Small Scale®
INTERACTIVE

Character pack – Documentation

This asset offers a collection of high-quality, pixel-art character sprites, designed for developers and artists seeking to inject classic, retro flair into their games. This package includes a variety of animated sprites perfect for RPGs, top-down action games, and other game genres. Accompanied by an example scene and pre-configured animators, PixelCharacters is ready to drop into your project or inspire new creations.

1. Introduction

Overview of the package

This character pack is created to be used in any pixelated, top down game. Each character has been animated to fit a wide veriaty of game ganres.

Whats included?

The package includes 8 characters with over 30 unique animations in 8 different view angles, perfect for your next top down game. The sprites are pixalated and are all 128 x 128 in size.

2. Getting started

How to import into your project

After purchasing the asset, the best way to import the package is via the package manager in the unity editor. Simply open up the package manager, search for the asset name and click on import.

3. Basic setup and requirements

The sprites in the asset does not require any specific setup and are ready to be used in your project.

4. Animator setup

Explanation of the Animator component structure

Each character has an Animator component. These animators are setup with transition between the different states for smooth animations and is mainly meant as an example. If you want to include them in your project you are free to do so, but **be adviced that they are not fully optimized for certain gameplay and not all animation clips that comes with the package are included in the animators.** These animators are mainly used for demonstration purposes.

How animations are triggered

All animations in the example animators are triggered using Booleans. There are no “trigger” parameters in the animators.

To trigger an animation, simply set the required parameters to true and when you want to exit the animation set the parameters to false again. For example the default state is “IdleEast”, if you want to go from “IdleEast” to “RunWest”, you simply set “IsRunning” to true and “MoveWest” to true. This will trigger the “RunWest” animation clip. If you want to go back to idle, you simply set “isRunning” to false, followed by the direction you want face (example “isWest” for facing west).

If you aim to use the example animators in your project its highly adviced to study the animators in the unity editor to see exactly how each transistion works.

The parameters used in the example animators are as follows:

- **isWalking** (Bool) Used for going from a state to Walking state.
- **isRunning** (Bool) Used for going from a state to Running state.

- **General direction bools, used for facing a certain direction when Moving.**
- **MoveNorthEast** (Bool)
- **MoveNorthWest** (Bool)
- **MoveEast** (Bool)
- **MoveWest** (Bool)
- **MoveSouthEast** (Bool)
- **MoveSouthWest** (Bool)
- **MoveSouth** (Bool)
- **MoveNorth** (Bool)

- **General direction bools, used for facing a certain direction when Idle.**
- **isWest** (Bool)
- **isEast** (Bool)
- **isNorthEast** (Bool)
- **isSouth** (Bool)
- **isSouthWest** (Bool)
- **isSouthEast** (Bool)
- **isNorth** (Bool)
- **isNorthWest** (Bool)

- Bools for different attacks
- **AttackAttackNorthEast** (Bool)
- **AttackAttackNorthWest** (Bool)
- **AttackAttackEast** (Bool)
- **AttackAttackWest** (Bool)
- **AttackAttackSouthEast** (Bool)
- **AttackAttackSouthWest** (Bool)
- **AttackAttackSouth** (Bool)
- **AttackAttackNorth** (Bool)
- **Attack2NorthEast** (Bool)

- **Attack2NorthWest** (Bool)
- **Attack2East** (Bool)
- **Attack2West** (Bool)
- **Attack2SouthEast** (Bool)
- **Attack2SouthWest** (Bool)
- **Attack2South** (Bool)
- **Attack2North** (Bool)
- **isAttackRunning** (Bool)
- **AttackRunNorthEast** (Bool)
- **AttackRunNorthWest** (Bool)
- **AttackRunEast** (Bool)
- **AttackRunWest** (Bool)
- **AttackRunSouthEast** (Bool)
- **AttackRunSouthWest** (Bool)
- **AttackRunSouth** (Bool)
- **AttackRunNorth** (Bool)
- **RangedAttackNorthEast** (Bool)
- **RangedAttackNorthWest** (Bool)
- **RangedAttackEast** (Bool)
- **RangedAttackWest** (Bool)
- **RangedAttackSouthEast** (Bool)
- **RangedAttackSouthWest** (Bool)
- **RangedAttackSouth** (Bool)
- **RangedAttackNorth** (Bool)
- **isAttackAttacking** (Bool)
- **isRangedAttacking** (Bool)
- **isSpecialAbility1** (Bool)
- **isSpecialAbility2** (Bool)

- **specialAbility1kNorthEast** (Bool)
- **specialAbility1NorthWest** (Bool)
- **specialAbility1East** (Bool)
- **specialAbility1West** (Bool)
- **specialAbility1SouthEast** (Bool)
- **specialAbility1SouthWest** (Bool)
- **specialAbility1South** (Bool)
- **specialAbility1North** (Bool)
- **specialAbility2kNorthEast** (Bool)
- **specialAbility2NorthWest** (Bool)

- **specialAbility2East** (Bool)
- **specialAbility2West** (Bool)
- **specialAbility2SouthEast** (Bool)
- **specialAbility2SouthWest** (Bool)
- **specialAbility2South** (Bool)
- **specialAbility2North** (Bool)
- **isCastingSpell** (Bool)
- **CastSpellNorthEast** (Bool)
- **CastSpellNorthWest** (Bool)
- **CastSpellEast** (Bool)
- **CastSpellWest** (Bool)
- **CastSpellSouthEast** (Bool)
- **CastSpellSouthWest** (Bool)
- **CastSpellSouth** (Bool)
- **CastSpellNorth** (Bool)
- **isKicking** (Bool)
- **KickNorthEast** (Bool)
- **KickNorthWest** (Bool)
- **KickEast** (Bool)
- **KickWest** (Bool)
- **KickSouthEast** (Bool)
- **KickSouthWest** (Bool)
- **KickSouth** (Bool)
- **KickNorth** (Bool)

- Bools for dying and taking damage
- **isTakeDamage** (Bool)
- **takeDamageEast** (Bool)
- **takeDamageWest** (Bool)
- **takeDamageNorth** (Bool)
- **takeDamageSouth** (Bool)
- **takeDamageNorthEast** (Bool)
- **takeDamageNorthWest** (Bool)
- **takeDamageSouthWest** (Bool)
- **takeDamageSouthEast** (Bool)
- **isDie** (Bool)
- **dieEast** (Bool)
- **dieWest** (Bool)
- **dieSouth** (Bool)

- **dieNorth** (Bool)
- **dieNorthEast** (Bool)
- **dieNorthWest** (Bool)
- **dieSouthEast** (Bool)
- **dieSouthWest** (Bool)
- **isFlipping** (Bool)
- **FlipNorthEast** (Bool)
- **FlipNorthWest** (Bool)
- **FlipEast** (Bool)
- **FlipWest** (Bool)
- **FlipSouthEast** (Bool)
- **FlipSouthWest** (Bool)
- **FlipSouth** (Bool)
- **FlipNorth** (Bool)
- **isRolling** (Bool)
- **RollingNorthEast** (Bool)
- **RollingNorthWest** (Bool)
- **RollingEast** (Bool)
- **RollingWest** (Bool)
- **RollingSouthEast** (Bool)
- **RollingSouthWest** (Bool)
- **RollingSouth** (Bool)
- **RollingNorth** (Bool)
- **isSliding** (Bool)
- **SlidingNorthEast** (Bool)
- **SlidingNorthWest** (Bool)
- **SlidingEast** (Bool)
- **SlidingWest** (Bool)
- **SlidingSouthEast** (Bool)
- **SlidingSouthWest** (Bool)
- **SlidingSouth** (Bool)
- **SlidingNorth** (Bool)
- **isPummeling** (Bool)
- **PummelNorthEast** (Bool)
- **PummelNorthWest** (Bool)
- **PummelEast** (Bool)
- **PummelWest** (Bool)
- **PummelSouthEast** (Bool)
- **PummelSouthWest** (Bool)
- **PummelSouth** (Bool)
- **PummelNorth** (Bool)
- **isAttackSpinning** (Bool)

- **AttackSpinNorthEast** (Bool)
- **AttackSpinNorthWest** (Bool)
- **AttackSpinEast** (Bool)
- **AttackSpinWest** (Bool)
- **AttackSpinSouthEast** (Bool)
- **AttackSpinSouthWest** (Bool)
- **AttackSpinSouth** (Bool)
- **AttackSpinNorth** (Bool)
- **isCrouchIdling** (Bool)
- **CrouchIdleNorthEast** (Bool)
- **CrouchIdleNorthWest** (Bool)
- **CrouchIdleEast** (Bool)
- **CrouchIdleWest** (Bool)
- **CrouchIdleSouthEast** (Bool)
- **CrouchIdleSouthWest** (Bool)
- **CrouchIdleSouth** (Bool)
- **CrouchIdleNorth** (Bool)
- **isCrouchRunning** (Bool)
- **CrouchRunNorthEast** (Bool)
- **CrouchRunNorthWest** (Bool)
- **CrouchRunEast** (Bool)
- **CrouchRunWest** (Bool)
- **CrouchRunSouthEast** (Bool)
- **CrouchRunSouthWest** (Bool)
- **CrouchRunSouth** (Bool)
- **CrouchRunNorth** (Bool)
- **isStrafingLeft** (Bool)
- **StrafeLeftNorthEast** (Bool)
- **StrafeLeftNorthWest** (Bool)
- **StrafeLeftEast** (Bool)
- **StrafeLeftWest** (Bool)
- **StrafeLeftSouthEast** (Bool)
- **StrafeLeftSouthWest** (Bool)
- **StrafeLeftSouth** (Bool)
- **StrafeLeftNorth** (Bool)
- **isStrafingRight** (Bool)
- **StrafeRightNorthEast** (Bool)
- **StrafeRightNorthWest** (Bool)
- **StrafeRightEast** (Bool)
- **StrafeRightWest** (Bool)
- **StrafeRightSouthEast** (Bool)
- **StrafeRightSouthWest** (Bool)

- **StrafeRightSouth** (Bool)
- **StrafeRightNorth** (Bool)
- **isRunningBackwards** (Bool)
- **RunBackwardsNorthEast** (Bool)
- **RunBackwardsNorthWest** (Bool)
- **RunBackwardsEast** (Bool)
- **RunBackwardsWest** (Bell)
- **RunBackwardsSouthEast** (Bool)
- **RunBackwardsSouthWest** (Bool)
- **RunBackwardsSouth** (Bool)
- **RunBackwardsNorth** (Bool)

5. Example scene

Overview

The asset comes with an example scene where you can preview the animations for each character. The scene included almost all main animations for each character.

You can change character by using the toggles to the left. These toggles sets each characters sprite renderer to false, essentially hiding them so only the selected character is shown. To move around the map, use the following keys:

W key: moves the character forward in the direction of the mouse cursor.
A and D key: makes the character strafe left and right, perpendicular to the mouse cursors position.
S Key: moves the character in the opposite direction from the mouse cursor.

Right mouse button: This triggers the regular attack of the selected character
Left Shift key: This triggers the roll animation
Left Ctrl key: this triggers the front flip animation
Left Alt key: this triggers the slide animation

1,2,3,4,5,6 key: these are used for the abilities of the character.

6. Customization

How users can customize the sprites

All animations are based on single sprites fired in sequence. There are no sprite sheets and each animation clip contains 15 sprites each. For a lower refresh rate, you can reduce the number of sprites in your animation to fit your needs. For example removing every other frame in the animation would make the animation less fluent. All sprites are dividable by 2 so you can reduce the sprites size to fit your need

Optimization

Before building your game, its highly adviced to pack the sprites into atlases to increase performance and reduce file size. Other performance tips is to reduce the number of frames in the animations, this will however reduce how fluent the animations look.

7. Contact

If you want to reach out you can send an email to: smallscaleinteractive@gmail.com