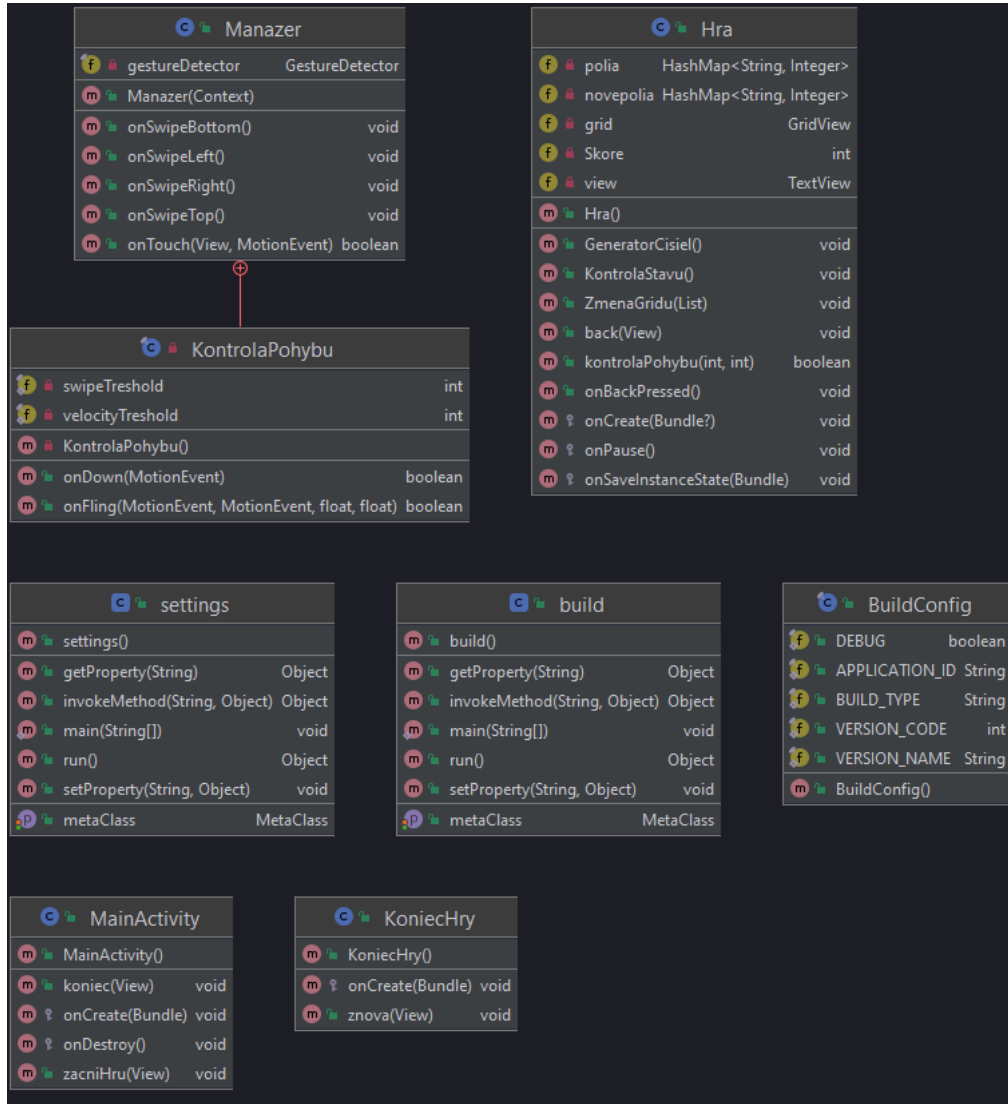
Hra

V triede Hra sa generuje hracia plocha aj s políčkami. Táto trieda obsahuje aj pravidlá pohybu ktoré slúžia na definovanie kedy sa majú čísla pohnúť a kedy sa majú spojiť. Trieda Hra tak tiež obsahuje počítadlo skóre ktoré počas celej hry zobrazuje a na konci ho posielá triede KoniecHry.

Manazer

Trieda spravujúca samotný pohyb.

UML DIAGRAM



POPIS IMPLEMENTÁCIE

V projekte sa nachádzajú 3 aktivity Hlavne menu, Hracia plocha, Finálna plocha. MainActivity sa stará o Hlavne menu. Hra sa stará o Hraciu plochu a KoniecHry sa stará o Finálnu plochu.

Po kliknutí na ikonu sa v MainActivity vytvorí nová aktivita Hlavne menu ktorá obsahuje 2 tlačidlá Start, Koniec a názov hry. Po kliknutí start sa zavolá metóda zacniHru ktorá vytvorí Intent na triedu Hra ktorý sa potom zavolá. Po zavolaní Triedy Hra sa zavolá metóda onCreate ktorá vytvorí Hraciu plochu a samotný grid so základnými dvoma číslami ktore sa najprv uložia do HashMapy a neskôr ich premením na List. V onCreate sa nachádza aj onTouchListener ktorý ak detekuje pohyb zavolá príslušne metódy.

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this setContentView(R.layout.main_game);
    this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    String[] pozicia = new String[]{"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16"}; // Vnutorny grid Pre GridView
    this.view = this.findViewById(R.id.Score);
    this.grid = this.findViewById(R.id.grid);
    Random randomCislo1 = new Random();
    Random randomCislo2 = new Random();
    int cislo1 = randomCislo1.nextInt( bound: 16) + 1; //od 1- 16
    int cislo2 = randomCislo2.nextInt( bound: 16) + 1;
    if (cislo1 == cislo2 && cislo2 != 16) {
        cislo2++;
    } else {
        if (cislo1 == cislo2) {
            cislo1--;
        }
    }
    Random random3 = new Random();
    Random random4 = new Random();
    int[] zakladneCisla = {2, 4};
    int gen1 = random3.nextInt(zakladneCisla.length);
    int gen2 = random4.nextInt(zakladneCisla.length);
    for (int i = 0; i < pozicia.length; i++) {
        if (Integer.parseInt(pozicia[i]) == cislo1) {
            this.polia.put(pozicia[i], zakladneCisla[gen1]);
        }
        if (Integer.parseInt(pozicia[i]) == cislo2) {
            this.polia.put(pozicia[i], zakladneCisla[gen2]);
        }
        if (Integer.parseInt(pozicia[i]) != cislo1 && Integer.parseInt(pozicia[i]) != cislo2) {
            this.polia.put(pozicia[i], 0);
        }
    }
    final List<String> pozicie = new ArrayList<>(Arrays.asList(pozicia)); //Vytvorenie listu a jeho naplnenie podľa array
    this.ZmenaGridu(pozicie);
    this.grid.setOnTouchListener(new Manazer( context: Hra.this) {
```

Po každom ťahu sa vždy skontroluje stav hry zavolaním `KontrolaStavu()`. Ak sa zisti že hra skončila ci už používateľ vyhral alebo prehral vytvorí sa nový intent do ktorého uloží finálne skóre a tak tiež výsledok aby trieda `KoniecHry` vedela ci hra skončila ako vyhrá alebo prehra.

```
public void KontrolaStavu() {
    for (String kluc : this.polia.keySet()) {
        if (this.polia.get(kluc) == 2048) {
            int hodnota = 0;
            Intent intent = new Intent( packageContext: Hra.this, KoniecHry.class); //Do intetn nastavime path ku class KoniecHry aby sme to vedeli neskor spustit
            intent.putExtra( name: "Vysledok", hodnota);
            intent.putExtra( name: "Vysledne skore: ", Skore);
            startActivity(intent); //Spusti triedu KoniecHry a tam sa ohodnotia vsetky veci
            finish();
            return;
        }
    }
    for (String kluc : this.polia.keySet()) {
        if (this.polia.get(kluc) != this.novepolia.get(kluc)) { //Kontrola ci sa stal pohyb
            return;
        }
    }
    int hodnota = 1;
    Intent intent = new Intent( packageContext: Hra.this, KoniecHry.class);
    intent.putExtra( name: "Vysledok", hodnota);
    intent.putExtra( name: "Vysledne skore: ", Skore);
    startActivity(intent);
    finish();
}
```

Po zavolaní triedy `KoniecHry` sa zavolá metóda `onCreate` ktorá vytvorí Finalnu plochu na ktorej sa zobrazí text podľa toho ci používateľ vyhral alebo prehral, Finalne skóre ktore si zistíme pomocou `getIntent` a jedno tlačidlo ktoré nás po stlačení vráti na hlavnú plochu.


```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    this setContentView(R.layout.konice_hry);  
    int vysledok = this getIntent().getIntExtra( name: "Vysledok", defaultValue: 3);  
    int skore = this getIntent().getIntExtra( name: "Vysledne skore: ", defaultValue: 0);  
    Button button = this.findViewById(R.id.VysledokBtn);  
    TextView textView = this.findViewById(R.id.VysledokTxt);  
    TextView Skore = this.findViewById(R.id.FinalSkore);  
    if (vysledok == 0) {  
        textView.setTextColor(Color.GREEN);  
        textView.setText("Vyhrál si");  
        button.setText("Menu");  
        Skore.setText(Integer.toString(skore));  
    } else {  
        textView.setTextColor(Color.RED);  
        textView.setText("Koniec hry");  
        button.setText("Hraj znovu");  
        Skore.setText(Integer.toString(skore));  
    }  
}
```