Goblins & Magic

by Hippo Games

TABLE OF CONTENTS

1 About

Creating your own mobile game is easier than you think! No matter what it will be - dungeon RPG, platformer or turn-based strategy, this package is very flexible and fits to everything! Create your game prototype quickly with my starter kit! All characters are prototyped with build-in Unity primitives, so it's very simple to modify them and create new ones. Simple animation examples will help you to explore basic Unity animation technique. Learn how to make magic effects made with default particle system look amazing! This package also contains AI implementation for creature's behavior. All C# code is designed and refactored according to Microsoft .Net guidelines.

2 How this package works

This package is game prototype. You'll find basic concepts and mechanics here and will learn:

- How to animate characters
- How to organize character structure
- How to use physics
- How to create and control your hero
- How to create monsters and add their behavior (AI)
- How to create game map using tiled objects
- How to work with particles to create simple magic effects

You'll find different scenes in this package. Select any scene from Project window and then press Run button. Creatures scene will show you all creatures from this package. In Survival scene you can play for goblin and learn how to move, jump and attack. In WallToWall scene you can run auto battle to see how AI controls many characters at once.

3 PROJECT CONTENTS

- Pixel goblin characters: Warrior, Shooter and Shaman (animated, 3d)
- Bonus characters: Knight, Bat, Slime
- Magic spells: Fireball, Ice Bolt, Poison Glob (particle system based)
- Map blocks (mesh and tiles)
- Coin (drop & collect)
- Base audio effects
- AI (implementation sample)
- 3 demo scenes (Survival, Creatures, WallToWall)

4 SETUP & TEST GUIDE

1. Create a new empty project
2. Import the package (Assets/Import Package/Custom Package)
3. Open and add scene [GoblinsAndMagic/Scenes/Survival.scene] to Scenes in Build (Build Settings)
4. Run project in editor mode or build on any device

5 SCENES

- /Scenes/Creatures.scene - creatures and animation overview
- /Scenes/Survival.scene - survival mode, use controls to play (Arrows, Space, J)
- /Scenes/WallToWall.scene - wall to wall mode, autoplay

6 HOW TO MODIFY AND EXPAND

You can use existing characters to create the new one. Place any prefab character on the scene and modify it. You'll find all prefabs in /Resources/Prefabs. All materials are placed into /Materials folder so you can simply add new materials here.

To modify and expand animations go to /Animations folder. All creatures use default Unity animation system.

To add new map tiles go to /MapTiles folder where you can add your sprites. Then you'll need to create new materials in /Materials/MapTiles.

If you want to change or extend existing scripts or add new behavior go to /Scripts folder. Also you'll need some basic #C knowledge to do this. You'll find all scripts description in the next section of this manual.

7 SCRIPT REFERENCE

Please refer to ScriptReference.chm for more detailed information.

Common/Tweens/Rotation.cs	Rotating objects
Common/Tweens/ScaleSpring.cs	Changing object scale
Common/Tweens/TweenBase.cs	Base class for all tween classes
Common/PrefabHelper.cs	Instantiating prefabs, monsters for example
Common/Screenshot.cs	Taking screenshots from camera
Common/Singleton.cs	Base class for singleton behaviors
Data/AIParams.cs	Parameters for monsters controlled by AI
Data/AnimationParams.cs	Animation parameters
Data/Controls.cs	Common creature controls used by player and AI
Data/CreatureParams.cs	Common creature parameters
Data/CreatureState.cs	Creature state
Enums/AttackType.cs	Creature attack type
Magic/Fireball.cs	Fireball object behavior
Magic/IceBolt.cs	Ice bolt object behavior
Magic/MagicBase.cs	Base class for all magic classes
Magic/PoisonGlob.cs	Poison glob object behavior
AI.cs	Simple AI implementation
Arrow.cs	Arrow object behavior
AudioPlayer.cs	Playing music and effects
BodyBox.cs	Used for repulsion creatures from each other by Z-axis to prevent creatures overlapping
Coin.cs	Coin object behavior
Creature.cs	Common creature behavior, please take attention to this class
DeadBlink.cs	Blinking creatures between death and destroy
Engine.cs	Entry point, the main controller
Eye.cs	Make eyes in gray style after death
KeyboardController.cs	Creature control using keyboard
Poison.cs	Poison effect behavior
StatusBars.cs	Health and mana bars behavior
Throwing.cs	Base class for throwing objects (arrows, magic)
WeaponBox.cs	Used for hit detection

8 FEEDBACK

Please ask all your questions on the asset page. You can also **RATE** ★★★★★ my asset and request new features. I'll be glad to answer you!