

Dokumentácia zápočtového programu – Hra s planétami

1. Úvod

Tento program je pokračovaním môjho zápočtového programu z predmetu Programovanie II. Aktuálna iterácia pridáva niekoľko nových pokročilejších featur, rôzne vylepšenia a napravuje mnohé programátorské chyby, ktorých som bol schopný sa s pribúdajúcimi programátorskými vedomosťami a skúsenosťami vyvarovať.

2. Špecifikácia

2.1 Základný popis

Program bude jednoduchá 2D hra. Pre hráča budú pripravených niekoľko situácií – levelov. Na začítaku levelu budú hráčovi predstavené hviezdy a planéty; ich pohyb bude simulovaný ako model newtonovskej gravitácie. Planéty môžu byť zničené nárazom do inej planéty, do hviezdy (aj priblížením príliš blízko na príliš dlho), alebo vzdialením sa príliš ďaleko. Úlohou hráča je pomocou regulácie sily gravitácie (gravitačnej konštanty) v reálnom čase zabezpečiť splnenie cieľov levelu (napr. prežitie aspoň daného počtu planét). Okrem obyčajných planét existujú aj špeciálne planéty (powerupy), ktoré majú pri zrážke určité špeciálne efekty.

2.2 Ďalšie featury

Načítanie powerupov v dobe prekladu neznámých pomocou Reflection. Možnosť zadávať ciele levelu ako lambda výrazy vo vstupnom súbore levelu – program bude schopný minimálne ich rozumnú podmnožinu parsovať a vytvárať aspoň k rozumnej podmnožine z daného lambda výrazu popis cieľu v prirodzenom jazyku.

3. Užívateľské rozhranie a používanie programu

Program používa jediný hlavný formulár. Väčšinu obrazovky zaberá prvok *PictureBox*, na ktorom sa vykresľuje grafika. V strede nad ním je centrálné umiestnený *Label*, ktorý odpočítava zostávajúci čas. Po uplynutí tohto času bude vyhodnotené, či hráč splnil požiadavky na prejdienie levelu a v pravej hornej časti, kde sa na pri načítaní levelu nachádza jeho popis – najmä požiadavky pre splnenie levelu. Úlohou hráča je, aby v momente vypršania času boli splnené všetky požiadavky.

Planéty sú graficky zobrazované ako farebné krúžky – červený označuje *Bad Planet* (zlú planétu), zelený označuje *Good Planet* (dobrú planétu), biely označuje *Neutral Planet* (neutrálnu planétu). Planéty pred sebou tlačia bielu šípku – tá znázorňuje vektor rýchlosti planéty, teda šípka smer šípky reprezentuje smer pohybu planéty a dĺžka šípky reprezentuje veľkosť rýchlosti pohybu planéty.

Powerup-y sú špeciálne planéty, ktoré môžu mať rôzne zaujímavé efekty. Ich výzor je pre každý druh Powerup-u špecifický. Powerup-y môžu byť pridávané ako pluginy (viac v sekcii 4 a 5.2). V základe program obsahuje 3 typy powerup-ov: fialový zväčšuje bezpečnú zónu, žltozelený skracuje zostávajúci čas a červený s krížikom dočasne zničí akýkoľvek objekt, s ktorým koliduje, ten je po čase znova „oživený“. Pri kolízii powerup-u s planétou nie je planéta zničená, pri kolízii 2 powerup-ov sú zničené oba.

Farba hviezd je závislá od ich teploty – aproximuje farbu vyžarovania telesa v závislosti od jeho teploty tak, ako v reálnom svete, teda v poradí rastúcej teploty červená prechádza do oranžovej, žltej, bielej až modrej pre tie najhorúcejšie hviezdy. Číslo napísané v strede hviezdy označuje jej hmotnosť - čím

je hviezda hmotnejšia, tým silnejšie priťahuje planéty. Teplota hviezdy, ako aj vzdialenosť planéty od hviezdy ovplyvňuje, ako intenzívne je planéta hviezdou spaľovaná, čoho účinkom je zmenšovanie planéty. Planéta, ktorá dosiahne príliš malú veľkosť je zničená. Okrem spálenia môže byť planéta zničená aj stretom s inou planétou (zničené obe planéty), alebo s hviezdou (len planéta zničená). Planéta je tiež zničená, ak vycestuje mimo tmavošedej bezpečnej zóny do červenej zóny (farba pozadia).

Ciele levela dosahuje hráč úpravou gravitačnej konštanty pomocou *ScrollBaru* (posuvníka) v pravej časti obrazovky. Posuvník je možné ovládať aj potočením kolečka myši s ukazovateľom myši kdekoľvek nad formulárom.

V pravom hornom rohu sa nachádzajú štyri ovládacie tlačidlá. Tlačidlá s nápisom << resp. >> slúžia na zvolenie predchádzajúceho resp. nasledujúceho levelu. Tieto tlačidlá sú deaktivované v prípade, že aktuálny level je prvý resp. posledný. Medzi nimi sa nachádza tlačidlo s nápisom > alebo // (podľa toho, či aktuálny level beží alebo je pozastavený), ktoré spúšťa resp. zastavuje aktívny level. Pod ním sa nachádza tlačidlo s textom *RESET*, ktoré uvádza aktuálny level do počiatočného stavu.

4. Vstupy a výstupy programu

Vstupmi programu sú vstupné textové súbory s levelmi. Jeden vstupný súbor definuje kompletne informácie o jednom leveli hry. Formát vstupného súboru je nasledovný (fontom Courier písané položky, obyčajným fontom vysvetlivky k jednotlivým položkám):

X – počet planét

X riadkov vo formáte:

Xpos Ypos Xvel Yvel radius type

Xpos, Ypos – Počiatočná pozícia planéty v pixeloch od ľavého horného rohu *PictureBox-u*.

Xvel, Yvel – Počiatočný dvojrozmerný vektor rýchlosti planéty v pixloch za tick

radius – polomer planéty v pixloch

type – typ planéty – 0 = *Good Planet* (dobrá), 1 = *Neutral Planet* (neutrálna), 2 = *Bad Planet* (zlá)

Y – počet hviezd

Y riadkov vo formáte:

Xpos Ypos Xvel Yvel radius mass temperature

prvých 5 údajov rovnako ako pri planétach

mass – hmotnosť hviezdy, temperature – teplota hviezdy

Z – počet PowerUp – ov

Z n-tíc riadkov vo formáte

PowerUpId – unikátny textový identifikátor typu PowerUp

Xpos Ypos Xvel Yvel radius

ďalšie informácie, ich parsovanie má na starosti príslušný *IPowerUpParser*

MaxDistance – určuje ako ďaleko od každej hviezdy sa nachádza zóna, v ktorej sa planéty smú nachádzať bez toho, aby boli zničené

GravityConstant – konštanta, ktorou násobíme všetky gravitačné sily v leveli, je pohyblivá, tento údaj určuje počiatočnú hodnotu

MinGravity

MaxGravity – spolu s predošlou určujú rozsah, v ktorom rozsah hráč smie počas hrania nastavovať gravitačnú konštantu

ScaleFactor – Faktor, ktorým sú pri zobrazovaní násobené všetky veľkosti objektov

`TimeLimit` – dĺžka trvania levelu v sekundách

`R` – počet podmienok pre úspech v leveli – všetky musia byť, aby hráč daný level splnil

`Requirement` – lambda funkcia typu `Func<Level, bool>` v jazyku C#, viac vid'. sekcia 5.3

`Description` – popis podmienky v prirodzenom jazyku, viac vid'. sekcia 5.3

Zbytok súboru môže obsahovať akýkoľvek text, ktorý bude užívateľovi zobrazený pri načítaní daného levelu. Program ignoruje vo vstupnom súbore každý riadok, ktorý začína znakom `#`. V priložených leveloch (textových súboroch) je pomocou `#`-komentárov vytvorená predloha.

Výstupom programu je samotné dianie na obrazovke – simulácia života planét v závislosti od vstupov užívateľa a vyhodnocovanie splnenia levelu.

5. Programátorská dokumentácia

5.1 Popis tried a členov

V tejto časti zhrniem prehľadovo triedy programu a ich najdôležitejšie metódy a premenné. Pre bližšie detaily vid'. komentovaný zdrojový kód.

Program je písaný objektovým štýlom so zameraním na rozšíriteľnosť a všeobecnosť, pričom som sa snažil, aby prílišná všeobecnosť neublížila prehľadnosti a jednoduchosti kódu.

Najdôležitejšou triedou v programe je trieda `Level`. Každá instancia tejto triedy obsahuje informácie o jednom leveli hry a obsahuje metódy na jeho ovládanie. Poskytuje pomerne rozsiahle verejné API, ktoré neobmedzuje programátora pri tvorbe pluginov (viac v časti 5.2) a písaní podmienok pre splnenie levelu (viac v časti 5.3).

Trieda `PhysicsEngine` zodpovedá za simuláciu pohybu planét a ďalších dejov s tým spojených ako napríklad detekcia kolízií, detekcia opustenia poveleného územia či prípadné zničenie vesmírnych telies. Hlavnou metódou je metóda `Tick()`, ktorá pomocou ďalších pomocných metód vykonáva kompletnú prácu simulácie fyziky.

Trieda `GraphicsEngine` zodpovedá za vykresľovanie stavu simulácie na obrazovku pomocou triedy `System.Drawing.Graphics`. Pri každom tiku (volaní metódy `Tick()`) vymaže vykresľovaciu plochu (`PictureBox pictureBox1`) a nakreslí aktuálny stav simulácie.

Trieda `LevelInputReader` je pomocnou triedou, ktorá slúži na načítanie textového vstupu (formát vstupu vid'. 4).

Trieda `PowerUpPluginLoader` slúži na načítanie pluginov (vid'. sekcia 5.2).

Abstraktná trieda `SpaceObject` reprezentuje všetky objekty, ktoré vidí hráč na svojej obrazovke. Jej potomkami sú trieda `Sun` a abstraktná trieda `MovingSpaceObject`. Potomakmi `MovingSpaceObject` sú triedy `Planet` a `PowerUp`.

Trieda `Planet` reprezentuje planéty obiehajúce okolo hviezd. Planéta zaniká (nastavením príznaku `IsDestroyed`) pri zrážke s ľubovoľným iným `SpaceObject`-om, pri vzdialení sa od všetkých hviezd (`Sun`), t.j. mimo bezpečnej zóny do červeného pásma, alebo postupným zmenšovaním jej veľkosti pri prílišnom priblížení k hviezde, pričom rýchlosť zmenšovania závisí od teploty hviezdy a vzdialenosti. Planéty majú 3 druhy (`Good` - zelené, `Neutral` - biele, `Bad` - červené) určené hodnotou premennej `Type` typu `PlanetType` (enum).

Objekty triedy `PowerUp` sú vlastne špeciálne planéty. Program podporuje pridávanie externých pluginov (viac v časti 5.2). Pri zničení `powerup-u` je zavolaná príslušná funkcia podľa príčiny zničenia, pričom `powerup` môže mať rôzne zaujímavé efekty (fantázií kladú medze len obmedzenia API, ktoré sú pomerne široké).

Trieda `Sun` reprezentuje hviezdu. Hviezdy sú nehybné a nepôsobí na ne gravitácia, ale pôsobia svojou gravitáciou na `MovingSpaceObject-y` a svojím teplom na planéty. Tiež bezpečná vzdialenosť (hranice červenej zóny) je určená ako zjendnotenie všetkých kruhov so stredom v nejakom slnku a polomerom definovaným vo vstupnom súbore v položke `maxDistance`.

Zaujímavou metódou je `TemperatureToColor()` (členská metóda triedy `Sun`), ktorá implementuje algoritmus (zdroj: <http://www.tannerhelland.com/4435/convert-temperature-rgb-algorithm-code/>) na hodnovernú aproximáciu farby hviezdy (Planckovej krivky) v závislosti od jej povrchovej absolútnej teploty (vlastnosť `int temperature`). Trieda `Sun` dedí od predka `SpaceObject` položky `xVel` a `yVel`, ktoré popisujú rýchlosť objektu. Pohyb slnka som sa však nakoniec rozhodol neimplementovať, ale tieto položky som ponechal v kóde pre prípadné rozšírenie.

Metóda `UpdatePlanetPos(Sun sun)` triedy `Planet` má za úlohou prepočítať pozíciu danej planéty vzhľadom k slnku, kt. je predané v argumente. Prepočet pozície prebieha nezávisle vzhľadom ku každému slnku a výsledný posun planéty je teda zložením týchto pohybov. Samotný pohyb lineárne aproximuje Newtonov zákon gravitácie, pričom považuje pôsobiacu silu počas trvania jedného tikku za konštantnú, rovnako ako rýchlosť planéty. Ak by jeden tik bol nekonečne krátky, simulácia by dokonale presná. V simulačnom modeli na seba planéty ani slnká vzájomne nepôsobia, teda gravitačné pôsobenie je simulované len medzi slnkami a planétami. Toto riešenie zaisťuje rozumnú programátorskú ako aj výpočetnú zložitosť a pre potreby hry sa ukazuje ako dostatočné.

5.2 Pluginy

Program podporuje načítanie pluginov neznámych v dobe prekladu pomocou `System.Reflection`. Pluginom môže byť akákoľvek .NET assembly, ktorá obsahuje aspoň 2 typy: jeden implementujúci interface `IPowerUpParser` a jeden typ, ktorý je potomkom triedy `PowerUp`. Parser (implementujúci `IPowerUpParser` má za úlohu načítanie daného `PowerUp-u` a vrátenie jeho vhodne vytvorenej instance. Položka `PowerUpId` reprezentuje unikátne ID daného `powerup-u`. Pri nájdení tohoto stringu vo vstupnom súbore pri načítaní `powerup-ov` je volaná funkcia parseru `Parse(LevelInputReader)`. Pomocou `LevelInputReader-u` je potom úlohou parseru vhodne načítať zo vstupného súboru (pomocou metódy `LevelInputReader.ReadLine()`, ktorá vracia ďalší riadok vstupného súboru, ignorujúc komentáre) vytvoriť instanciu `PowerUp-u` a vrátiť ju. V triede `PowerUp` je potrebné okrem piatich metód, ktoré sú volané pri zničení `PowerUp-u` implementovať aj metódu `Draw` – ako nakresliť konkrétny `PowerUp`.

Defaultná cesta k pluginom je zložka `Zapocetak.Net1\plugin`. Ako plugin môže byť načítaná akákoľvek .NET assembly, ktorá vo svojom názve obsahuje reťazec *powerup* a má príponu *.dll*.

5.3 Podmienky pre splnenie levelu

Vo vstupných súboroch je možné definovať podmienky pre úspešné splnenie levelu ako lambda výrazy v jazyku C#. Podmienkou môže byť ľubovoľný lambda výraz v jazyku C#, ktorý má jediný parameter typu `Level` vracajúci hodnotu typu `bool`. Každá podmienka okupuje vo vstupnom súbore 2 riadky – prvý riadok obsahuje samotnú lambda funkciu a druhý riadok obsahuje popis pre hráča v prirodzenom jazyku. Program tiež podporuje generovanie popisov z lambda funkcií pre niektoré výrazy (Expression). Pre automatické generovanie stačí, aby bol popis presne reťazec `auto` alebo `ostal`

prázdný. Generovanie popisu je možné z binárnych výrazov: aritmetických (+, -, *, /), logických (and, or) a predikátových (>, <, ==, <=, >=). Pri narazení na nepodporovaný výraz program vráti jednoducho `ToString()` z danej `Expression`. Ďalšími sú prístup k niektorým členom triedy `Level`, konkrétne `PlanetCount`, `PlanetCountByType`. Pre pohodlnejší zápis pri prístupe k položke `PlanetCountByType` je možné indexovať skrátené a ako index použiť jeden z reťazcov *Good*, *Neutral*, *Bad*.