Uživatelská příručka – RPG Demo

Ovládání:

Pohybové ovládání hráče:

* + A / levá šipka – pohyb vlevo
  + D / pravá šipka – pohyb vpravo
  + W / horní šipka – pohyb nahoru
  + S / dolní šipka – pohyb dolů

Ovládání inventáře hráče:

* I – otevrít/zavřít inventář
* Kliknutí na předmět v inventáři – zahodit předmět na zem
* Kliknutí na předmět v inventáři s otevřenou truhlou – přesunout předmět do druhého inventáře

Ovládání útoku hráče:

* Levé tlačítko myši – útok

Ovládání interakcí s objekty:

* E – interagovat

Obsah obrázku snímek obrazovky, kreslené, pixel, Animace

Popis byl vytvořen automatickyLze interagovat pouze s objekty, nad kterým je při přiblížení hráče tento obrázek:   
 -například u dveří:



Ostatní ovládání:

* Esc – Pozastavit hru

Tvorba levelů:

Level je uložen ve formě json souborů ve složce “Assets/Level“. Každý json soubor představuje jeden GameObject – objekt interagující s herní logikou. Vlastnosti GameObjectu jsou definovány komponenty, které obsahuje. Tyto hodnoty těchto komponentů jsou, co zapisujeme do těchto json souborů.

Objekt Vector2: Seřazená dvojice floatů

Přehled komponentů a jejich vlastností:

* Component (abstract class – všechny komponenty implementují tyto vlastnosti)

Summary: Base for all components, components define the properties of GameObjects

* + (string) Type – jméno typu komponentu
  + (bool) enabled – určuje, jestli je komponent aktivní a interaguje s herní logikou
* Transform : Component

Summary: A component responsible for position and scale of the GameObject

Pozn.: Každý GameObject musí obsahovat tento komponent

* + (Vector2) position – udává pozici GameObjectu v herním světě (v pixelech), kde x = horizontální pozice a y = vertikální pozice
  + (Vector2) size – udává velikost GameObjectu (např. při size = (2, 2) budou obrázky 2x větší), x = horizontální velikost a y = vertikální velikost
* Sprite : Component

Summary: A component responsible for holding visual information about the GameObject

* + (string) fileSourceDirectory – cesta k souboru obrázku, který se má vykreslit
  + (int) orderInLayer – určuje, v jakém pořadí se kreslí objekty (vyšší jsou později -> nad ostatními)
  + (bool) flipX – určuje, zda-li je obrázek obrácen podle y-osy
  + (bool) flipY – určuje, zda-li je obrázek obrácen podle x-osy
* UISprite : Component

Summary: A component responsible for rendering overlay UI elements

* + (Anchor) screenAnchor – určuje, od které části obrazovky se bere pozice tohoto GameObjectu – default: TopLeft
    - Anchor { TopLeft = 0, TopCentre = 1, TopRight = 2, MiddleLeft = 3, MiddleCentre = 4, MiddleRight = 5, LeftBottom = 6, MiddleBottom = 7, RightBottom = 8 }
  + (Anchor) pivotPointAnchor – určuje, od které části obrázku se bere pozice tohoto GameObjectu – default: TopLeft
  + (Vector2) offset – udává odchylku pozice obrázku od pozice GameObjectu
  + (int) orderInUILayer - určuje, v jakém pořadí se kreslí UI objekty (vyšší jsou později -> nad ostatními)
  + (string) fileSourceDirectory – cesta k souboru obrázku, který se má vykreslit
* Tilemap : Component

Summary: A component responsible for rendering a grid of images

* + (int) orderInLayer - určuje, v jakém pořadí se kreslí objekty (vyšší jsou později -> nad ostatními)
  + (Dictionary<string, string>) tileSources – slovník propojující zkratky používané při vytváření tilemapy a source jejich obrázků
  + (string[]) tilemapString – array stringů používající zkratky z tileSources pro vytvoření tilemapy
  + (Vector2) tileSize – udává požadovanou velikost každého tilu (obrázku)
  + (Vector2) mapSize – určuje počet tilů (obrázků) v tilemapě
* Collider : Component (abstract class – všechny collidery implementují tyto vlastnosti)

Summary: Base class for all components responsible for detecting collisions

* + (bool) isTrigger – udává, jestli skrz tento collider může hráč projít
  + (Vector2) offset – udává odchylku pozice Collideru od pozice GameObjectu
  + (string) tag – text pro označení Collideru
    - Pozn.: Z důvodu fungování enginu se nesmí používat tagy EnemyHurtbox, EnemyHitbox, Interactible
* BoxCollider : Collider

Summary: A component responsible for detecting collisions in a box area

* + (Vector2) size – určuje velikost collideru (obdélníku)
* CircleCollider : Collider
  + Summary: A component responsible for detecting collisions in a circle area
  + (Vector2) radius – určuje poloměr kruhu collideru
* ToggleCollider : Component

Summary: Component responsible for an interactible object which can activate and deactivate a collider

* + (string) offSpriteImageSource – cesta k souboru obrázku, který tento GameObject má když je collider vypnutý (nechte prázdný aby se neměnil)
  + (string) onSpriteImageSource – cesta k souboru obrázku, který tento GameObject má když je collider zapnutý
* NPC : Component

Summary: Interactible component responsible for creating a dialogue with the player

* + (Dialogue) startingDialogue – dialog, který se spustí při interakci s NPC
    - Dialogue

Summary: A part of the NPC dialogue system, stores all information about how the conversation goes

* + - * (string) id – text používaný k propojení dialogů
      * (DialoguePart[]) parts – části dialogu
      * (Response[]) responses – odpovědi na dialog
    - DialoguePart

Summary: A part of the NPC dialogue system to be used in a Dialogue with a message and the time it'll be shown

* + - * (string) message – text, co NPC říká
      * (int) delay – čas, co text zůstane viditelný (v ms)
    - Response

Summary: A part of the NPC dialogue system to be used in a Dialogue to continue with a specific Dialogue

* + - * (string) linkID – id Dialogu spuštěném při výberu této odpovědi
      * (string) responseText – text ukázaný na tlačítku odpovědi
  + (List<Dialogue>) dialogues – list Dialogů co NPC má
  + (string) npcName – jméno NPC ukázané v dialogovém okně
* CombatStats : Component

Summary: Component responsible for storing combat stats about a GameObject, adds a Hurtbox and an EnemyHitbox to enemies

* + (bool) isPlayerStats – určuje, jestli se GameObjektu vytvoří EnemyHurtbox a EnemyHitbox
  + (int) health – uchovává počet životů co GameObject má
  + (int) damage – udává kolik životů ztratí nepřítel když je zasažen tímto GameObjectem
  + (int) movementSpeed – udává rychlost pohybu GameObjectu
  + (int) agroRange – udává, na jakou vzdálenost začne nepřítel pronásledovat hráče
  + (int) deagroRange – udává, na jakou vzdálenost přestane nepřítel pronásledovat hráče
  + (float) knockbackMultiplier – udává násobitel vzdálenosti, na kterou je nepřítel odhozen při zásahu
  + (int) staggerTime – udává, jak dlouho je nepřítel po zásahu nehybný
  + (int) immuneTime – udává, jak dlouho nemůže GameObject znovu dostat zásah po zásahu
* Animator : Component

Summary: A component responsible for animating a GameObject with   
a sprite component

* + (Dictionary<string, Animation>) animations – Animace tohoto GameObjectu
    - Animation

Summary: An animation to be used with the Animator component

* + - * (string) name – jméno animace
      * (bool) loop – určuje, jestli se animace opakuje
      * (AnimationFrame[]) animationFrames – jednotlivé snímky animace
    - AnimationFrame

Summary: A single frame of an animation

* + - * (string) fileSourceDirectory – cesta k souboru obrázku tohoto snímku
      * (int) time – doba, kterou je snímek zobrazen
  + (string?) playOnLoad – určuje, která animace se spustí při načtění komponentu (nechat prázdné pro žádnou)
  + (float) playBackSpeed – určuje rychlost přehrávání animací (default: 1)
* Inventory : Component

Summary: An interactible component responsible for giving a GameObject an inventory to store items in

* + (List<ItemInInventory>) items – předměty v daném inventáři
    - ItemInInventory

Summary: Info about an item stored in an inventory

* + - * (int) itemCount – počet předmětů
      * (string) itemName – název předmětu
      * (string) fileSourceDirectory – cesta k souboru s obrázkem předmětu
  + (bool) isPlayerInventory – určuje, jestli je toto hráčův inventář
* DroppedItem : Component

Summary: A component responsible for handling items dropped on the ground

* + (ItemInInventory) item – the dropped item

Vzor GameObjektu:

{

"components": [],

"name": "unikatni jmeno",

"isActive": true

}

Vzory Komponentů:

{

"Type": "Transform",

"position": {

"x": 0,

"y": 0

},

"rotation": 0,

"size": {

"x": 1,

"y": 1

},

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "Sprite",

"orderInLayer": 0,

"fileSourceDirectory": "Assets/soubor.png",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "BoxCollider",

"size": {

"x": 0,

"y": 0

},

"isTrigger": false,

"offset": {

"x": 0,

"y": 0

},

"tag": ”a”,

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "Tilemap",

"orderInLayer": 0,

"mapSize": { "x":3, "y":3 },

"tileSize": { "x":16, "y":16 },

"tileSources": {"g":"Assets/soubor1.png", "d": "Assets/soubor2.png", “ ”: “”},

"tilemapString":

[

“ ”, “ ”, “ ”,  
 “g”, “d”, “g”,  
 “ ”, “ ”, “ ”,

],

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "ToggleCollider",

"offSpriteImageSource": "Assets/souborPriOtevrenychDverich.png",

"onSpriteImageSource": "Assets/souborPriZavrenychDverich.png",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "Animator",

"animations": {

"animName": {

"name": "animName",

"animationFrames": [

{

"fileSourceDirectory": "Assets/soubor1.png",

"time": 100

},

{

"fileSourceDirectory": "Assets/soubor2.png",

"time": 100

}

],

"loop": false

}

},

"playBackSpeed": 1,

"playOnLoad": "animName",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "CombatStats",

"isPlayerStats": false,

"health": 100,

"damage": 10,

"movementSpeed": 5,

"immuneTime": 1000,

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "CircleCollider",

"radius": 0,

"isTrigger": false,

"offset": {

"x": 0,

"y": 0

},

"tag": "a",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "DroppedItem",

“Item“:

{

(item…)

}

}

--------------------------------------------------------------------------

{

"Type": "NPC",

"startingDialogue": {

"id": "start",

"parts": [

{

"message": "text npc 1",

"delay": 3000

},

{

"message": "text npc 2",

"delay": 5000

}

],

"responses": [

{

"linkID": "link1",

"responseText": "text odpoved 1"

},

{

"linkID": "link2",

"responseText": "text odpoved 2"

}

]

},

"dialogues": [

{

(dialogues with ids link1, link2,…)

},

],

"npcName": "jmeno npc",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "UISprite",

"screenAnchor": 0,

"pivotPointAnchor": 0,

"offset": {

"x": 0,

"y": 0

},

"orderInUILayer": 0,

"fileSourceDirectory": "Assets/soubor.png",

"enabled": true

}

--------------------------------------------------------------------------

{

"Type": "Inventory",

"items": [

{

"itemCount": 1,

"itemName": "name",

"fileSourceDirectory": "Assets/soubor.png"

}

],

"enabled": true

}