

---

# WorldDoc

*Release 0.1*

TwoBitMachines

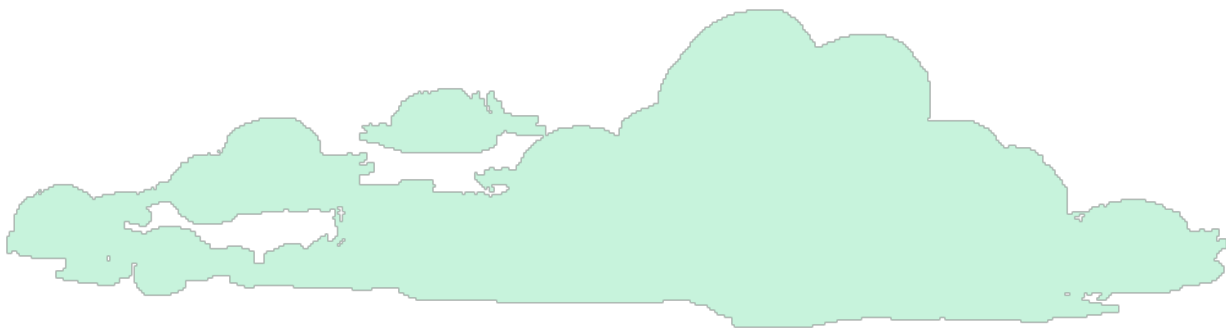
Apr 26, 2022

## INTERACTABLES

1	Bridge	1
2	Rope	3
3	Water	5
4	Ladder	7
5	High Jump	8
6	Foliage	9
7	Jump	11
8	Dash	12
9	Hover	13
10	Swim	14
11	Ladder	15
12	Rope	16

## BRIDGE

A bridge creates dynamic movement. Characters can walk and jump on them.



---

**Tip:** Characters, by default, are enabled to interact with bridges. If this property is not desired, disable it in the character's collision settings to save unnecessary collision checks.

---

Property	
Planks	The number of planks in the bridge.
Gravity	The force of gravity acting on the bridge.
Bounce	The force exerted on the bridge when interacting with characters.
Stiffness	The larger the number, the less sag the bridge will have. For performance, keep this value below 20.
Plank	Create a gameobject and add the plank's sprite. Make this a child of the bridge gameobject and set the reference. This will be used as a template to instantiate the remaining planks. Change the transform's scale to achieve the desired plank width. Lastly, the offset will shift each plank visually.
Area	The system will check for plank collisions once the character is inside the bridge area. The area width is set automatically, but the height must be specified. The offset will offset the area in the y direction.
Create	Once all the settings are chosen, press this button to create the bridge. Anytime you change the bridge's position or a setting, recreate the bridge to enact the changes.
View	If enabled, the bridge gizmos will be visible.

---

**Important:** The start of the bridge corresponds to the transform's position. Make sure the transform's handle position is set to Pivot (and not to Center) for proper placement. A scene handle tool, a red circle, is used to specify the end of the bridge. The distance between the start and end points determines the length of the bridge.

---

## ROPE

The player can use ropes to swing or for simple idle interactions.

---

**Note:** The player's Rope ability must be enabled to interact with ropes.

---

Property	
Type	If Swing is enabled, the player will swing on the rope. If Idle is enabled, the player will pass through the rope, causing it to move.
Rope End Radius	If Swing is enabled, once player and end tether are within this radius, the player will latch onto the rope automatically.
Rope Radius	If Idle is enabled, the system will check for rope collisions if the player is inside this radius. The center of this radius is set automatically.
Tether Radius	If Idle is enabled, it is the radius of each tether used to detect the player.
Force	If Idle is enabled, it is the movement force applied to a tether upon interaction.
Tethers	The number of tethers in the rope.
Gravity	The force of gravity acting on the rope.
Stiffness	The larger the number, the less sag the rope will have. For performance, keep this value below 20.
Double Anchor	Both the start and end of the rope are anchored.
Rope Start	Create a gameobject and add the tether's sprite. Make this a child of the rope gameobject and set the reference. This will be used as a template to instantiate the remaining tethers. Change the transform's scale to achieve the desired tether height;
Rope End	Every time the rope is created, it destroys and recreates all the tethers. Sometimes the end tether contains components like Health. To prevent having to add these components every time the rope is recreated, specify the end tether gameobject to prevent it from being destroyed.
Create	Once all the settings are chosen, press this button to create the rope. Anytime you change the rope's position or a setting, recreate the rope to enact the changes.
View	If enabled, the rope gizmos will be visible.

---

**Important:** The start of the rope corresponds to the transform's position. Make sure the transform's handle position is set to Pivot (and not to Center) for proper placement. A scene handle tool, a red circle, is used to specify the end of the rope. The distance between the start and end points determines the length of the rope.

---

Method	
ApplyImpactAtEnd (float directionX, float impact)	This will apply an impact force in the x direction to the end of the rope. This is automatically used by the player for swinging.
ApplyImpact (float value, Vector2 direction)	Each tether contains the component Tether. This class contains this method. Call it to apply a force to a tether in the specified direction. Ignore the value parameter and instead set the impact force in the inspector field of the Tether class.
UnlatchEndAnchor ( )	If double anchor is set true, you can set it false by calling this method. The end anchor will become free, letting the rope fall down.

---

**Tip:** It's possible to add a Health and Collider component to each tether for further interaction. This can be useful if the rope needs to collide with Projectiles. The Health component is equipped to call the ApplyImpact() and UnlatchEndAnchor() methods through Unity Events.

---

## WATER

Water is a dynamic area where the player can float and swim.

---

**Note:** The player's Swim ability must be enabled to interact with water.

---

Property	
Shape	If Square is enabled, the system renders the water using square blocks. No textures or sprites are required. If Round is enabled, the system renders the water using a Mesh Renderer, creating a more curved wave. A Texture2D and Material are required.
Type	If Float is enabled, the player will stay above the water line. If Swim is enabled, the player can swim inside the water.
Segments	The number of elements that create the water. The higher the number, the less blocky the water will look.
Texture2D	If Shape mode is Round, provide the Texture2D that will be used to render the water.
Material	If Shape mode is Round, provide the Material that will be used by the Mesh Renderer.
Amplitude	The maximum height of the wave.
Frequency	Dictates the number of waves in the water.
Speed	How quickly a wave moves across the water.
Spring	The force exerted on the water when interacting with the player.
Damping	How quickly the spring force dissipates.
Turbulence	This adds random noise into the water, creating a chaotic effect.
Random Current	This will change the direction of the speed at intervals specified by this value. This value is randomized slightly to add unpredictability.
Create	Once all the settings are chosen, press this button to create the body of water. Anytime you change the water's position or a setting, recreate the water to enact the changes.

Body	For Square mode.
Top	The color of the water line.
Thickness	The thickness of the water line.
Taper	The wave's water line will be thicker at its crest, and thinner at its trough.
Middle	The color at the middle of the water.
Bottom	The color at the bottom of the water.
Phase	The bottom of the water has wave like motion as well. Specify the phase of this wave.
Offset	Offset the position of the bottom wave.
Speed	How quickly the bottom wave moves across the water.



---

## CHAPTER FOUR

---

### LADDER

The humble ladder is used for climbing.

---

**Note:** The player's Ladder ability must be enabled to interact with ladders.

---

Property	
Size	The width and height of the ladder.

## HIGH JUMP

Launch a character into the air upon contact.

Property	
Jump Force	The amount of force launching a character.
Radius	The radius of the collision circle. Once collision is made, the character will be launched. The offset will change the collision's center position in the y direction.

## FOLIAGE

Decorate an environment with foliage to make it come alive. The foliage can sway with the wind and respond to character movements.

Property	
Jiggle	The motion effect produced when interacting with characters. Smaller values produce softer motions.
Damping	How quickly the jiggle effect dissipates.
Uniformity	The tendency for foliage to sway in the same direction if the foliage has the same y position.
Wind Strength	The force of the wind swaying the foliage.
Wind Frequency	How quickly the wind changes direction.
Create Texture	Press this button to add a new Texture2D, which represents the foliage. Each Texture2D must have the same size as the specified Vector2 field, or else the tool will not work.

**Warning:** The system groups all the Texture2D images of the foliage into an array. Thus, every Texture2D must be of the same size and share the same settings for this process to work correctly. As a reminder, this component is working with Texture2D and not Sprites.

Texture2D	
Texture2D	The current Texture2D image of the foliage. The delete button will remove this Texture2D and all of its instances from the scene.
Orientation	This determines what vertices to sway. If Bottom is enabled, place foliage on ground. If Top is enabled, place foliage on a ceiling. If Left or Right are enabled, place foliage on walls.
Depth	Specify the rendering order of the Texture2D images relative to each other. As of now, there is no way specify a sorting layer. The player is either in front or in back of the foliage – never in between.
Interaction	Choose how active the foliage is with character interactions. Maybe some foliage are dense and don't need to sway as much as others. A value of zero will disable all interactions with characters.

Paint Brushes	Place foliage in the scene with brushes.
Single Brush	Place a single foliage image.
Random Brush	Choose as many foliage images as desired and drag the brush in the scene. The density value specifies how many images the brush can place per position.
Eraser	Use this brush to erase foliage images.
Instances	Every Foliage component can only have a maximum of 1023 images in the scene.

---

**Tip:** If the brush tool is active, right click in the scene or repress the current brush button to deactivate it.

---

---

**Note:** The foliage system was designed with performance in mind. All foliage instances exist in code only (they're not gameobjects), and the character interactions are handled by Unity's Job System.

---

## JUMP

The most fundamental ability in any platformer.

---

**Note:** Jump height and jump time are set in the Collision settings in order to calculate the force of gravity. However, this ability must still be enabled if the player is required to jump.

---

Property	
Button Trigger	On user button interaction, choose exactly when the player jumps.
Min Jump Height	If this value is greater than zero, the player will have a variable jump height, and min jump will be the lowest jump height possible.
Air Jumps	The number of extra jumps the player can perform in the air.

---

**Important:** If an ability already contains a jump force, do not add the Jump ability as an exception. For instance, the Wall ability contains a few jumping options that it will execute internally. The Jump ability is geared for jumping on ground.

---

**DASH**

Increase the speed of the player to quickly cover distance.

Property	
Buttons	The buttons that need to be tapped in order to trigger a dash.
Dash Direction	If Horizontal Axis is enabled, the dash will occur along the x axis. In this state, only the left and right buttons are used. It is also possible to use only one button and leave the other empty. If Multi Directional is enabled, all the buttons that are set will be utilized to move the player in one of eight directions along the x and y axis.
Button Taps	If Single Tap is enabled, pressing the button only once will trigger a dash. If Double Tap is enabled, pressing the button twice is required to trigger a dash.
Tap Threshold	If Double Tap is enabled, the threshold is the time interval in which the double tap must occur for the dash to trigger successfully.
Duration	If Instant is enabled, the player will traverse the dash distance in one frame. If Incremental is enabled, the player will traverse the dash distance according to the dash time.
Dash Time	The time it will take to traverse the dash distance;
Dash Distance	The total distance traversed while dashing.
Cool Down	The time interval before the next dash can be triggered.
On Ground Only	If enabled, the player must be on the ground in order to begin a dash.
Cool Down	If enabled, the force of gravity will not affect a dash.

## HOVER

Escape gravity by letting the player hover in the air.

Property	
Thrust	The forced used to propel the player upward. This force will be proportional to the jump force.
Maintain	The tendency for the player to remain in the air. A value of one will prevent the player from descending downward, unless the descend button is pressed.
Thrust Button	Press this button to create thrust.
Descend	The force that will drive the player downward. The descend button is optional. If it's not used, the player will descend on its own according to the maintain value.
Descend Button	Press this button to create downward thrust.
Exit	If On Ground Hit is enabled, the player will exit the hover state when the player touches the ground. If Button is enabled, the player will exit the hover state when the specified button is pressed.
Air Friction X	The air resistance applied to the player while hovering in the x direction.
On Thrust	Every time the thrust button is pressed, this Unity Event will be invoked.
On Descend	Every time the descend button is pressed, this Unity Event will be invoked.

**SWIM**

Allow the player to swim or float on any body of water. The body of water will determine if the player either floats or swims. If floating, the player will remain above the water line. If swimming, the player will swim inside the body of water.

---

**Note:** The player must have the Swim ability enabled to interact with water.

---

Property	
Spring	If floating, when the player enters the water, it will oscillate on the water line before coming to a rest. This force dictates how quickly the oscillations occur.
Damping	How quickly the spring force dissipates.
Weight	How quickly the player sinks while swimming.
Water Impact	The force exerted on the water upon entry. The force exerted while the player moves in the water will be proportional to this value and the player's velocity.
Water Friction X	Water resistance applied to the player in the x direction.
Water Friction Y	Water resistance applied to the player in the y direction.
Jump	The force used to jump out of the water.
Switch Button	If water Switch Type is set to Yes, holding this button will transition the player from a floating state to a swimming state. To return to a floating state, the player must reach the top of the water.
On Enter Water	On water entry, a Unity Event containing the entry position is invoked. This could be useful for adding particle effects.
On Exit Water	On water exit, a Unity Event containing the exit position is invoked.



## LADDER

The player can interact with ladders.

Property	
Latch	If Automatic is enabled, the player will automatically latch to the ladder on contact, provided the player has a negative y velocity and a zero x velocity. If Enter Button is enabled, specify the button that must be pressed in order for the player to latch onto the ladder.
Climb	If Manual is enabled, specify the buttons (Up, Down) for climbing the ladder. If Automatic is enabled, the player will climb the ladder automatically.
Climb Speed	How quickly the player climbs the ladder.
Stand On Top	If enabled, the player can stand on top of the ladder.
Align To Center	If enabled, the player's x position will align with the center of the ladder.

---

## CHAPTER TWELVE

---

### ROPE

The player can interact with ropes.

Property	
Swing Strength	The force added to the swing motion.
Jump	If latched, the force used to jump away from the rope.