

URHORTS - PLATFORMA PRO TVORBU REALTIMOVÝCH STRATEGICKÝCH HER

Autor: Karel Maděra | Vedoucí: Mgr. Pavel Ježek, Ph. D. | 2019

Úvod

Realtimové strategické hry (RTS) jsou poddruhem strategických her, ve kterém se změny stavu odehrávají v přímé závislosti na změně času v reálném světě. Vývoj tohoto druhu her je složitým procesem, skládajícím se z mnoha částí a zasahujícím do mnoha oborů.

Cíle

Cílem práce je vytvoření platformy pro tvorbu 3D RTS her, která poskytne implementace nejčastějších součástí RTS her, jakými jsou pohyb jednotek či střelba na cíl. Pro dodání součástí specifických pro vytvářenou hru bude vytvořen systém, který umožní tyto části zabalit do „balíčku“. Tento balíček následně platforma na straně koncového hráče umožní nahrát a použít pro hraní hry. Pro definici chování herních prvků a logiky umožní platforma využití jazyka C# a platformy .NET.

Pro tvůrce balíčků bude platforma poskytovat následující:

- knihovnu funkcí implementující obvyklé součásti RTS her, která umožní jednoduchou implementaci logiky hráčů, jednotek, budov, projektilů a úrovní v podobě pluginů;
- editor úrovní, umožňující vytvoření map za použití nástrojů definovaných v platformě či nástrojů definovaných v pluginech;
- systém balíčků, umožňující distribuci vytvořené logiky, map, 3D modelů, textur a dalších součástí hry.

Koncový hráč poté bude schopen takovýto balíček nahrát do své nainstalované instance platformy. Po nahrání bude hráč schopen:

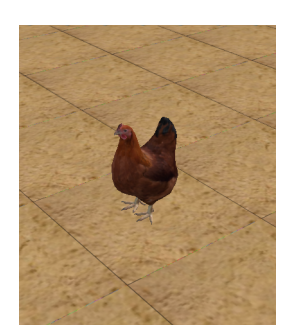
- hrát mapy uložené v balíčku,
- editovat mapy uložené v balíčku a přidat takto změněné mapy zpět do balíčku,
- využít obsah balíčku k tvorbě nových map.

Ukázková hra

Z pohledu koncového hráče je výsledkem naší práce RTS hra, obsahující obsahující například následující herní prvky:



Wolf, neboli vlk, je jednotka útočící na blízko. Tato jednotka nedokáže poškodit nepřátelské obranné budovy.



Chicken, neboli kuře, je jednotka útočící na dálku, používající vajíčka jako projektily. Oproti vlkům dokáže tento typ jednotek chodit přes vodu a poškodit obranné budovy.

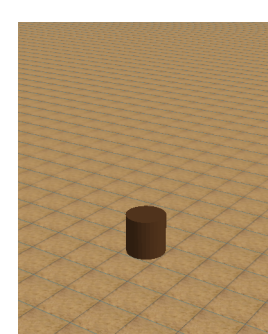


Dog, neboli pes, je jednotka vytvářená budovou dřevorubce, která se pohybuje mezi budovou a nejbližším stromem a získává tak dřevo. Tuto jednotku nelze manuálně vytvářet ani ovládat, obojí je plně automatizováno.

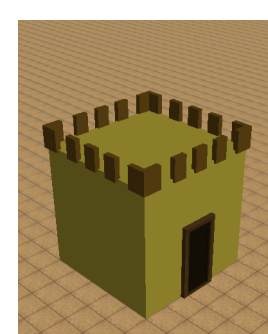
Tree, neboli strom. Stavba stromů je omezena na editaci úrovně. Rychlost růstu stromů a pravděpodobnost jejich rozmnožení je závislá na typu dlaždice, na které se nacházejí.



TreeCutter, neboli dřevorubec, umožňuje hráči získávat ze stromů dřevo. Tato budova vytváří dvě jednotky typu **Dog**, které následně pendlují mezi nejbližším stromem a touto budovou, čímž získávají pro hráče dřevo.

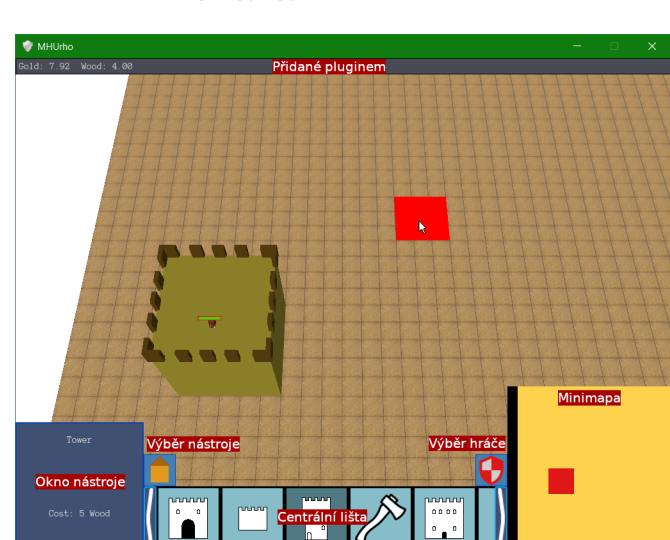


Keep, neboli tvrz. Každý hráč vlastní právě jednu budovu tohoto typu, při jejímž zničení prohrává. Cílem hry je tedy zničit protivníkovu tvrz bez ztráty své vlastní tvrze. Tvrz umožňuje pohyb jednotek po své střeše. Tvrz dále slouží pro tvorbu jednotek během hry.



Ukázková úroveň pro dva hráče, na které můžeme vidět některé z budov a jednotek popsaných výše, i jiné, které jsme zde nepopsali. Dále můžeme vidět terén, který byl vytvořen spuštěním úrovně v editačním módu.

Grafické uživatelské rozhraní v editačním módu úrovně. Množina nástrojů a obsah spodních lišt je definován tvůrcem balíčku. Horní lišta poté byla přidána čistě tvůrcem balíčku.



Ukázková hra je pouze balíček

Všechny součásti ukázkové hry, popsané v předcházející části, a mnohé další, jsou obsaženy v balíčku **ShowcasePackage**. Tento balíček je sice distribuován spolu s platformou, slouží však pouze jako demonstrace možností platformy, které může využít kterýkoli tvůrce balíčků.

Proces vytváření balíčku můžeme vidět na diagramu 1. Jednotlivé kroky jsou blíže popsány v následujících částech.

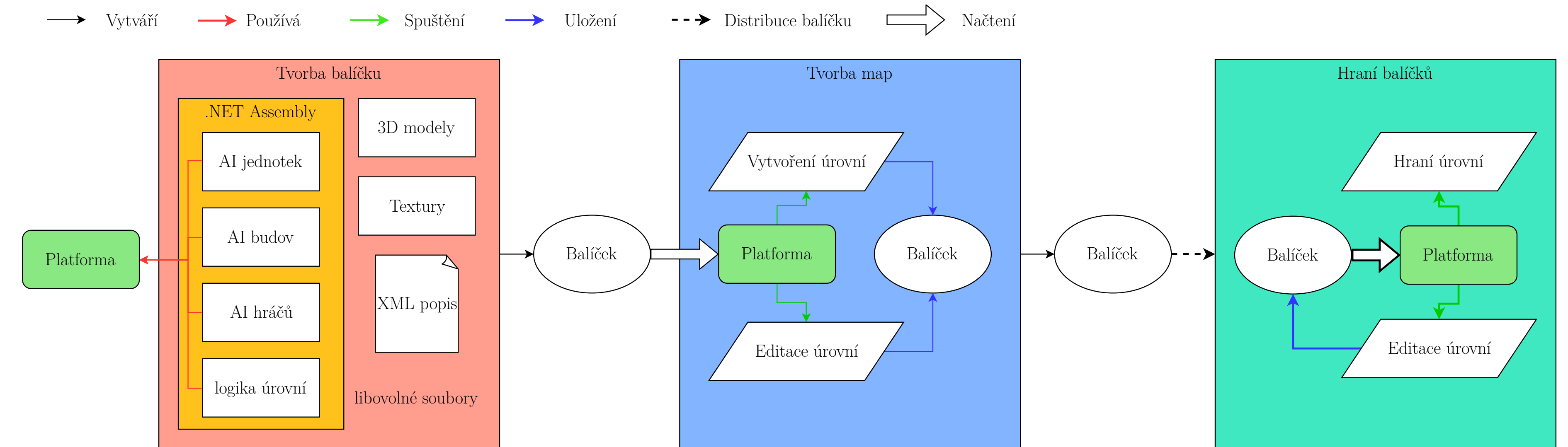


Fig. 1: Tvorba a použití balíčku.

Tvorba pluginů

Chování hráčů, jednotek, budov, projektilů či celé úrovně je definováno pomocí pluginů, jejichž tvorba probíhá za použití jazyka C#. Výsledné assembly jsou poté součástí obsahu balíčku, jak můžeme vidět na diagramu 1.

Při tvorbě pluginů je možné využít schopností .NET frameworku verze 4.7.2, naší platformy, či engineu UrhoSharp. Dále je možné využít libovolných knihoven pro .NET framework, které ale musí následně být zahrnuty do obsahu balíčku, odkud jsou poté při běhu načítány.

Plugins v platformě

Platforma využívá systém **Reflection** pro získání instancí typových pluginů, které můžeme vidět na diagramu 2. Tyto pluginy reprezentují typ hráčů, jednotek, projektilů, úrovní, či, jako v případě ukázky, budov.

Typový plugin je následně využit pro tvorbu instancí pluginů ovládajících jednotlivé herní prvky daného typu. Těmto instancím jsou směřovány události, které nastávají pro tuto instanci ovládaný herní prvek, a jejich obsluhou je implementováno chování daného herního prvku. Příkladem události může být zničení jednotky, která bude směřována na plugin ovládající hráče vlastního zničenou jednotku.

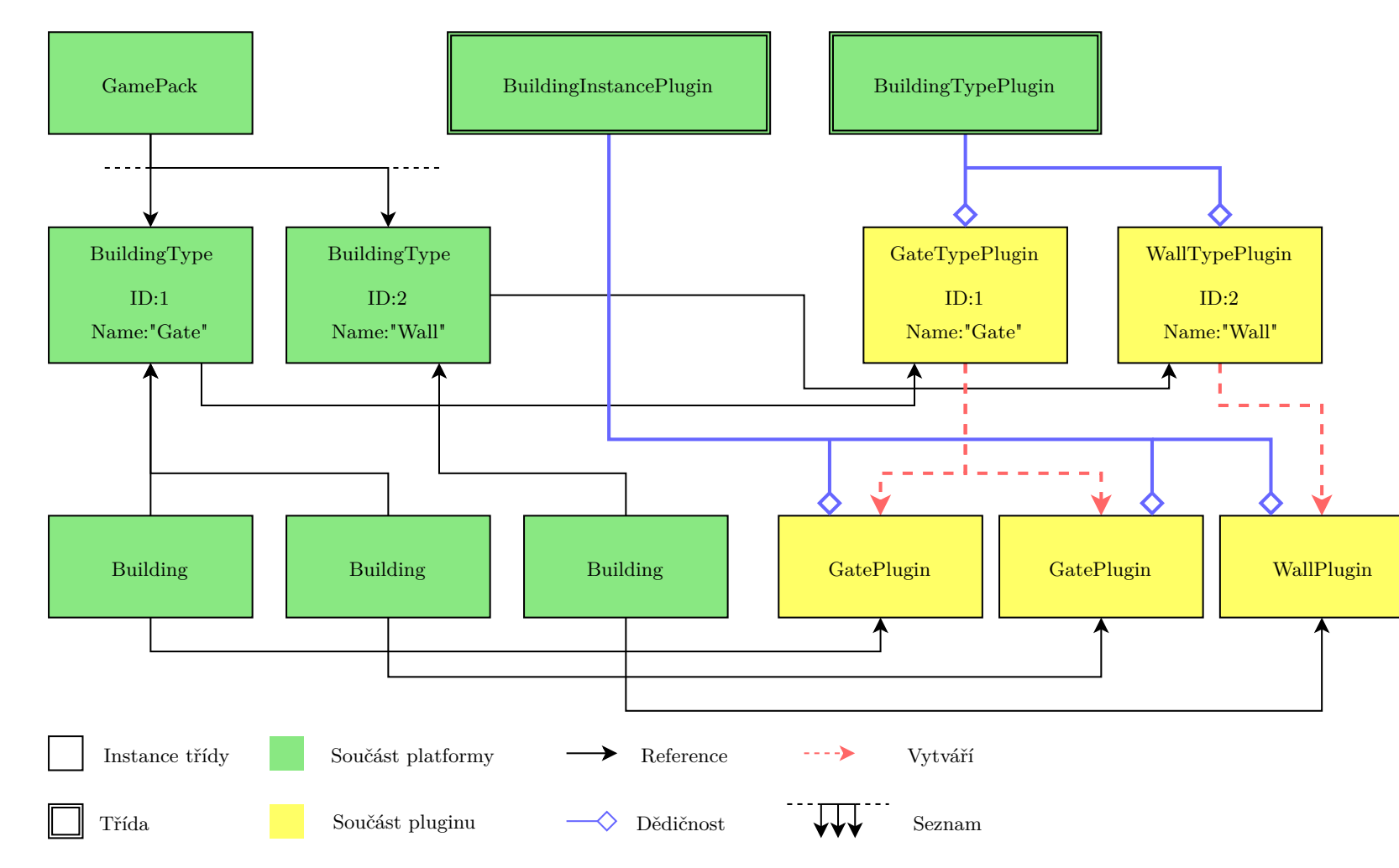


Fig. 2: Integrace pluginů do struktury platformy.

Definice typu jednotky

Příklad definice typu jednotky:

```
<unitType name="Chicken" ID="3">
  <assets type="xmlprefab">
    <path>Assets/Units/Chicken/prefab.xml</path>
  </assets>
  <assemblyPath>ShowcasePackage.dll</assemblyPath>
  <extension>
    <cost>
      <Wood>3</Wood>
      <Gold>4</Gold>
    </cost>
  </extension>
  <iconTextureRectangle left="0" top="200" right="100"
    bottom="300"/>
</unitType>
```

Definice balíčku

Balíček je popsán jediným XML souborem, který můžeme vidět na diagramu 1. Tento soubor definuje typy herních prvků, jako například jednotek, budov, projektilů, hráčů či úrovní. Příklad obsahu takovéto definice můžeme vidět v části *Definice typu jednotky*, vztah s datovými strukturami platformy pak můžeme vidět na diagramu 3.

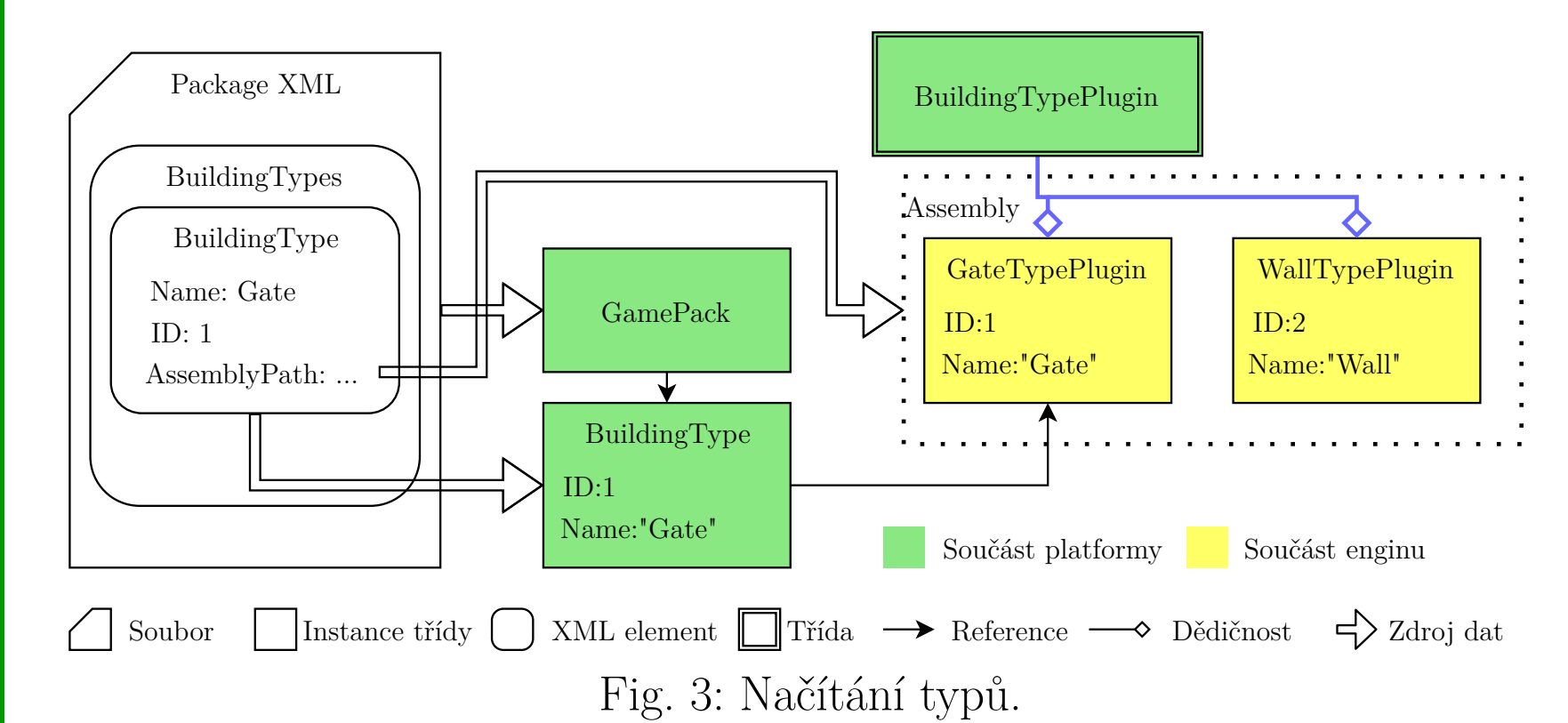


Fig. 3: Načítání typů.

Nástroje pro editaci

Součástí rozhraní instanačního pluginu úrovně je specifikace nástrojů dostupných při spuštění úrovně pro editaci či pro hraní. Platforma obsahuje implementace základních nástrojů, umožňujících například změnu typu dlaždice, jednoduché modelování terénu, vytváření a ovládání jednotek či vytváření budov.

Hlavním účelem tohoto systému je ale umožnit tvůrcům balíčků vytvořit vlastní nástroje, umožňující ovládání specifické pro jejich hru. Schopnosti nástrojů vytvořených v pluginu nejsou nijak omezeny vůči schopnostem nástrojů poskytovaných platformou.

Vytvoření úrovní

Po vytvoření balíčku je možné tento balíček nahrát do platformy a využít pluginem definované nástroje, využívající zbylý obsah balíčku, k tvorbě a editaci úrovní. Tyto nové vytvořené úrovně jsou následně uloženy zpět do balíčku.

Koncový uživatel

Po získání balíčku a přidání do platformy získává koncový uživatel přístup k obsahu balíčku. Tento obsah může následně využít pro hraní úrovní obsažených v balíčku, editaci těchto úrovní či vytvoření nových úrovní za použití nástrojů definovaných tvůrcem balíčku.



KarelMad@email.cz
<https://github.com/MK4H/MHUrho>