

# Randio

## 2D adventura založená na pseudonáhodném generování

### Obsah

1. Uživatelská dokumentace.....	1
1.1 Principy hry.....	1
1.2 Průběh hry.....	2
1.3 Popis ovládání.....	6
1.4 Požadavky.....	6
1.5 Další plány s programem.....	6
1.6 Licence.....	6
2. Programová dokumentace.....	7
2.1 Struktura.....	7
2.2 Kompilace a běh programu.....	14
2.3 Zajímavá funkcionalita.....	14
2.3.1 Generování jmen.....	14
2.3.2 Generování překážek.....	14
2.3.3 L-Systémy.....	14
2.4 Nedokonalosti a problémy.....	15
2.4.1 Kolize.....	15
2.4.2 Nemožnost zaručit dokončitelnost levelu.....	15
2.4.3 Pohyb skrz prostory velikosti jednoho bloku.....	15
2.4.4 Hratelnost.....	15

## 1. Uživatelská dokumentace

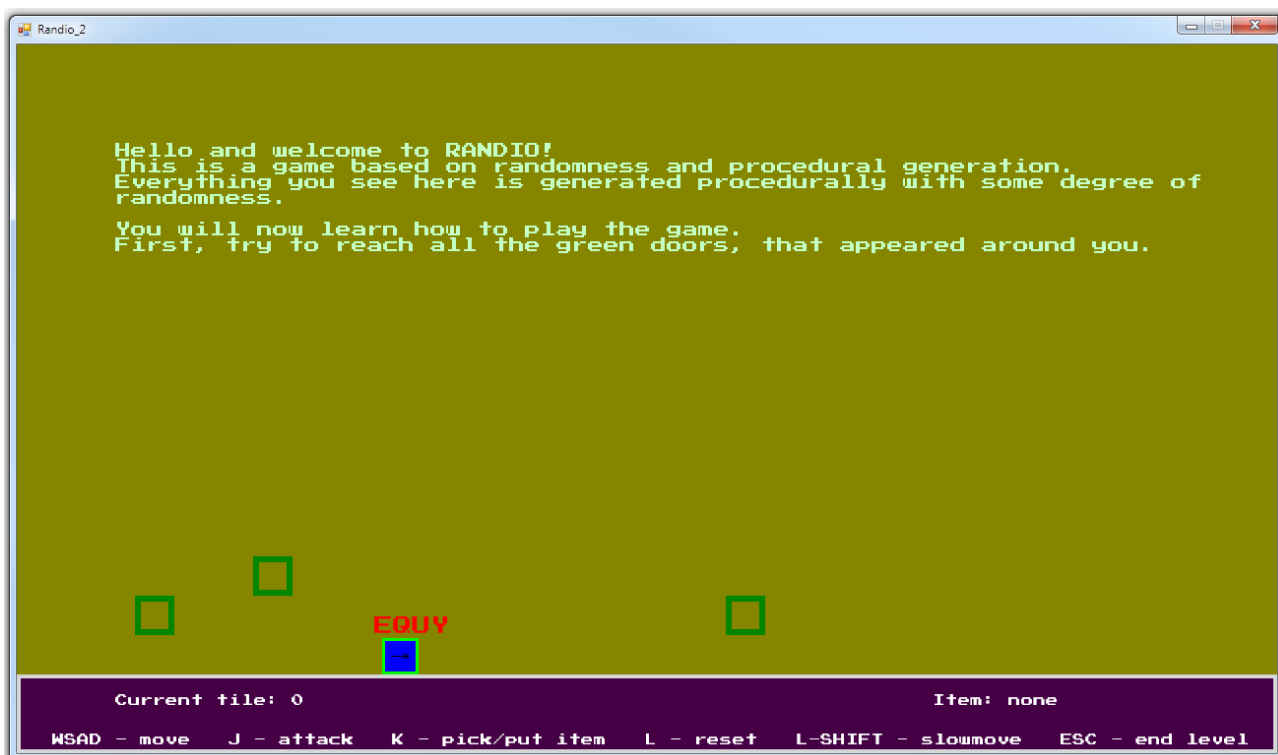
### 1.1 Principy hry

Randio je hra založená na myšlence, že neobsahuje žádné předem připravené pevně dané součásti (grafika, atd.). Všechno, co ve hře hráč vidí a způsob jakým to je uspořádáno, je generováno pseudonáhodně v rámci určitých pravidel.

Generování podléhá následující herní prvky:

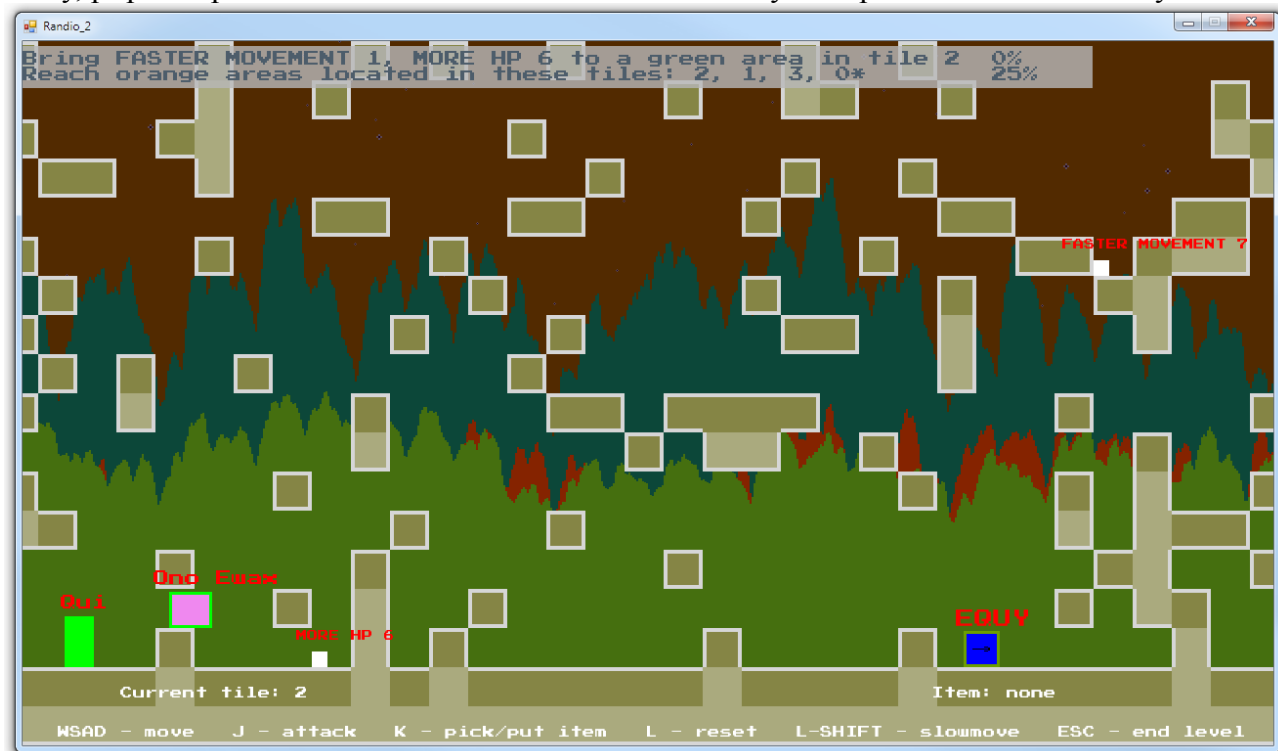
- typ pozadí (tři možné, mezi kterými se náhodně vybírá)
- rozvržení bloků v prostoru
- použitá barevná paleta
- jméno hráče i nepřátel ovládaných umělou inteligencí
- rozvržení a vlastnosti nepřátel a předmětů
- úkoly nutné k úspěšnému splnění levelu

## 1.2 Průběh hry



Po spuštění hry se hráč objeví na takzvané „Intro“ obrazovce, kde mu jsou na praktických příkladech vysvětleny základy ovládání hry (princip hry, pohyb, interakce s nepřáteli a předměty a úkoly).

Po skončení tutoriálu se načte samotná hra, kde se hráč může volně pohybovat a snažit se plnit úkoly, či jen prozkoumávat prostředí. Hra je ukončena ve chvíli, kdy hráč splní všechny zadané úkoly, případně pokud se hráč rozhodne hru sám ukončit kdykoliv předtím stiskem klávesy.



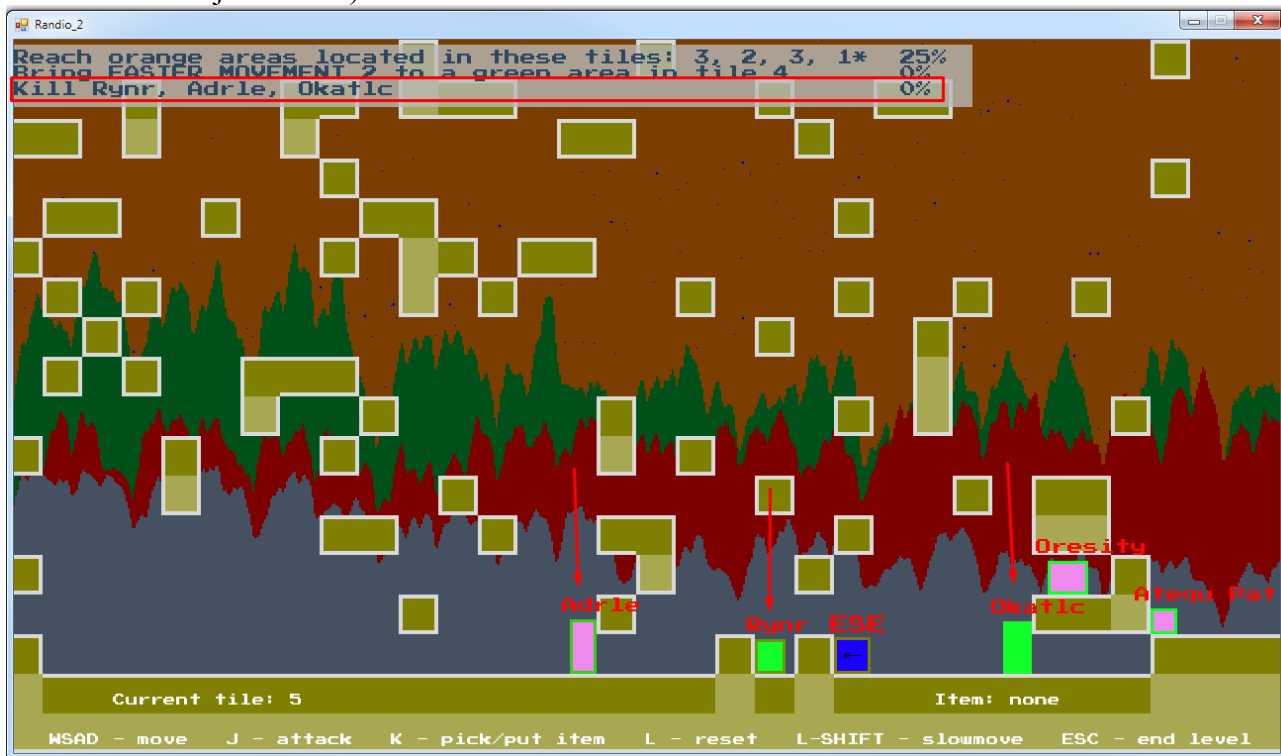
Po ukončení hry je hráči zobrazena obrazovka se statistikami – délka hry, počet smrtí, splněné úkoly a další.



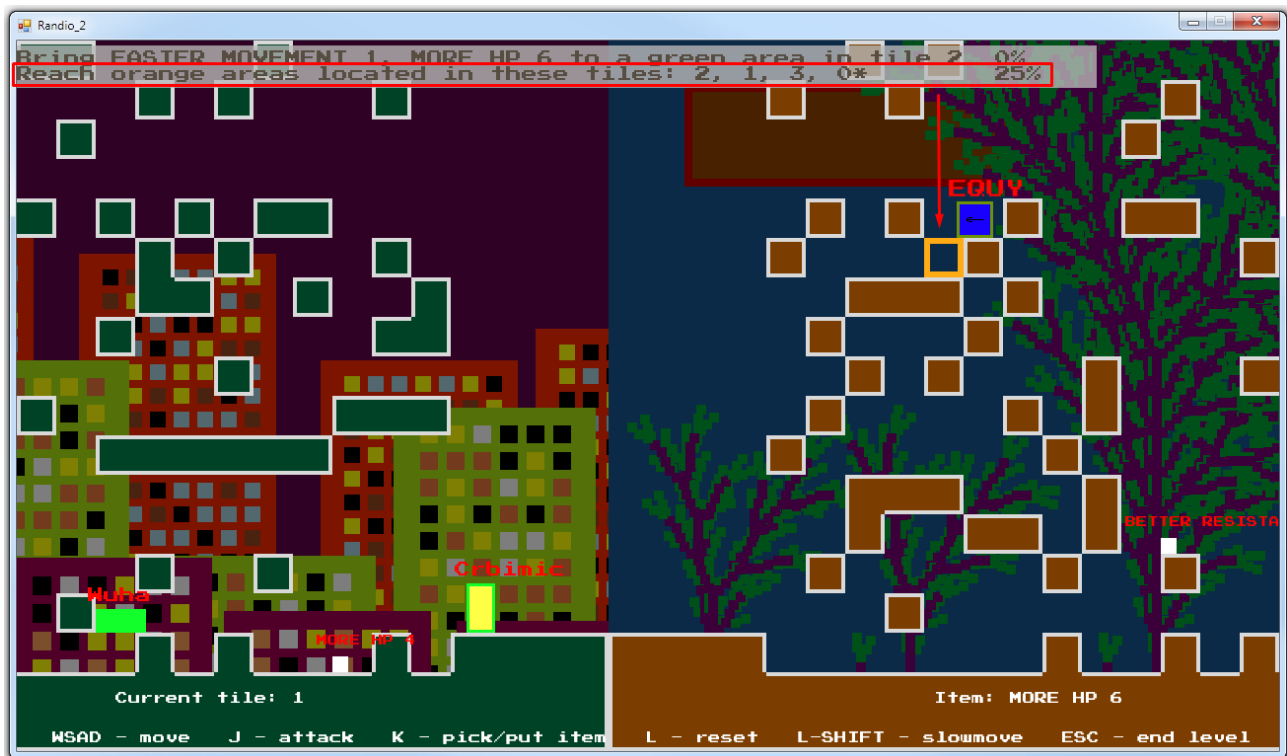
Ve hře jsou nejdůležitějšími prvky úkoly, nepřátelé a předměty.

Úkoly jsou zobrazeny v levé horní části obrazovky. Dělí se na tři typy:

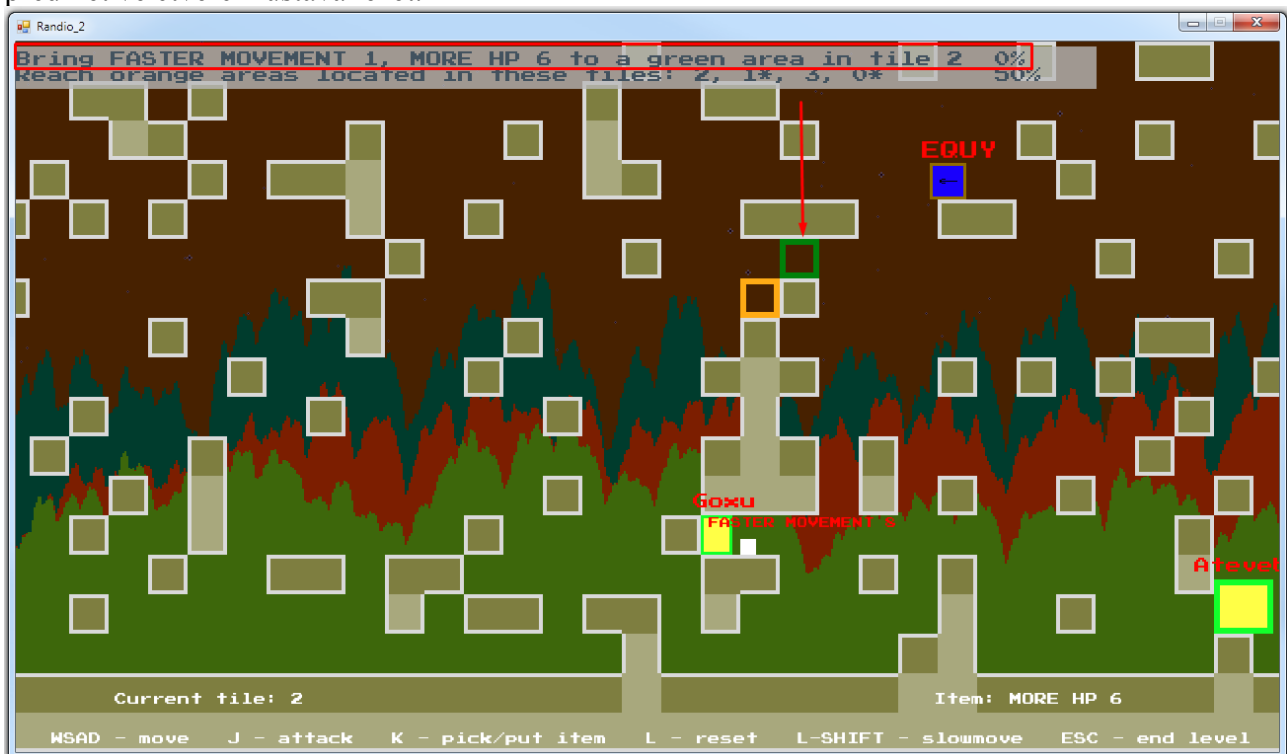
1. zabít nějaké konkrétní nepřátele (v případě, že nepřítel je mrtev, započítá se jako úspěšně zabítý, nezáleží na tom jak zemřel)



2. projít určitými body na mapě, nacházejícími se v určitých konkrétních částech mapy (tilech), vyznačeny jako oranžové čtverce, ve chvíli, kdy do nich hráč vstoupí, zmizí a započte se úspěšné dokončení



3. donést určité předměty na určité místo na mapě – předmět je nutné položit do zeleně označeného čtverce nacházejícího se na dané části mapy (tilu). Úspěch je započítán jen tak dlouho, dokud předmět ve čtverci zůstává ležet.



Splněné části úkolů jsou označeny hvězdičkou (\*)

Nepřátelé ve hře se dělí na tři typy, každý typ má odlišnou barvu a odlišný vzorec chování.

Fialový nepřítel se snaží dostat k hráči a zabít ho.

Zelený nepřítel se naopak snaží od hráče utéct pryč.

Žlutý nepřítel se pohybuje náhodně, na hráče útočí jen někdy a jen v případě, že se hráč dostane příliš blízko.

Fialový a zelený nepřítel mají určitý „dohled“ a jejich chování se projevuje pouze tehdy, pokud se v tomto dohledu hráč objeví. Je-li hráč příliš daleko, chovají se stejně jako žlutý nepřítel.

Úroveň životů nepřítele i hráče je indikována barvou okraje. Zelená znamená plný počet životů. Čím méně životů postava má, tím blíže červené je barva jejího okraje.

Postavy s úzkým okrajem mají normální vlastnosti, postavy s širším okrajem jsou silnější a odolnější. Nepřátelé jsou tím silnější, čím dále směrem doprava se na mapě objevili.



Předměty jsou ve hře reprezentovány jako malé bílé čtverce, nad kterými je zobrazeno jejich jméno. Dělí se na čtyři typy – zvýšení síly, zlepšení obrany, zrychlení pohybu a zvýšení počtu životů.



Předmět na hráče působí jen v případě, že ho hráč má u sebe. Je možné mít u sebe vždy jen jeden.

## 1.3 Popis ovládání

Hra se ovládá pomocí klávesnice. Klávesy s popisy jejich funkce jsou po celou dobu zobrazené v dolní části obrazovky.



Základní ovládání:

WSAD – základní pohyb (nahoru, dolů, doleva, doprava)

J – útok na nepřítele

K – zvednutí/položení předmětu

Další funkce:

L – resetuje hráče na začátek části mapy, ve které se právě nachází (užitečné v případě, že hráč se například zasekne na místě, ze kterého se nedá vrátit zpět – není zaručeno, že mapa bude vždy zcela průchozí)

L-SHIFT (levý shift) – zpomalí pohyb hráče (užitečné například pro jednodušší trefování se do úzkých průchodů)

ESC – v průběhu tutoriálu ukončí tutoriál, v průběhu hry ukončí hru a zobrazí statistiku, v průběhu zobrazování statistiky ukončí program.

## 1.4 Požadavky

Hra je postavena na platformě Microsoft .NET 4.5, jejíž knihovny je nutné mít pro správné spuštění hry nainstalované. Hardwarově nemá hra žádné přehnané nároky, stačí podpora OpenGL či DirectX ve verzi 9 nebo vyšší a je možné ji spustit na prakticky jakémkoliv PC s libovolným OS (podmínkou je jen možnost používat prostředí .NET – v případě Windows nativně, v případě Linuxu např. pomocí Mono).

## 1.5 Další plány s programem

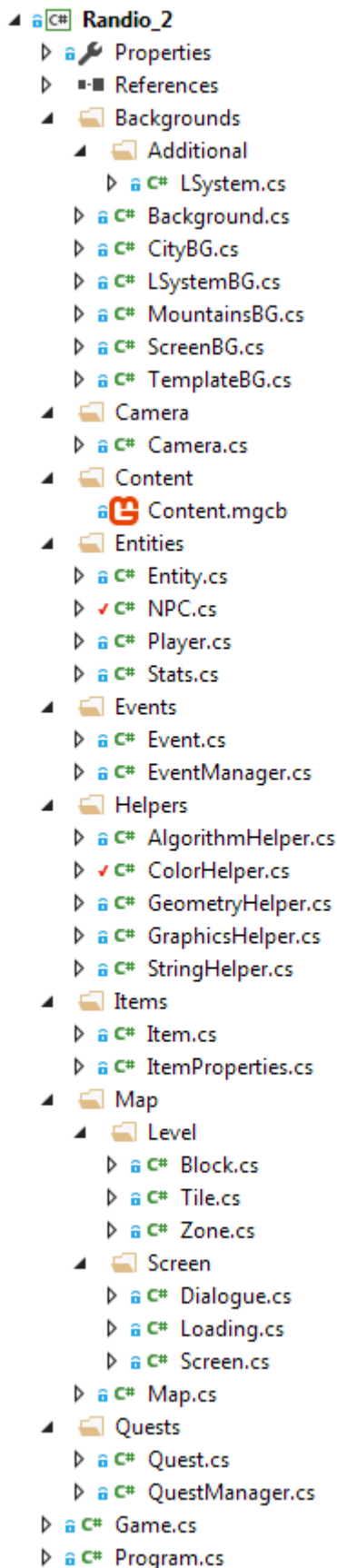
V tuto chvíli nemám s programem žádné konkrétní plány do budoucna. Zdrojový kód i s instrukcemi ke správné kompilaci a spuštění bude i nadále umístěn na GitHub repozitáři, kde bude volně dostupný. Rád bych hru doplnil ještě o zvukové efekty, případně i o pseudonáhodně generovanou hudbu, jelikož audio je něco, co v programu momentálně zcela schází.

## 1.6 Licence

Hra a všechny její součásti jsou licencované pod licencí MIT, jejíž plné znění je přiloženo k programu.

## 2. Programová dokumentace

Program je psán objektově v jazyce C# (platforma .NET 4.5) s použitím knihovny MonoGame. Ke kompilaci používám Microsoft Visual Studio 2015 a jeho integrovanou funkci Build. Struktura programu je následující:



Až na pár výjimek se v každém souboru nachází jedna stejně pojmenovaná třída, u každé třídy i u každé metody se nachází komentář popisující její funkci.

Pro vývojáře obsahuje hra i tzv. „debug menu“ - spouští se klávesou O.

V debug menu fungují následující klávesy:

P – dokud je klávesa stlačena, aktivuje updatování a fyziku.

Pokud klávesa není stlačena, hra se „zastaví“.

Šipky – pohybují kamerou do stran v rámci zobrazených částí mapy (vykreslují se jen ty části, na které hráč může dohlédnout).

### 2.1 Struktura

#### Program.cs

- Automaticky generovaný spouštěč hry, stará se o inicializaci třídy Game a její spuštění

#### Game.cs

- Základní správa hry – obsahuje globální proměnné a parametry hry, metody pro inicializaci a načítání součástí hry

Metody:

- Initialize – připraví renderování, kameru a první objekt mapy (intro obrazovku)

- LoadContent – připraví SpriteBatche a načte font (font je jediná součást hry, která má vlastní asset ve složce Content)

- Update – provádí se při každém vykreslení hry, stará se o updatování informací o uživatelském vstupu, volá Update metody všech herních objektů, stará se o přepínání herních „obrazovek“ (intro/hra/outro)

- Draw – provádí se při každém vykreslení hry, volá Draw na všech herních objektech

Instance:

- Map, Camera, Stats, Dialogue, Loading

## **Složka „Backgrounds“:**

### **Složka „Additional“:**

#### **LSystem.cs**

- Obsahuje pomocné třídy LSystem a Turtle
- Lsystem – implementuje logiku L-Systémů, přidávání a uchovávání axiomu, pravidel a iterování
- Turtle – slouží k vykreslování L-Systémů krok po kroku

#### **Background.cs**

- Rodičovská třída pro všechna pozadí, definuje jejich vlastnosti
- Pozadí je oproti popředí vždy o 50% tmavší
- Všechna pozadí mají barvy generované náhodně pomocí „palety“

#### **Metody:**

- CreateBackgroundTexture (naplní proměnnou Texture vytvořeným pozadím levelu)
- CreateBlockTexture (naplní proměnnou BlockTexture vytvořenou texturou pro běžný blok)
- CreateTopmostBlockTexture (potřebujeme-li v pozadí používat jiný vzhled pro bloky, které se nacházejí „navrchu“ jiných bloků, vytvoříme ho zde a uložíme do BlockTopmostTexture)

#### **CityBG.cs**

- Pozadí typu „město“
- Obsahuje individuální budovy s okny, vše jako objekty
- Několik úrovní vzdálenosti budov, podle toho se překrývají a mají barvu

#### **LSystemBG.cs**

- Pozadí typu „příroda“
- Obsahuje pomocí L-Systému generované stromy a mraky

#### **MountainsBG.cs**

- Pozadí typu „hory“
- Generování inspirované Teragen algoritmem
- „Vlny“ různých barev ve třech úrovních vzdálenosti od hráče a hvězdy

#### **ScreenBG.cs**

- Jednotvářná barva, používá se jako pozadí u tutoriálu a statistik na konci hry

#### **TemplateBG.cs**

- Šablona s popisky pro snadnou implementaci dalších pozadí

## **Složka „Camera“:**

### **Camera.cs**

- Implementuje kameru pro zobrazování hry a pohyb po mapě, umožňuje kameru vycentrovat na daný Rectangle a získat její ViewMatrix pro správné zobrazení mapy

## **Složka „Content“:**

### **Content.mgcb**

- Správce herních assetů, obsahuje pouze jedinou referenci – font.xnb



## **Složka „Entities“:**

### **Entity.cs**

- Rodičovská třída pro NPC a Player, implementuje obecné vlastnosti a funkcionalitu (fyzika, kolize, gravitace..)

#### **Metody:**

- TakeDamage – voláno útočníkem, pokud je tato entita zasažena. Zkontroluje průběh útoku, může odečíst HP, případně nastavit entitu jako mrtvou (a upravit herní statistiky)
- Reset – resetuje entitu na začátek tilu (používá se při resetování hráče pomocí „L“)
- UpdateOutlineColor – nastaví barvu okrajů entity na stav odpovídající současnému HP
- GetFirstEntityInSight – vrací nejbližší entitu v dosahu a směru, jakým je tato entita „natočena“, nebo null, pokud taková entita neexistuje
- GetFirstItemInSight – vrací nejbližší item v dosahu a směru, jakým je tato entita „natočena“, nebo null, pokud takový item neexistuje
- ApplyItemProperties – aplikuje vlastnosti drženého itemu na entitu
- DisapplyItemProperties – odstraní bonusové vlastnosti drženého itemu z entity
- CheckTile – volá se během Update a zjišťuje index tilu, na jakém se entita právě nachází (u hráče je tato informace zobrazena v levé dolní části obrazovky)
- ApplyPhysics – aplikuje veškerou fyziku na hráče (pohyb, skoky, gravitaci, kolize)
- CheckJump – kontroluje, zda hráč skáče a podle toho upravuje jeho pohyb
- TerrainColisionsXY – stará se o řešení kolizí s terénem na obou osách (pro každou osu zvláštní volání)
- EntityColisionsXY – stará se o řešení kolizí s ostatními entitami na obou osách (pro každou osu zvláštní volání)

#### **Instance:**

- Item

### **NPC.cs**

- Dědí třídu Entity
- Ovládá NPC (nepřátele), implementuje AI, jejich vlastnosti a interakce
- Vzorec chování pro NPC je vybrán náhodně ze tří možností (sleduje hráče, utíká od hráče, ignoruje hráče)
- Má určitý „dosah“, ve kterém hráče vidí a může na něj reagovat, pokud je hráč v dosahu útoku, útočí na něj.
- Maximální počet NPC vytvořených na jednom Tilu je 10, na celé mapě tedy (počet Tilů \* 10)

### **Player.cs**

- Dědí třídu Entity
- Ovládání hráče (interakce s klávesnicí), vlastnosti a logika hráčovy postavy
- Zpracovává vstup z klávesnice a reaguje na něj, stará se o útoky na NPC a brání pokládání Itemů

### **Stats.cs**

- Statistiky o průběhu hry, počtu zabitých nepřátel, obdrženém poškození, počtu smrtí, dokončených úkolech a délce hry

### **Složka „Events“:**

#### **Event.cs**

- Umožňuje nastavovat události prováděné v určitý čas. Jednotlivé Eventy jsou spravovány přes EventManager

#### **EventManager.cs**

- Spravuje Eventy, čas a jejich spouštění

### **Složka „Helpers“:**

#### **AlgorithmHelper.cs**

- Obsahuje metody pro generování náhodných čísel

Metody:

- GetRandom – vrací náhodné číslo mezi min (inkluzivně) a max (exkluzivně), int nebo float
- BiasedRandom – vrací náhodné číslo v daném intervalu, přikloněné spíše k jedné či druhé straně

#### **ColorHelper.cs**

- Spravuje barvy a generování barevných palet

Metody:

- InvertColor – obrátí barvu na opačnou
- BlackWhiteContrasting – vrátí černou či bílou, podle toho, která je kontrastnější k dané barvě
- ChangeColorBrightness – Upraví jas dané barvy (+ i -)
- Generate – vygeneruje novou barevnou paletu s daným počtem barev

Podpůrné třídy:

- RYB – implementuje RYB barevný model
- Points – implementuje „barevné kolo“

## **GeometryHelper.cs**

- Pomoc při práci s geometrií (průniky objektů, převody úhlů, atd.)

Metody:

- GetIntersectionDepth – vrací hloubku průniku dvou Rectanglů
- AngleToVector – převede úhel na Vector2
- DegToRad – převod stupňů na radiány
- VectorDistance – vrací vzdálenost dvou vektorů

## **GraphicsHelper.cs**

- Obsahuje metody pro manipulaci s grafikou (objekty, okraje, textury)

## **StringHelper.cs**

- Slouží ke generování jmen Entit

Metody:

- Reset – resetuje zapamatovaná jména (pamatují se, aby se zabránilo generování duplicit)
- GenerateName – vrací nově generované jméno o specifikované délce, defaultně 8
- GenerateWord – podle základních lingvistických pravidel vrací generované „slovo“ dané délky

## **Složka „Items“:**

### **Item.cs**

- Předměty, které, pokud je hráč má u sebe, poskytují určitý bonus. Také mohou být součástí úkolů.

### **ItemProperties.cs**

- Třída obsahující vlastnosti itemu

## **Složka „Map“:**

### **Složka „Level“:**

#### **Block.cs**

- Bloky jsou stavební kameny mapy a tvoří „terén“

#### **Tile.cs**

- Sdružuje pozadí a bloky a je jedním kompletním dílem mapy

Metody:

- CreateEntities – vytvoří na Tilu 1 – 10 NPC s vlastnostmi danými indexem Tilu, na kterém se nacházejí (čím vyšší, tím lepší)
- CreateBlocks – vytvoří náhodné rozvržení bloků (pravděpodobnost vzniku neprůchodného rozvržení je velmi nízká, ale není to zcela ošetřeno)

Instance:

- Background, Block, NPC

**Zone.cs**

- Barevný čtverec označující důležité místo na mapě (oranžové k dosažení, zelené k položení itemů, červené k ukončení tutoriálu, atd.). Metodou Deactivate se dá jeho zobrazení a vyhodnocování kolize s ním vypnout.

**Složka „Screen“:****Dialogue.cs**

- Potvrzovací dialog. Slouží k potvrzení přeskočení obrazovky či ukončení hry. Rodičovská třída pro Loading

**Loading.cs**

- Dědí třídu Dialogue.cs, zobrazuje okno s informací o tom, že se hra načítá  
- Identické jako Dialogue, jen používá jiné indexy barev v CreateGraphics a nemá asociované akce kláves

**Screen.cs**

- Dědí třídu Map, slouží jako mapa pro intro a outro – navíc funkce zobrazování animovaného textu

**Map.cs**

- Hlavní třída implementující celou mapu – sdružuje jednotlivé tily, NPC, hráče, atd.

Metody:

- CheckOutOfMap – zkontroluje, kde vůči mapě se nachází daná Y souřadnice (nad/v/pod)
- GetTileForX – vrátí Tile, na kterém se nachází daná X souřadnice
- GetTileByIndex – vrátí Tile s daným indexem
- ResetPlayer – přesune hráče na začátek současného Tilu
- GlobalToTileCoordinates – převede globální souřadnice (platné pro celou mapu) na lokální (relativní ke konkrétnímu Tilu)
- TileToGlobalCoordinates – převede lokální souřadnice na globální
- GetAllItems – vrátí seznam všech itemů položených na mapě
- IsBlock – vrátí true, pokud se na daných souřadnicích nachází Block
- CreateTiles – vygeneruje náhodný počet Tilů, limitovaný nastavením celkové maximální délky mapy
- CreateQuests – vygeneruje 1-3 questů, každý jiného typu
- CreateQuestBackground – spočítá správnou velikost pro bílé poloprůhledné pozadí pod popisem questů
- GetNewZone – vrátí náhodný Zone umístěný někde na mapě (měl by se nacházet v oblasti dosažitelné hráčem, ale není to zaručeno)

- CreateItems – vygeneruje náhodný počet itemů rozložených po celé mapě, čím vyšší X souřadnici item má, tím větší bonus hráči dává
- CreateExitZone – vytvoří červenou „ukončovací“ zónu
- GetVisibleTiles – vrací seznam Tilů, které jsou viditelné na mapě nebo dost blízko tomu, aby viditelné byly
- GetVisibleNPCs – vrátí seznam všech viditelných NPC

Instance:

- Player, NPC, Quest, Zone

### **Složka „Quests“:**

#### **Quest.cs**

- Stará se o parametry questů, jejich vytváření a kontrolu dokončení

#### **QuestManager.cs**

- Spravuje všechny Questy

## 2.2 Kompilace a běh programu

O celkový běh programu se stará třída Game, jejíž metody Update a Draw jsou spouštěny pro každý vykreslovaný snímek. O jejich spouštění se stará samotné prostředí MonoGame, do kterého není možné (ale ani nutné) zasahovat. Rychlost obnovování je nastavena na výchozí rychlost obnovování monitoru (standardně 60 snímků za sekundu).

## 2.3 Zajímavá funkcionalita

### 2.3.1 Generování jmen

Podle zadané maximální délky (výchozí je 8 znaků) generuje algoritmus jména entity (hráče a NPC). Generování probíhá rozdělením na jedno nebo dvě „slova“ a následným generováním slov. Jsou dva předem připravené seznamy znaků – samohlásek a souhlásek, L a R se nacházejí v obou. Z těchto dvou seznamů se střídavě vybírají náhodné znaky (s vyšší pravděpodobností výběru znaku blíže k začátku seznamu – seznamy jsou seříděny podle frekvence výskytu znaků v angličtině). Při výběru určitých znaků se nastaví tzv. „zakázané znaky“ pro další iteraci, například po výběru „l“ ze seznamu samohlásek se nemůže vybrat „l“ ze seznamu souhlásek, atd.

Nakonec je slovu nastaveno velké počáteční písmeno a je přidáno k celkovému jménu.

### 2.3.2 Generování překážek

Generování překážek (bloků) funguje v několika částech. Na začátku existuje pole booleanů o velikosti rovné počtu bloků, které se do daného Tilu vejdou. Nejdříve jsou zaplněny všechny řady pod definovanou výškou „země“. Na krajích Tilů je vždy několik bloků položených o jednu „úroveň“ výše, které dělají spojnicí mezi Tily. Dále je pak pro každou možnou pozici bloku náhodně rozhodnuto, zda na ní, či na nějakou z vedlejších pozic, bude blok umístěn.

Pravděpodobnost umístění bloku na právě vybranou pozici je 20%, dále pak 6%, že bude umístěn blok vedle současné pozice, a dalších 6%, že bude umístěn na současnou pozici a i nad ní.

Poté jsou podle hodnot v daném poli booleanů generovány bloky s okraji danými podle toho, zda jsou spojené s dalším blokem vedle, či ne.

### 2.3.3 L-Systémy

Jedno z pozadí používá při svém generování princip L-Systémů pro generování „stromů“. Používám pro to vlastní implementaci celého principu generování i vykreslování. V systému je zavedený axiom „F“ a jedno pravidlo – „1FF-[2-F+F-F]+[2+F-F+F]“, kde čísla určují barvy daných sekcí, bloky v závorkách se po vykreslení vrací do počátečního stavu, F je vykreslený blok a +/- otáčí další pohyb vykreslovače o 22° na jednu či druhou stranu. O jejich vykreslování se pak stará tzv. Turtle, který se pohybuje podle vzorce vzniklého po všech iteracích systému a krok po kroku vykresluje čtvercové bloky specifikované barvy pod daným úhlem.

Stromů je vygenerováno postupně šest iterací, od nulté do páté. Z těchto předgenerovaných iterací se poté náhodně vybírají samotné stromy, které jsou vloženy do textury pozadí tak, aby „vyrůstaly“ z úrovně země. Za stromy jsou ještě vykresleny „mraky“, které mají dotvářet pocit venkovní přírody.

## 2.4 Nedokonalosti a problémy

### 2.4.1 Kolize

Hra má jeden hlavní nedostatek, na který je možné během hraní narazit, a to jsou kolize entit s jinými entitami a prostředím. Místy se může stát, že například jedno NPC vytlačí druhé skrz spodní část mapy a druhé NPC vypadne z mapy pryč. Toto je ošetřeno tím, že NPC, které by se objevilo mimo viditelnou část mapy, se resetuje na začátek svého Tilu. Důvod pro tento problém je způsob, jakým jsou kolize řešeny – nejdříve kolize s terénem na ose X, poté s entitami na stejné ose a nakonec znovu terén a entity, ale na ose Y. Bohužel existuje množství případů, kdy by bylo vhodnější řešit kolize v jiném pořadí (tzn. dávat například kolizím s nejnižší vrstvou bloků na mapě nejvyšší prioritu, aby jimi nešlo propadnout), ale vzhledem ke způsobu implementace by to znamenalo velké množství speciálních podmínek pro konkrétní případy, které by vyžadovaly zbytečně velké množství dalšího kódu.

### 2.4.2 Nemožnost zaručit dokončitelnost levelu

Vzhledem ke způsobu, jakým jsou v levelu generovány překážky, není možné zaručit, že každý vygenerovaný level bude možné projít až na konec. Teoreticky je možné, že se například uprostřed levelu vygeneruje rovná stěna bloků, kterou nebude možné nijak překonat. Dalším faktorem, který může k nedokončitelnosti levelu přispět, je generování úkolů – nachází-li se na mapě dost velká část, která je od zbytku levelu uzavřena ohraničením bloky, může se v ní objevit cílová oblast, či NPC, které má hráč za úkol zabít. Zatím k takové situaci nedošlo, ale teoreticky je možné, ač silně nepravděpodobné, že nastane.

Problém s absencí kontroly generování překážek byl vyřešen přidáním klávesy na resetování hráče na začátek současného Tilu. Zatím všechny případy neprůchozího Tilu, se kterými jsem se setkal, šly vyřešit tím, že hráč, který po resetu začne padat z horního levého okraje, dopadne na nějaký výše položený blok a problémový úsek vrchem přeskochí. Pokud by ani to nebylo možné, stejně jako v případě nemožnosti dokončit nějaký z úkolů, hra se dá ukončit stisknutím klávesy ESC, která hráče přesune na souhrn herních statistik.

### 2.4.3 Pohyb skrz prostory velikosti jednoho bloku

Během testování hry jsem narazil na problém, že hráč nebyl schopen se strefit do otvorů velikosti jednoho bloku, jelikož bylo nutné, aby hráč, v té době velký stejně jako jeden blok, byl napozicován na pixel přesně u otvoru, aby jím mohl projít. Jedna možnost byla řešit „dopozicování“ kódem, který by nějak detekoval úmysl hráče otvorem projít, ale nakonec jsem se rozhodl pro jednodušší řešení problému – hráč byl o 4px zmenšen a byla přidána klávesa na zpomalení pohybu, aby se hráč mohl před pokusem projít otvorem přesněji napozicovat. Testování prokázalo, že tato metoda fungovala velmi dobře. S dalšími problémy s průchodností jsem se poté už nesetkal.

### 2.4.4 Hratelnost

Dalším problémem, který je ale způsoben samotnou ideou programu, je hratelnost. Randio není hra v pravém slova smyslu, spíše jde o ukázkou, že je možné udělat skákačku s co největším množstvím pseudonáhodně generovaných součástí a vlastností. Navíc se hra v mnohých věcech odchyluje od klasických her – nemusí být nutně možné dokončit všechny úkoly, hráč se může do určité míry po mapě „teleportovat“ a neexistuje nic jako výhra či prohra, hra na konci vždy jen zobrazí statistiky.