Buildbox™ 2.0

User Manual

This document is current for version 2.0 of Buildbox™

Contents Summary

Contents

Buildbox™ Screens

Welcome

This is the first screen you will see in Buildbox™.

This is the first screen you will see when opening the software. You can also access it via the menu bar (Help -> Welcome Screen)
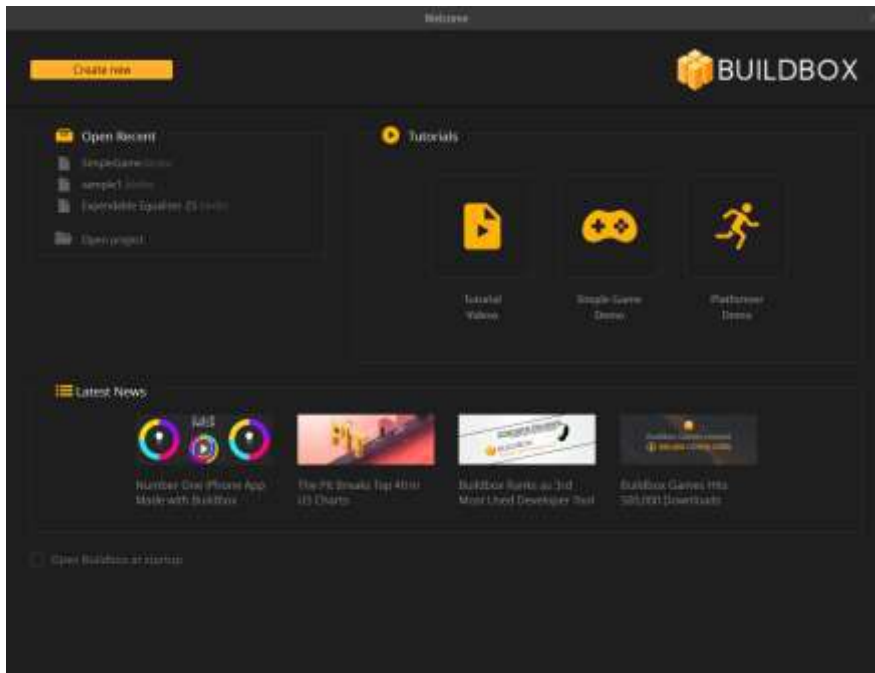
There are and three main three sections:

- **Open Recent:** this is a list of your recently saved Buildbox™ projects. The last item brings up a dialog to browse to any project on your computer.

- **Tutorials:** These may change over time. A time of this document, there is a button for Tutorial Videos that takes you to the website, and two example tutorial games.

- **Latest News:** Latest news from our website. Click on one to view in a browser.

You will also see the [Create New] button that shows the Creator screen for a new project, and a tick box down the bottom to have BuildBox™ open every time your boot up your computer.

Creator

This is what you see when you request a new game in Buildbox™.
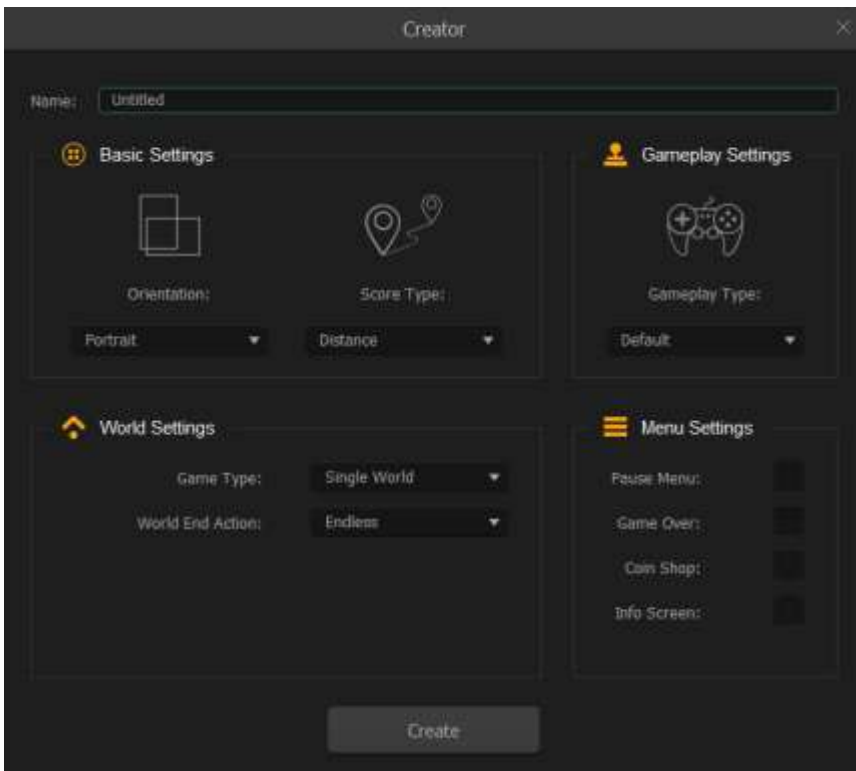
Set the following options, then press the [Create] button. All of these can be changed later – they are however the fastest way of getting your base settings correct before you start. Pay particular attention to the menu and world settings, as they setup a lot of elements, automatically, that you would otherwise have to set up by hand:

- **Name:** Name of the project.

- **Basic Settings:**

  - **Orientation:** which way is "up"

    - **Portrait**

    - **Landscape**

  - **Score Type:**

    - **Distance:** Score based on distance travelled in game.

    - **Coins Collected:** Score based on coins collected.

    - **Points Collected:** Score based on points collected.

- **Gameplay Settings:** The following presets are available – they will be used for physics settings on every World when the project is created. Each option, excluding Default, comes with an example level to get you started:

  - **Default**

  - **360 Shooter:** space shooter where you can shoot in any direction like Smash TV.

- o **Around the World:** an orbiting game where you jump over objects as you circle a globe.

- o **Avoidance:** avoidance game. Drag character around screen or use arrow.

- o **Dogfight:** vertical shooting game.

- o **Downward Bounce:** like Jupiter Jump, where player jumps down when button pressed.

- o **Fall Buttons:** pitfall style, with left and right controls

- o **Fall Switch:** pitfall style, with single touch control to change direction between left and right.

- o **Flappy:** Like Flappy Bird.

- o **Gravity Portal:** side-scroller where player controls two characters at once, and a touch of the screen swaps gravity between up and down.

- o **Gravity Runner:** side-scroller where a touch of the screen swaps gravity between up and down.

- o **Impossible:** like the Impossible Game where game scrolls from right, and the main user control is jump.

- o **ISO Jump Slide:** isometric slide and jump game. Tap screen to jump.

- o **ISO Jump:** isometric slide and jump game. Tap screen to jump.

- o **Jetpack:** like Jetpack Joyride or similar.

- o **Jumping:** like Twee Jump or Jump Pack. Jump up the platforms.

- o **Motorcross:** Physics-driven vehicle game where the user controls acceleration/deceleration.

- o **Platformer:** Classic platformer where you move around and jump over platforms in a side scrolling world.

- o **Racing:** classic vertical car racing game like Spy Hunter

- o **Runner:** sideways running game, without shooting.

- o **Shooting Runner:** sideways running game, with shooting, like Robot Unicorn Attack.

- o **Side Shooter:** classic space shooter game where enemies come from the right and there is no gravity, like R-Type.

- o **Stage Clear:** Move around and collect all coins to complete stages of the game.

- o **Stick Jump:** side scroller where the game field scrolls when the Character is in the air, and remains mostly stuck otherwise.

- o **Wall Jump:** a vertical version of Gravity Runner. Character moves up the screen, and will move to the left or right till it hits a wall. A tap on the screen will change direction.

- o **Wall Reverse:** Character jumps repeatedly off the air by tapping the screen, and must avoid the obstacles as it bounces from wall to wall till it reaches the top.

- o **Zig Zag:** Similar to Wall Jump, but the Character is moving diagonally upwards all the time. Tap to swap movement direction between left to right.

- **World Settings:**

  - o **Game Type:**

    - **Single World:** (Buildbox™ mode) A single game world only.

    - **Multi World:** More than a single game world to select and/or move through.

  - o **World Amount:** Will only appears if Multi World selected. The number of worlds to create. Values 1 to 10.

  - o **World End Action:**

    - **Endless:** game never ends

    - **Next World:** (only if Multi world selected) Move to next world.

    - **World Select:** (only if Multi world selected) Take user to screen to select next world to play.

    - **End Scene:** put in a game over scene.

  - o **Single UI for World:** (only if Multi World selected) Tick for a single UI, or leave un-ticked for a separate screen for each World.

  - o **Menu Settings:** screens to create in your initial game

    - **Pause Menu:** shows when game is paused.

    - **Game Over:** shows when player dies.

    - **Coin Shop:** allows purchase of in game currency.

    - **Info Screen:** to show information about the game – like instructions.

Scene Editor

This is the screen you will be spending most of your time in, while using Buildbox™. It is where you edit the scenes that make up a World.
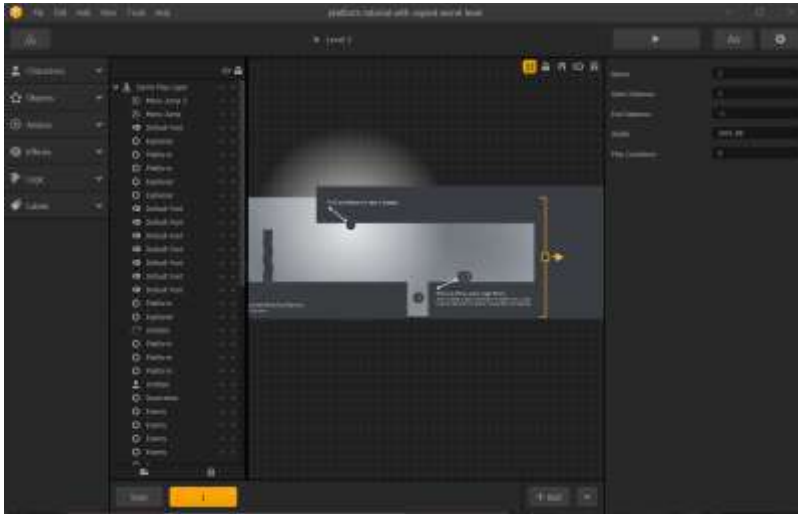
At the top-right of the central second of the Scene Editor you will see four buttons:

- **Show Game Frame:** overlay the game frame over the current Scene in yellow.
- **Lock Backgrounds:** Lock all background elements. Note: you can lock and unlock individual background elements though the Scene Tree.
- **Snap Movement:** snap to a grid on object placement, to make things easier to line-up.
- **Activate Connection Mode:** See Connections
- **Debug Mode:** Show additional information on screen.

Scene Tree

The second column of the Scene editor shows a tree of all items in the scene. You can select any item in the scene by clicking on it in the tree. By default all items are listed in the order they were created. You can rearrange the tree by clicking and dragging an item.

All non-background items are under Game Play Layer at the top. Press the trash icon down the bottom to delete the currently selected item.

There are two columns to the right, one with an eye symbol, and the other with a lock symbol. Next to each item there are two dots – that line up with those symbols. Clicking those tots will toggle their setting.

The dots under the eye you can use to suppress display of an item. The dots under the lock symbol will allow you to lock an item down so you can't accidentally move it.

Group Layer

Press the folder icon down the bottom to create a Group Layer at the current location (you will be prompted for a name). A Group Layer cannot be embedded in another Group Layer. There are three you can see on the right when a Group Layer is selected:

- **Auto Depth Sorting:** Turn on the options, below. The further up the selected Axis, the object will either ascend or descend on the Z axis ie move toward or away from the screen toward the viewer.
- **Sort Direction:**
  - **X Axis:** change Z order/depth based on movement down the X axis.
  - **Y Axis:** change Z order/depth based on movement down the X axis.
- **Sort Order:**
  - **Ascending:** Move toward the viewer based on the selected axis, above.
  - **Descending:** Move away from the viewer based on the selected axis, above.

Drag-Drop to Create New Items

The easiest way to add a character, objects, Actions and backgrounds is to drag and drop a PNG file onto the scene editor.

As soon as the mouse is over the scene editor you will see the following overlay. Drop the PNG file onto the relevant segment. You should see the item appear in the asset bar on the left, and also appear in the current scene you are editing.

Background

To lock or unlock the background, press the padlock button at the top of the main scene edit area. When the padlock is not highlighted, you can select and edit the background elements.

The following options are available:

- **Position(x,y):** object position in scene.

- **Rotation:** rotation of the background image.

- **Scale(x,y):** allows resizing and distortion of the background image.

- **Opacity:** how transparent. 1 is fully opaque and 0 fully transparent.

- **Image:** this is a PNG dropbox.

- **Speed:** the speed the background will move. Different speeds on different background layers will show a nice parallax effect.

- **Tiling Offset:** not usually needed, however this lets you change the point at which the background image is butted against the next section for tiling.

- **Fixed Rotation:** keep tiling images straight up and down even as the background is rotated. To see this effect it is best to play with a rotated image and see what happens with the setting ticked or not.

Keep in the mind the game direction setting when set up your background.
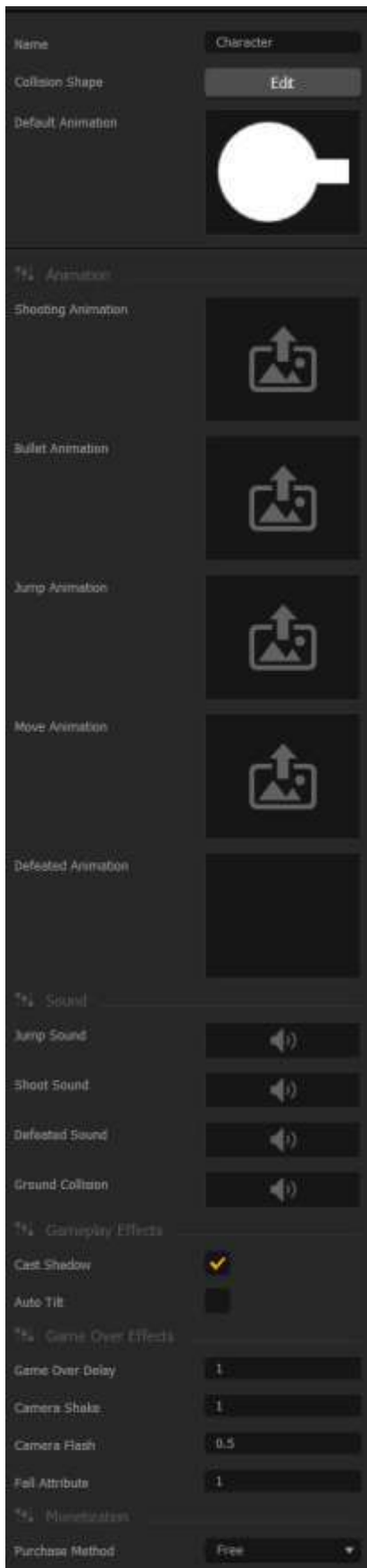
Character

Buildbox™ supports multiple characters the player uses interacts with. To set the initial position of the character, view the start scene, and drag the player to your preferred position - which can be out of view if you want to have the player drop into view or something similar.

All other settings are done by opening **Character** up on the let-hand toolbar, and clicking on the character found there. These options you can set are found on the right, once the character is selected:

- **Name:** this is for your reference only, and will appear on the character selection screen.

- **Collision Shape - Edit:** set the collision shape used by the character. See the Collision Shape Editor.

- **Default Animation:** this is the sprite that gets populated initially. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Shooting Animation:** The animation to run when Character is shooting. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Bullet Animation:** The animation to show on the bullet object that is spawned when the Character fires. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Jump Animation:** The animation to run when Character is jumping. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Move Animation:** The animation to run when Character is moving. Why is this here? Well it is is useful if you have a game with a bipedal character can stand still, or run. In which case the default animation will be the character standing still, and this animation will show the character running. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Defeated Animation:** The animation to run when Character will die. PNG Drop Box, however clicking on the pencil opens the Animation Editor.

- **Jump Sound:** MP3 file – played when jumping.

- **Shoot Sound:** MP3 file – played when shooting.

- **Defeated Sound:** MP3 file – played when character dies

- **Ground Collision:** MP3 file - played when character collides with the ground.

- **Gameplay Effects**

  o **Cast Shadow:** if player will cast a shadow when struck from a light from a light effect.

  o **Auto-Tilt:** the character will tilt in the direction of movement.

- **Game Over Effects:** - effects to play when character dies.

  o **Game Over Delay:** seconds to delay starting game-over effects.

  o **Camera Shake:** seconds to shake the game display, or 0 to disable.

  o **Camera Flash:** seconds to show camera flash effect, or 0 to disable.

  o **Fall Attribute:** a non-zero value will cause the Character to fall off the bottom of the screen (assuming gravity is in a down direction). The number will determine how gar up (against gravity) the Character will bounce up before falling.

- **Monitization:**

  o **Purchase Method:** This determines how a player will be able to access this character.

    ▪ **Free:** Available by default. You should obviously have at least one Character set to this so the beginning player can play the game.

    ▪ **In Game Currency:** Can be bought with 'coins' earned in the game.

    ▪ **In App Purchase:** Can be purchased with IAPs from an app store.

    ▪ **Rewarded Videos:** Can be purchased with a rewarded video view. Gives another list if selected – of rewarded video suppliers:

    ▪ **Add Network:** - shown if Rewarded Videos selected, above.

      • **HeyZap:** This is an ad network that allows you to select the best ads in terms of earnings across various networks. Useful for rewarded videos given that inventory across ad networks can be low.
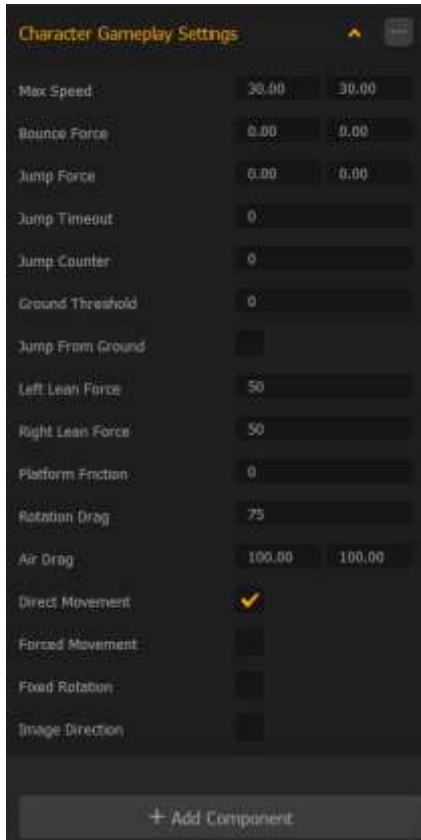
- **Character Gameplay Settings:**

  o **Max Speed (x and y):** max speed the character can move in.

  o **Bounce Force (x and y):** force the Character will exert on another object. Note: values here will constrain the movement of the player in an unrealistic way - eg with a vertical bounce force, hitting an angled surface will bounce you straight up and not sides - usually this is what you require. That is why there is an X and Y value to specify. If you require more standard physics for your bounce, leave this at 0,0 and put a value for World Bounce force. Also this will not affect objects attached to the character.

  o **Jump Force (x and y):** force on character when you press the jump button.

  o **Jump Timeout:** is used for variable jump. A value of 0.01 gives consistent jump, higher values give option to have a variable height jump depending on how long button is held down. Note: this was incorrect in previous version of the User Manual – functionality in BuildBox™ has not changed.

  o **Jump Counter:** how many jumps can be made before touching the ground. Note: this was incorrect in previous version of the User Manual – functionality in BuildBox™ has not changed.

  o **Ground Threshold:** Detection of ground is used for allowing Jump. If you have platforms with angles the character may sit across, it may be too far above ground to jump – even though parts of the character are still on the ground. You can fix this by increasing this value.

  o **Jump from Ground:** can the player jump off the game-field boundary.

  o **Left Lean Force:** force that will lean/rotate character left when lean-left button pressed.

  o **Right Lean Force:** force that will lean/rotate character right when lean-right button pressed.

  o **Platform Friction:** friction force that will stop character sliding against platforms. Note: this will not affect objects attached to the character.

  o **Rotation Drag:** force that will slow down any rotation of the character.

  o **Air Drag (x and y):** force that will slow down the character's movement over time.

  o **Direct Movement:** when checked, character can be moved around the screen by directional controls.

- o **Forced Movement:** if ticked, them a movement of the background will force the player character to move – keeping it in the same position on screen. If not ticked, then the character – unless being moved by the person playing the game – will stay fixed on the scenery, causing it to move toward the edge of the screen.

- o **Fixed Rotation:** player will not rotate no matter what happens.

- o **Image Direction:** point character in the direction of movement.

Character Custom Components

At the bottom of the character properties section, is a button – [+ Add Component]. There are several options.

Advanced Move

Options avilable are:

- **Name:** for your reference only.
- **Collision Shape:** The collision shape that will be used to trigger this.
- **Animation:** Animation to play when this is triggered.
- **Linear Velocity(x,y):** Velocity vector to add to the Character.
- **Angular Velocity:** Angular velocity to add to the Character.
- **Event:** - triggering event for this move
  - **None:** not triggered.
  - **On Start:** As soon as Character appears.
  - **Collision:** When collision shape is hit by another object.
- **Timeout:** how many seconds after triggering does this move stop.

Damage

- **Damage:** Amount of damage inflicted on another object.
- **Damage Delay:** How long till the above damage is inflicted.

Health

- **Health:** The amount of health this character has to lose before dying.
- **Lives:** How many times this character has to lose before dying completely.

Rotation Mapper

- **Animation:** The animation to show while rotating.

Ghost Jump

Jump typically used for isometric games.
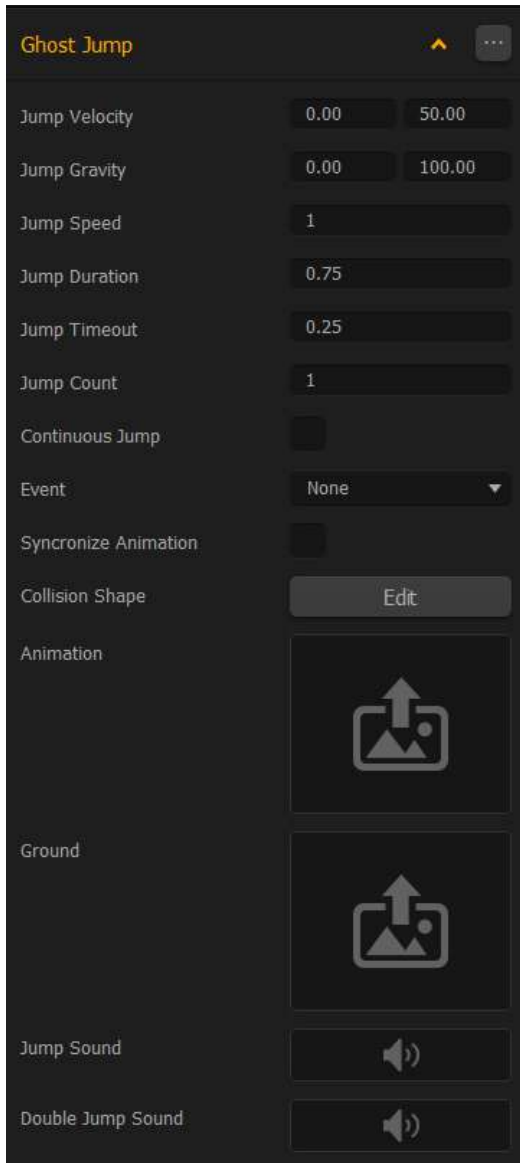
- **Jump Velocity(x,y):** Velocity vector to add to current movement, when jumping.
- **Jump Gravity(x,y):** Gravity imposed on jump while in action.
- **Jump Speed:** Values greater than 1 will be faster then normal, less than 1 will be slower.
- **Jump Duration:** Time taken to execute jump
- **Jump Timeout:** Time taken for jump to timeout and lose power.
- **Jump Count:** How many jumps can be executed before hitting the ground.
- **Continuous Jump:** Holding down jump will keep the jump going as a single action instead of multiple distinct actions.
- **Event:**
  - **None:** will not wait for an event.
  - **Button:** will execute when jump button pressed.
- **Syncronize Animation:** if ticked, animation will sync with main character animations.
- **Collision Shape:** custom collision shape used while jumping.
- **Animation:** animation used while jumping.
- **Ground:** Graphic underlay for when character on the ground.
- **Jump Sound:** sound played when jump starts.
- **Double Jump Sound:** sound played when double jump starts.

Character Icon
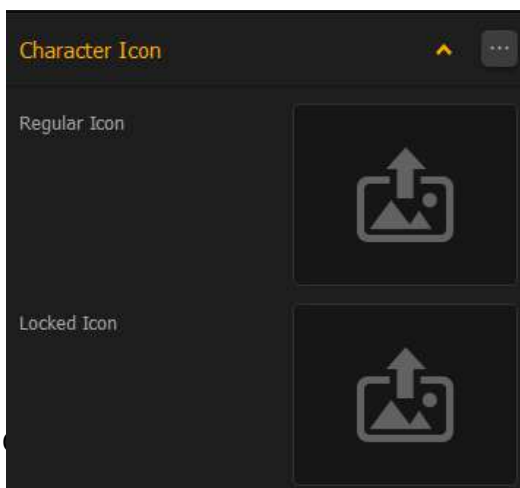
- **Regular Icon:** Normal icon seen in character selector.
- **Locked Icon:** Locked icon seen in character selector.

21
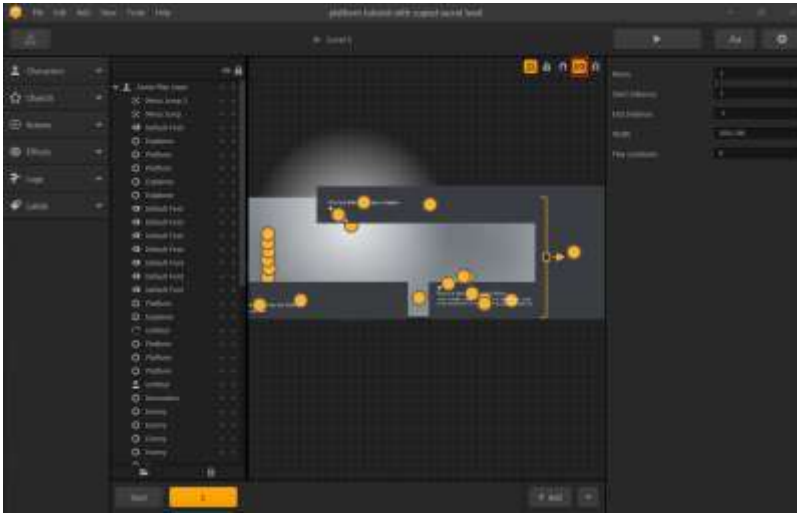
Connections

To setup and edit connections, press the button highlighted, above. You will see a yellow dot at the centre of each item in the scene.

To create a connection, drag and drop from the source to the destination entity's yellow dot. The source entity will be in control of the other entity. Ie where the source moves the destination will move too.

Note: if you connect an object to the Character, and have platform objects that are set to destroy the Character, be sure that you set the parent object's Health value to 1 higher than any object's Damage value (even if that Damage value is 0). If you do not do this it will die on contact.

It should be noted that connections may move about their axis, if the object they are attached to is able to move. You can see this by placing three objects on screen:

- First object fixed

- Second object physical

- Third object physical

The second object will turn in place, while the third object will swing around if pushed (using the first object as a hinge).

For more advanced usage, see Motors, under the Advanced section.

Objects

Once an object has been created (see Drag-Drop to Create New Items) you can now edit them in the asset bar on the left, or edit any instances that are sitting in the current scene (referred to as sub-objects).

We will deal with the former, first. To edit the object, click on it in the asset bar, and you will see the options on the right, as you can see in the screenshot. The options are:

- **Name:** used for internal reference only. It will appear in the Scene Tree

- **Collision Shape:** see this reference.

- **Default Animation:** Shown most of the time object is on screen. See Animation Editor.

- **Defeated Animation:** shown when object dies. See Animation Editor.

- **Idle Sound:** sound that is played when object is awake and on-screen.

- **Death Sound:** sound that is played when object is dying.

- **Wakeup Sound:** sound that is played when the object wakes up.

Object Custom Components

At the bottom of the properties section, is a button – [+ Add Component]. There are several options. Characters also have Components, but with a different set of options. Note: there are a different set of custom components you can add to an Object Instance – see next heading.

Advanced Move

Options avilable are:

- **Name:** for your reference only.
- **Collision Shape:** The collision shape that will be used to trigger this.
- **Animation:** Animation to play when this is triggered.
- **Linear Velocity(x,y):** Velocity vector to add to the Character.
- **Angular Velocity:** Angular velocity to add to the Character.
- **Event:** - triggering event for this move
  - **None:** not triggered.
  - **On Start:** As soon as Character appears.
  - **Collision:** When collision shape is hit by another object.
- **Timeout:** how many seconds after triggering does this move stop.

Damage

- **Damage:** Amount of damage inflicted on another object.
- **Damage Delay:** How long till the above damage is inflicted.

Health

- **Health:** The amount of health this character has to lose before dying.
- **Lives:** How many times this character has to lose before dying completely.

Rotation Mapper

- **Animation:** The animation to show while rotating.

Sub-Objects / Object Instances
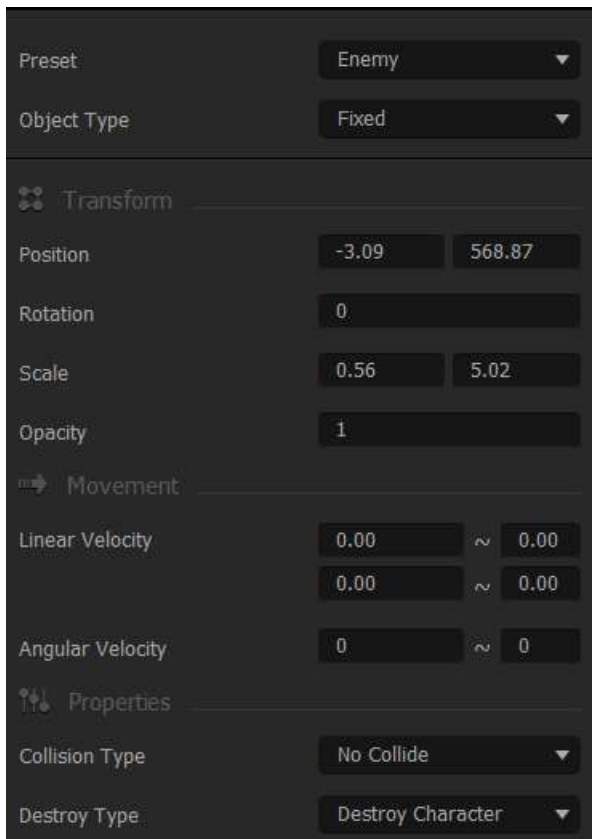
So now you have defined your objects, you may be wondering where most of the settings are, and how to have different versions of the same object. Sub-Objects are where we do this. You create a sub-object by clicking and dragging an object from the object bar on the left, onto the scene.

Just put the object where ever you want it in the scene. Click on your object to select it, and look to the options bar on the right. Now there are various settings that really only apply to different object types. A very fast way to set most of the properties you would want for particularly object types is the very first field:

- **Preset:** - List of standard object types. These are a quick base, only. Once you select your preset, you can change all the settings – nothing restricts you.

  o **Platform:** this is a non moving object that is often used for a platform for the characters to walk on.

  o **Physics Object:** this is for an object that you want to obey the laws of game physics. It will move around, bounce, etc and respond to forces acting on it.

  o **Enemy:** this is an object that will kill the character on contact, and can be killed by attacking it.

  o **Enemy Bullet:** similar to the above, this will spawn copies of itself at intervals.

  o **Character Bullet:** this will spawn copies of itself at intervals. All copies produced will kill the character on contact.

  o **Decoration:** this is used for scenery. It affects nothing, is affected by nothing. Its job is to look good.

  o **Wheel:** as it sounds. Used for wheels including wheels attached to the character.

- **Object Type:** - how the object moves

  o **Fixed:** Object will not be effected by forces of gravity or other objects working upon it.

  o **Physics:** Object will be effected by gravity and other external forces.

- **Transform:**

  o **Position(x,y):** object position in scene.

  o **Rotation: ro**tation in degrees clockwise. 0 is not rotated at all.

- o **Scale(x,y): h**ow much bigger or smaller in either direction the sub-object is compared to the base object. 1 is normal size.

  o **Opacity:** how transparent the object is.

- **Movement:**

  o **Linear Velocity(x,y):** object starting speed/direction. 0 is not moving at all.

  o **Random factor:** an amount between 0 and this value will be added to the basic value.

  o **Angular Velocity & Random factor:** value in degrees per second that the object will rotate about its pivot point. Positive values are clockwise.   Random factor is an amount between 0 and this value will be added to the basic value.

- **Properties:**

  o **Collision Type:**

    ▪ **No Collide:** object will never collide with anything.

    ▪ **Collide:** object will collide with other objects

- **Destroy Type:** This has been split from Collision Type for greater flexibility. Determines if the Object's *Damage* value will be subtracted from an objects *Health* value on contact (both those values are set on the parent Object)

  o **No Destroy:** Nothing will contact when this object's collision box touches any others.

  o **Destroy Character:** Will damage the player Character on contact, only. Destruction of object collided with can apparently avoided by giving it a high health value.

  o **Destroy Enemy:** Will damage Enemy objects on contact, only. Destruction of object collided with can apparently avoided by giving it a high health value.

  o **Destroy All:** Will damage any enemy or player Character. Destruction of object collided with can apparently avoided by giving it a high health value.

Sub-object Custom Components

To access these, press the [+ Add Component] button at the bottom of screen.
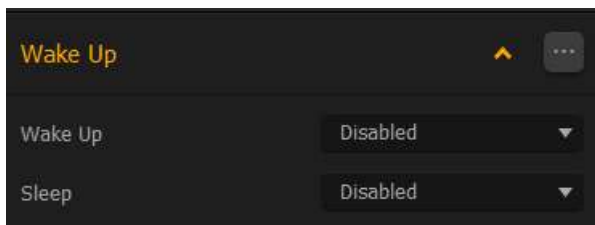
- **Wake Up:**
  o **Disabled:** object will start awake.
  o **Distance Based:** object will wake up when the character is the specified pixels away from the centre point.

- o **Collision Based:** object will wake up when the character's collision box collides with it.
  - o **Wakeup Distance:** distance in pixels (shows if Distance Based)

- **Sleep:**

  - o **Disabled:** object will never sleep

  - o **Distance Based:** object will sleep after moving the character is the specified pixels.

  - o **Sleep Distance:** distance in pixels (shows if Distance Based).
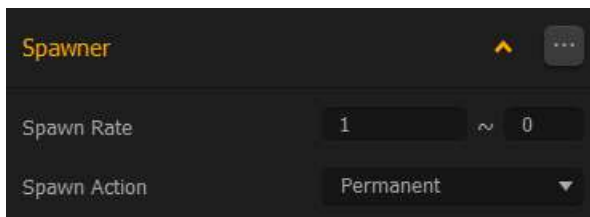
- **Spawn Rate:** time in seconds between object spawning.

- **Spawn Action:**

  - o **Permanent:** always spawn.

  - o **Shooting:** spawn when shoot button pressed.

  - o **Single Shooting:** as above but approximately only a single "bullet" on screen at any one time.

  - o **Jumping:** spawn when jump button pressed.

Actions

Actions are mostly configured from the parent in the object bar to the left. The basic settings are:

Name	Coins
Collision Shape	Edit
Default Animation	
Action Type	None ▼

☑ Animation

Action Animation	
Animation Placement	Character ▼
Animation Behaviour	Overlay ▼
Animation Repeat	Looped ▼

◀) Sound

Idle Sound	◀)
Start Sound	◀)
End Sound	◀)

⊶ Properties

Reward Coins	1.00
Show coin reward	✔
Reward Points	0.00
Show points reward	
Duration	1.00
Appearance Chance	100.00

⊶ Activate Effects

Camera Shake	0.00
Camera Flash	0.00

- **Name:** for your reference only.

- **Collision Shape:** see this.

- **Action Type:**

  o **None:** do nothing.

  o **Coin:** do nothing but add **Reward** number of coins to the player's coin collection.

  o **Kill All Enemies:** Kill all objects of type Enemy, that have Destroy set to "Destroy Character" and are active (ie objects are not asleep).

  o **Invincibility:** make player invincible for a time, and will kill enemies on contact.

  o **Powerup Magnet:** suck Powerup Actions (such as coins and abilities, but excluding Checkpoints) toward the character while active.

  o **Strike:** when collected, a button when pressed will kill enemies in contact with the player.

  o **Set Checkpoint:** set a checkpoint, that will be used for Restart From Checkpoint button.

  o **Next Checkpoint:** move player character to next checkpoint.

  o **Restart Checkpoint:** give ability to restart at last checkpoint.

  o **Gameplay:** Allow any gameplay property to be changed. Will allow the *Gameplay properties* options to appear. Any game play property found in *Project Settings* can be changed.

Animation:

- **Action Animation:** PNG Drop Box for the animation to play when triggered.

- **Animation Placement:**

  o **Character:** animation plays on character.

  o **Action:** animation plays on Action location.

  o **Full Screen:** animation plays full-screen.

- **Animation Behaviour:**

  o **Overlay:** run animation OVER the location.

  o **Replace:** replace really only makes sense with placement of Action and Character. You could replace character animation to change what the playing character looks like while the Action is working. (For some examples, in an fantasy game you could have an animation of your character doing something with their hands for a Kill All Enemies action, or a cloud of smoke for Next Checkpoint)

- **Animation Repeat:**

  o **Looped:** play animation continuously a long as Action is working.

  o **Single Play:** run animation once and stop.

Sound:

- **Idle Sound:** sound to play when Action is on screen and not yet collected.

- **Start Sound:** sound to play when Action is initially triggered.

- **End Sound:** sound to play when the Action finishes.

Properties:

- **Reward Coins:** how many coins will it take to purchase this Action. Leave to 0 if not purchasable.

- **Show Coin Reward:** display the coin value when player picks up the action.

- **Reward Points:** how many points will be rewarded for collection.

- **Show Points Reward:** display the point value when player picks up the action.

- **Duration:** how many seconds Action will remain in effect.

- **Appearance Chance:** a value of 100 will always show this Action, and anything less than this will show it sometimes. 50 would mean show half the time the game is played.

Properties:

- **Camera Shake:** shake the camera when Action is triggered.

- **Camera Flash:** run a camera flash effect on the whole screen when Action is triggered.

Gameplay Properties

Shown only when Action Type is set to Gameplay. These are a mix of world and character settings that will change when this action is tripped.
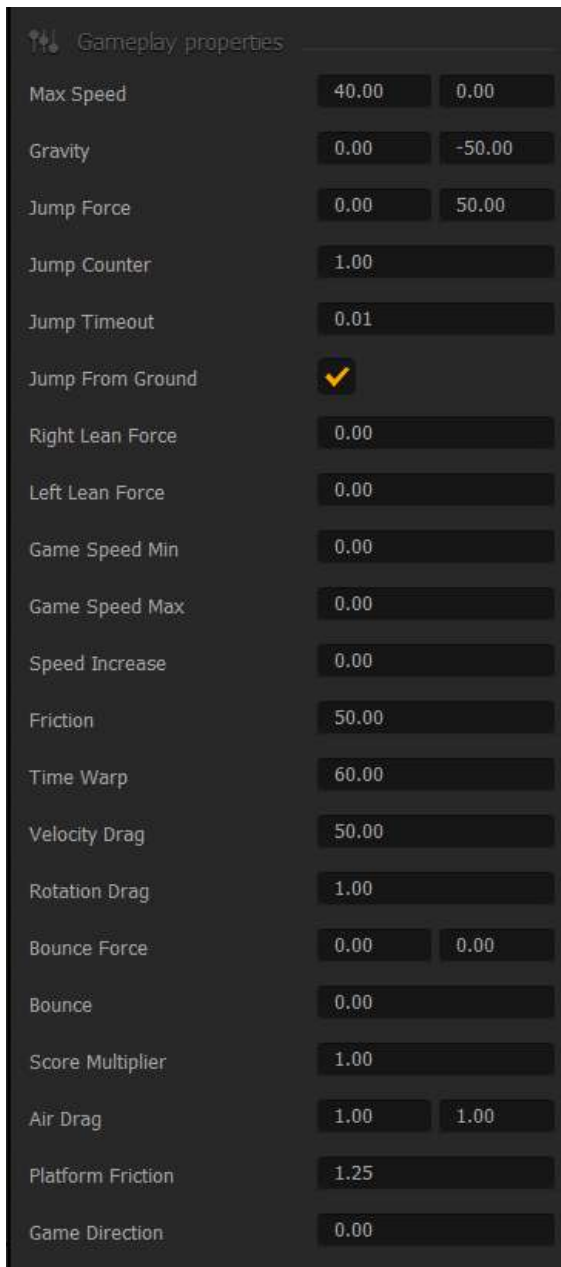
- **Max Speed (x and y):** max speed the character can move in.

- **Gravity (x and y):** continuous force acting on everything as gravity does in the real world.

- **Jump Force (x and y):** force on character when you press the jump button.

- **Jump Counter:** how many jumps can be made before touching the ground.

- **Jump Timeout:** is used for variable jump. A value of 0.01 gives consistent jump, higher values give option to have a variable height jump depending on how long button is held down.

- **Jump from Ground:** can the player jump off the game-field boundary.

- **Right Lean Force:** force that will lean/rotate character right when lean-right button pressed.

- **Left Lean Force:** force that will lean/rotate character left when lean-left button pressed.

- **Game Speed Min:** start speed at which game will move past the player (regardless of player movement).

- **Game Speed Max:** final maximum speed at which game will move past the player (regardless of player movement).

- **Speed Increase:** amount to add to speed increase.

- **Friction:** friction between all objects.

- **Time Warp:** how long to wait before adding 'speed increase' to the current game speed.

- **Velocity Drag:** force used to slow anything that is moving down. Directly opposes all movement unlike air-drag for character, though very similar.

- **Rotation Drag:** force that will slow down any rotation of the character.

- **Bounce Force (x and y):** force the Character will exert on another object. Note: values here will constrain the movement of the player in an unrealistic way - eg with a vertical bounce force, hitting an angled surface will bounce you straight up and not sides - usually this is what you require. That is why there is an X and Y value to specify. If you require more standard physics for your bounce, leave this at 0,0 and put a value for World Bounce force. Also this will not affect objects attached to the character.

- **Bounce:** bounce force between all objects

- **Score Multiplier:** Scoring is 1 point per pixel moved. A value here will increase that. Eg value of 10 will mean 10 points per pixel.

- **Air Drag (x and y):** force that will slow down the character's movement over time.

- **Platform Friction:** friction force that will stop character sliding against platforms. Note: this will not affect objects attached to the character.

- **Game Direction:** In clock-wise degrees where 0 has the character moving left. -90 will be moving up, 90 will be moving down, and 180 will mean moving left.

Action Placement

Placement of Action instances (a sub-Action) is the same as any other asset type.   Simply drag and drop the Action onto the scene.

As for Objects, you can select it, and set the following properties:

- **Position(x,y):** position on the scene.

- **Rotation:** in degrees.

- **Scale(x,y):** how big the Action is. 1 is full size.

- **Opacity:** how transparent the object is – from 1 to 0 where 1 is completely opaque.

Effects

All the Effects work in a similar way. They are not Objects, so cannot be given physical properties. Some you can attach to objects however so you can in some cases make appear to have physics properties - by making the attached object bounce and move.

Trail
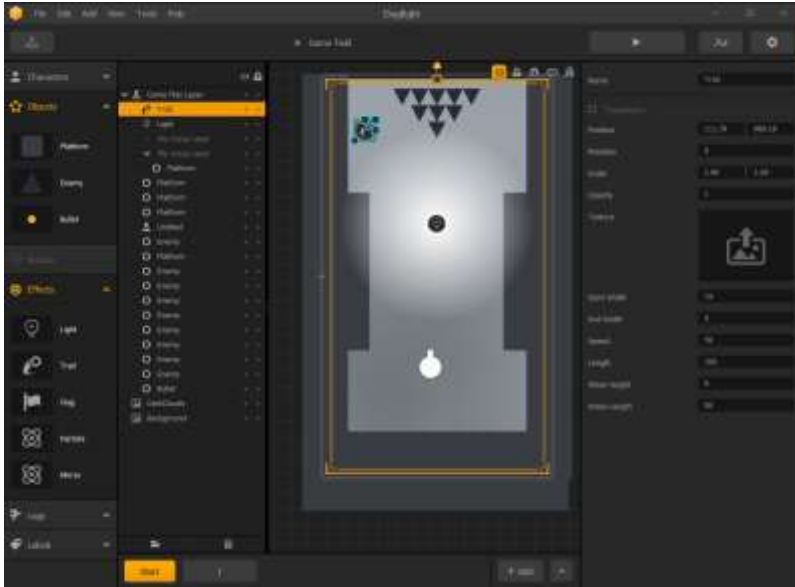
Trails can do all sorts of things, largely limited by your imagination. They can be used to show creepy tentacles, waving grass, smoke, etc. Options for a trail are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** what direction it is pointed

- **Scale (x,y):** after the other options are calculated, the scale values are used to resize/distort the trail.

- **Texture:** a PNG drop box that lets you paint the trail with a particular texture.

- **Start Width:** how wide the tail will start with.

- **End Width:** how wide the tail will be at the end.

- **Speed:** how fast the tail will move.

- **Length:** how long in pixels the tail should be.

- **Wave Height:** how high each wave movement will be - the distance from the very bottom to the very top of the wave.

- **Wave Length:** how long each wave movement will be - the distance from peak to peak.

Light
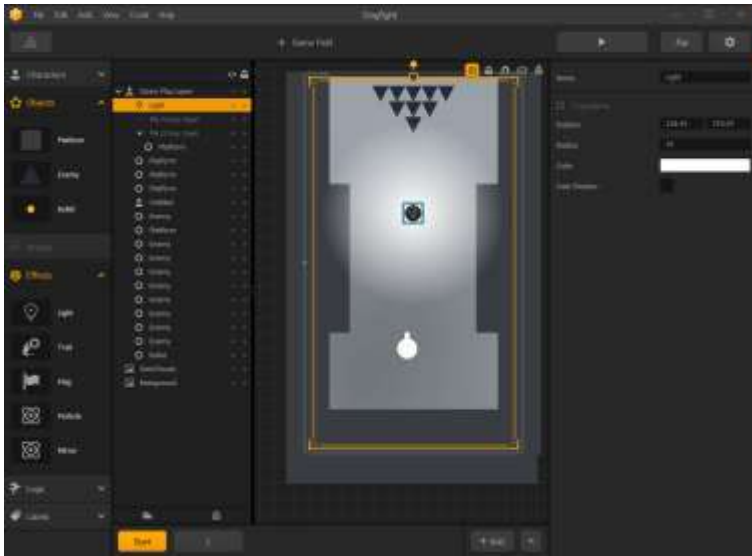
Lights are a nice simple way of providing atmosphere to a game. Be cautious about over using them however a too many will start to use up available CPU time. Options for a light are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** at first glance this may have no use, however if you scale the light into an oval then rotation is useful for pointing that object in a particular direction.

- **Scale (x,y):** You would usually leave this at a value of 1,1 - because setting Radius is a better way of making the light bigger or smaller - however mixed value of x and y will let you distort the light.

- **Radius:** how big the light is.

- **Color:** what color the light is. This can have some non-obvious effects, like making the light object into a spreading darkness by making it a dark or black color. It is a full color-picker so you can set anything including transparency.

- **Cast Shadow:** a great way to add atmosphere. The various objects' collision boxes will be used for the shadows cast.

Notes:

- Shadows will not appear in the editor. You need to run the Preview mode to see them.

- You can achieve a similar effect to light by careful use of an imported image Object. See the Character in the example game, Spark.

- Lights and shadows are a little computationally expensive, so can be revisited when you need to optimise a game.

Flag
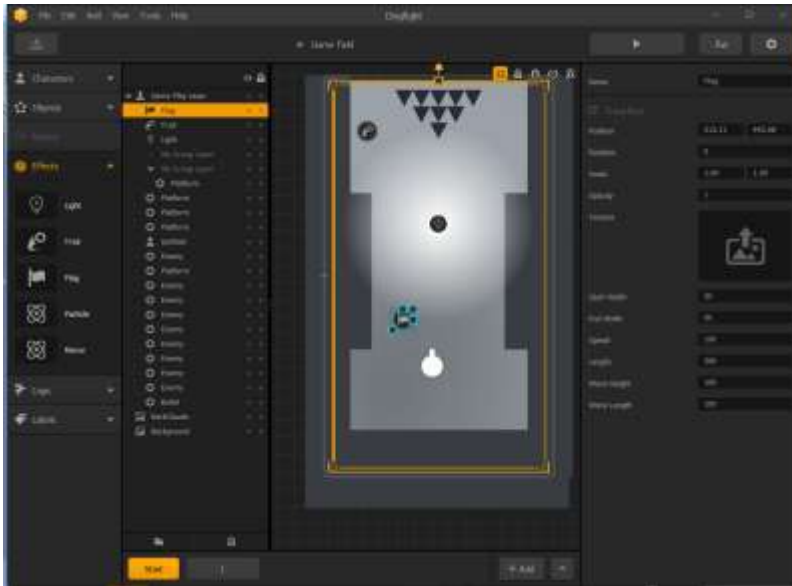
A flag is just what it sounds - an image that appears to be rippling in a breeze. Now you are not limited to just flags - you may use this effect to animate jellyfish or similar. Just remember that if you want this effect on an enemy, that there needs to be an actual Object attached that will kill the player character. Available options are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated.

- **Scale (x,y):** You would usually leave this at a value of 1,1 - because the other settings will make thing bigger already. You can use these to distort the result.

- **Texture:** a PNG drop box that lets you paint the trail with a particular texture.

- **Start Width:** how wide the tail will start with.

- **End Width:** how wide the tail will be at the end.

- **Speed:** how fast the tail will move.

- **Length:** how long in pixels the tail should be.

- **Wave Height:** how high each wave movement will be - the distance from the very bottom to the very top of the wave.

- **Wave Length:** how long each wave movement will be - the distance from peak to peak.

Particle
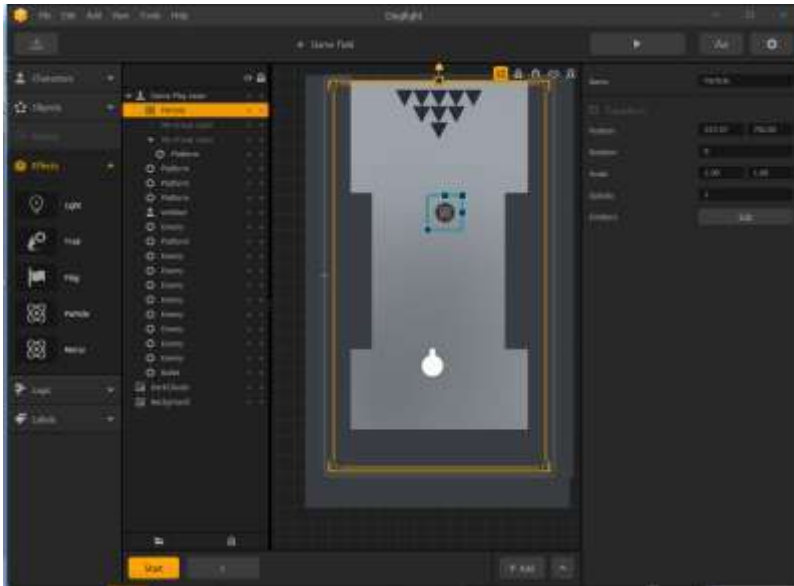
There are the following basic settings:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated.

- **Scale (x,y):** You would usually leave this at a value of 1,1 - because the other settings will make thing bigger already. You can use these to distort the result.

- **Opacity:** how transparent the particles will be.

- **Emitters:** This brings up the Particles Editor, where the magic happens. See the following section for details.

Particles Editor

With this screen you have the full power of any of the market leading particles editors – but built into the Buildbox™ program. If you find a particle effect you like in another program, just note down the settings and fill them out.

It may not be obvious at first, but you can create a particle effect using multiple emitters. The example screenshot shown is using two. Just remember that using a lot of particle effect can use a lot of CPU capacity.

On the left of screen you will see a list of emitters. Similar to the Scene Editor, you can lock or hide an emitter. You can delete them by clicking on the trash icon down the bottom, or add a new emitter by clicking on the [+] icon.

Also like the Scene Editor, at the top of the centre section you will see the debug icon, which turns on or off the emitter symbols. It can be handy to turn them off to better see the animation.

Drag your emitters around till you are happy with their placement, in the knowledge that in-game they will be placed according the centre of the screen marked with the cross.
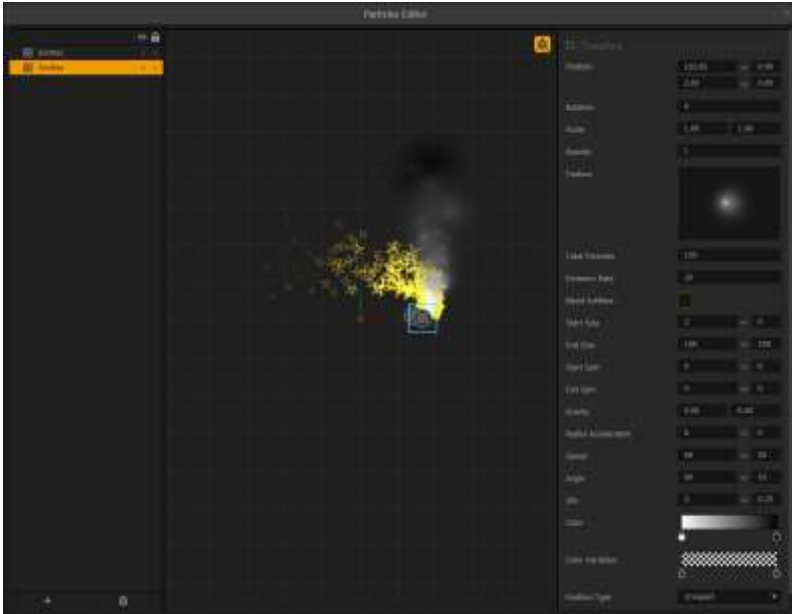
The following options are available for each emitter:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated. Not normally used.

- **Scale (x,y):** Does not have an actual effect.

- **Opacity:** how transparent the emitter is.

- **Texture:** the image used for every particle this emitter spawns.

- **Total Particles:** How many particles to have on the on screen at any one time. More particles will not be spawned until existing particles disappear. Increase this to increase your particle volume ie more smoke, fire, etc.

- **Emission Rate:** How fast to emit particles.

- **Blend Additive:** overlapping particles add to each other making the overlapped areas brighter. Useful for flames.

- **Start Size (and random variation):** the size of the particle when it is first spawned.

- **End Size (and random variation):** the size of the particle when it is destroyed.

- **Start Spin (and random variation):** the spin of the particle when it is first spawned.

- **End Spin(and random variation):** the spin of the particle when it is destroyed.

- **Gravity(x,y):** gravity or wind force on the particles.

- **Radial Acceleration(and random variation):** How much acceleration away from the point of emission is given to a particle. An explosion will have a high number, here, while smoke or water will have 0.

- **Speed(and random variation):** initial speed of particles when spawned.

- **Angle(and random variation):** initial direction of travel of particles when spawned.

- **Life(and random variation):** how long will the particles live before they are removed.

- **Color:** Color tint of the particles.

- **Color Variation:** How that color tint will be varied.

- **Position Type:** - how will the particles will act once emitted into the world. The effect of this setting can only be seen when you have a moving particle emitter.

  - **Free:** Particles are attached to the world and unaffected by the emitter movement.

  - **Relative:** Particles are attached to the world but will follow the emitter as it moves. (this mimics real-world behaviour)

  - **Grouped:** Particles are attached to the emitter and move directly with it.
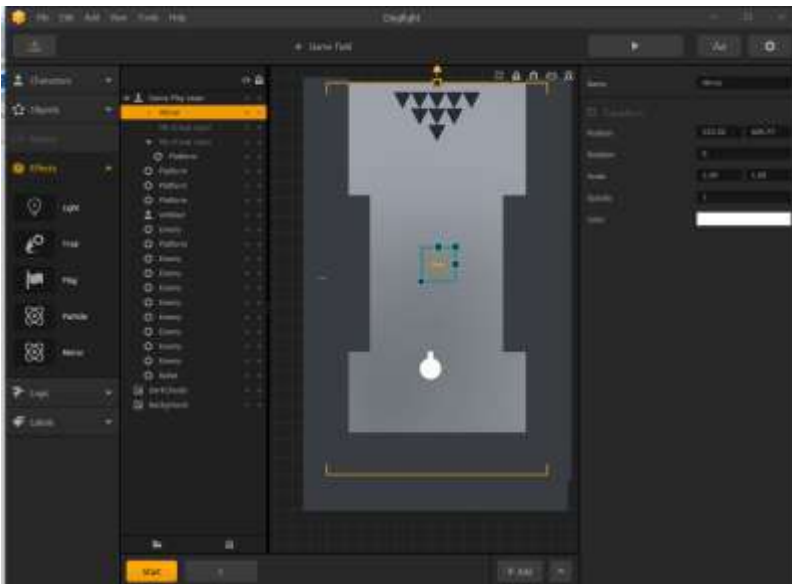
Mirror

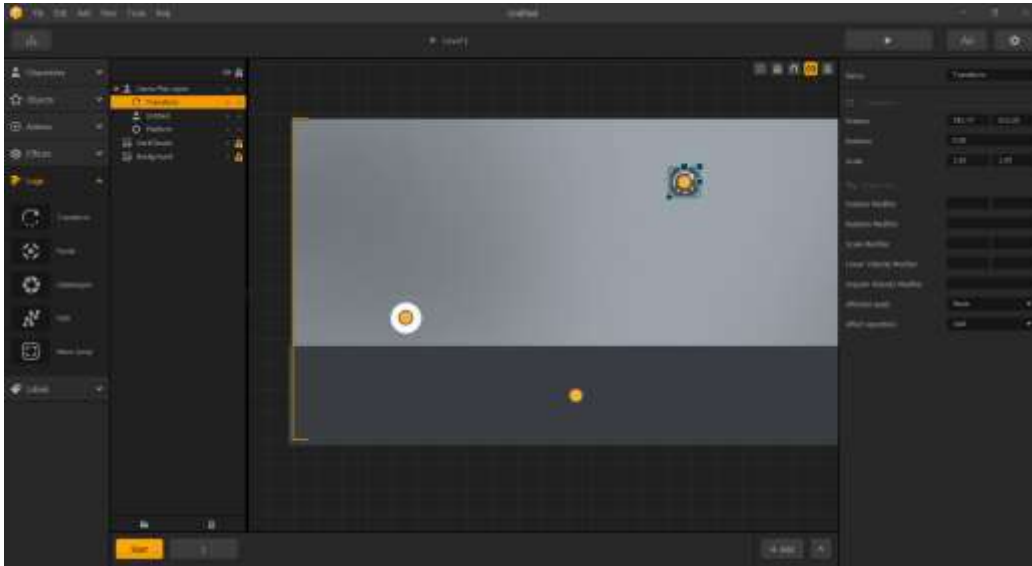This is a reflecting water effect. The following options are available for each mirror:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated. Determines what side of the mirror is nearest the screen.

- **Scale (x,y):** Does not have an actual effect.

- **Opacity:** how transparent the emitter is.

- **Color:** Color tint for the mirror.

Logic

This is where objects that dynamically control game objects appear. It appears under Effects and above Labels on the main bar to the left. There is currently a single Logic item available. They are not the only thing that can be used to alter game play during game play. Actions can also be used to alter game play settings.

To use a Transform item, simply drag onto the active scene. Once this is done a single instance will be created, and settings can be configured on the right (when selected) as shown above. There is no effective limit to the number of Logic items that can be used on a scene.

Transform

Objects colliding with this will have their setting changed. The available settings are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated. Not normally used.

- **Scale (x,y):** Does not have an actual effect.

- **Position Modifier (x,y):** move the affected object in the x and y directions.

- **Rotation Modifier:** angle in degrees to rotate the object.

- **Scale Modifier (x,y):** change the size of the object. Negative values can be used to invert the object – useful for an object that needs to turn around the go back in the opposite direction so the object does not end up moving backwards.

- **Linear Velocity Modifier (x,y):** change to the speed the object is moving

- **Angular Velocity Modifier (x,y):** change to the speed the object is rotating.

- **Affected asset:** a dropdown of all Objects in the game. One modifier is needed per Object you wish to affect. This is also where a good naming convention will make your job as a game builder much easier! If you want to Affect the player Character or World, you need to look at Action, Gameplay type (see Gameplay Properties).

- **Affect operation:** how to apply the settings listed here, to the Object values in-game.

- o **Add:** Add values to the existing values for the affected object.

- o **Multiply:** multiply existing values for the affected object. Useful when turning an object around by applying -1 values for example.

- o **Replace:** change all values you have set. If you set all values then all values will change. If you set only some values – leaving the rest blank – then only the changed values will alter.

Portal
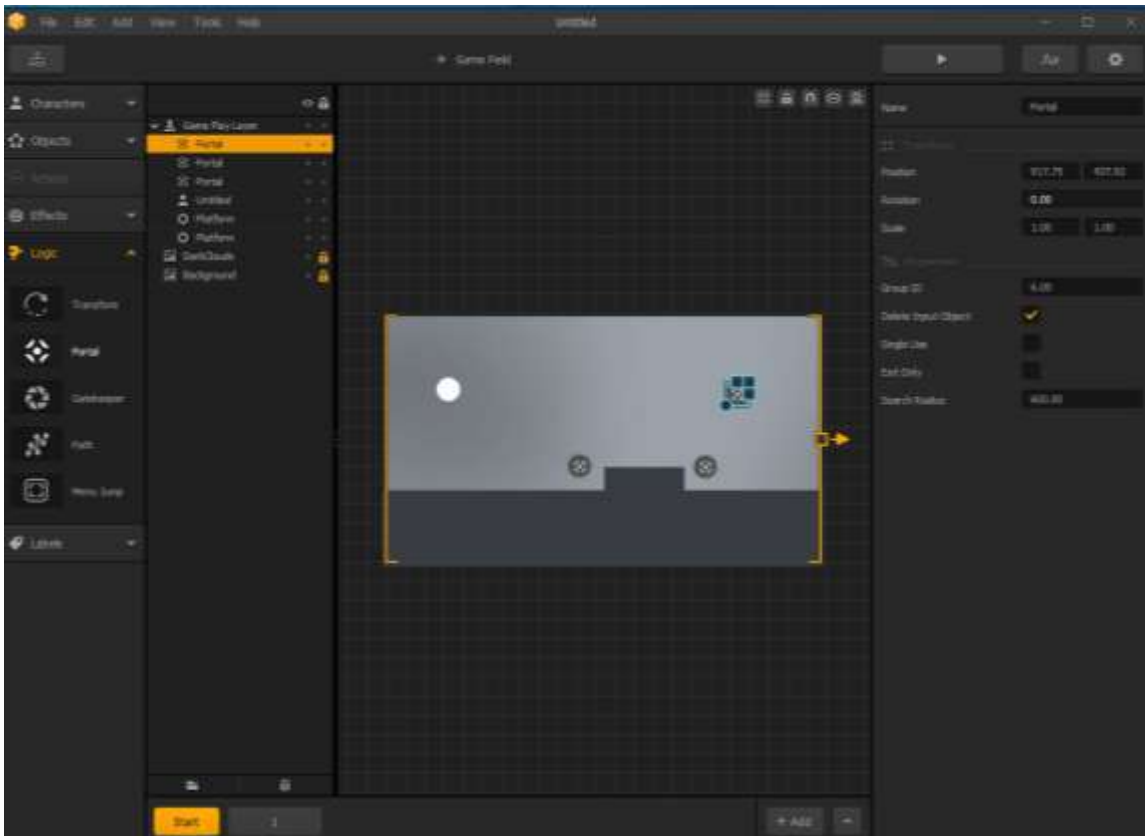
Portals are used for jumping the Character from one place in a World, to another – and optionally duplicating the Character (something Agent 86 would be familiar with). The available settings are:

- • **Position (x,y):** as for any other placed item.

- • **Rotation:** angle in degrees that this should be rotated. Not normally used.

- • **Scale (x,y):** does not have an actual effect.

- • **Group ID: t**his portal will transport to(/duplicate) or from other portals with this group ID. If there are more than two portals with the same id, the Character will be duplicated.

- • **Delete Input Object: i**f this is not ticked, the portal will be a duplicator. The original will still be at the entering Portal.

- • **Single Use: c**an only be activated once in the game.

- • **Exit Only:** portal will not activate when moved into. It will still act as an exit, however.

- • **Search Radius:** How far away the character can be and still trigger the portal open.

GateKeeper
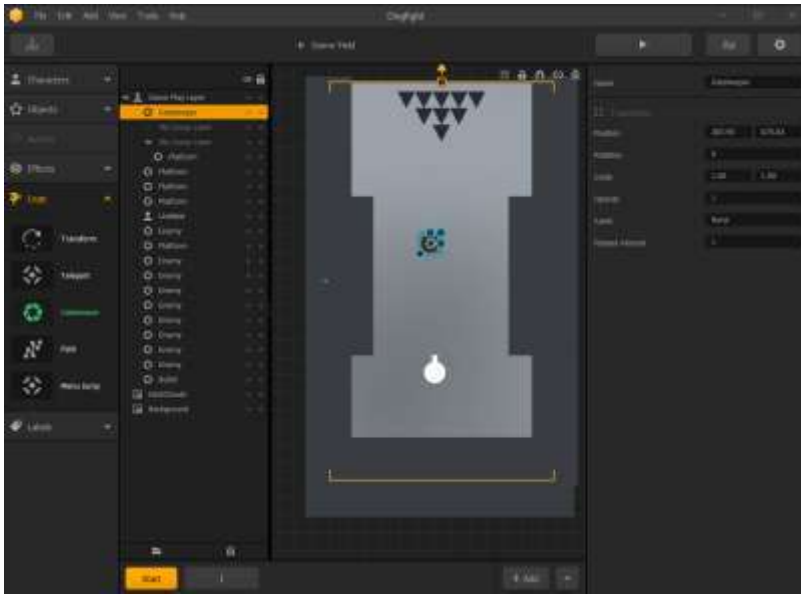
Gatekeepers are used to prevent a player Character from passing more than a number of times through an area.   Used to create, for an example, a one way door (though they are more flexible than that). The available settings are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated. Not normally used.

- **Scale (x,y):** Does not have an actual effect.

- **Passed Amount:** Number of times the Character will be allowed to pass through the GateKeeper. Every time the Character passes through it, this number will reduce by one until it reaches 0. Once it is zero, it will be impassable.
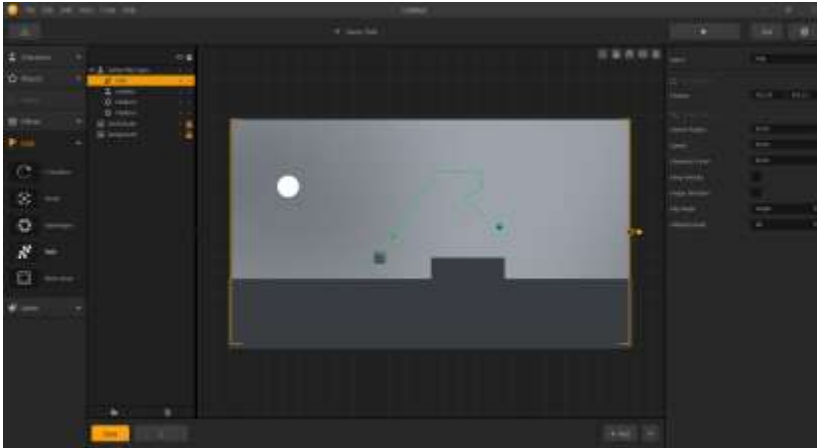
Path

A Path will let you take the Character on a ride. Control by the player ceases, while the Character moves along the line you have defined. If there are physical Objects, or Logics, etc, they will interact (or interrupt) the Character while on the Path.

After dragging onto the design screen, you will see the Path represented as a green line. Note the following elements:

 Click this to select the Path

 Click the larger version of this to add a segment to the end of the line. Click any of the smaller versions in the middle of a segment split a segment in two (if they are not currently visible, the split action will still occur).

 Click and drag a corner like this to move it.

The available settings are:

- **Name:** for internal reference, only.

- **Position (x,y):** as for any other placed item.

- **Search Radius:** How close the Character needs to be to activate this path.

- **Speed:** how fast the Character should move. If this was set to a very high value, it would look a lot like the actions of a Portal (provided there was nothing in the way to interact or stop the Character of course).

- **Attraction Force: h**ow much force each point in the path will attract the Character, in turn. This is a setting that is a lot of fun to play with. With very low values you can get effects similar to Angry Birds™ Space around each point – that can be resisted by the player controls and other forces including the World's gravity. Allowing the player to fight the Path, means they can eventually move out of its influence, when they get far enough away.

- **Keep Velocity:** when Character finished path, it should have its original velocity.

- **Image Direction:** tick this to keep the player pointing the 'right' direction on the Path.

- **Play Mode:** how the Character will move along the Path

    o **Single:** from beginning to end, and exit.

    o **Loop:** from beginning to end, then jump back to the start.

    o **Ping Pong:** from beginning to end, then end to beginning.

- **Affected Asset:**

    o **All:** for all Objects and Characters.

    o **Characters:** for Characters only.

    o **Assets:** for Objects only.

Menu Jump
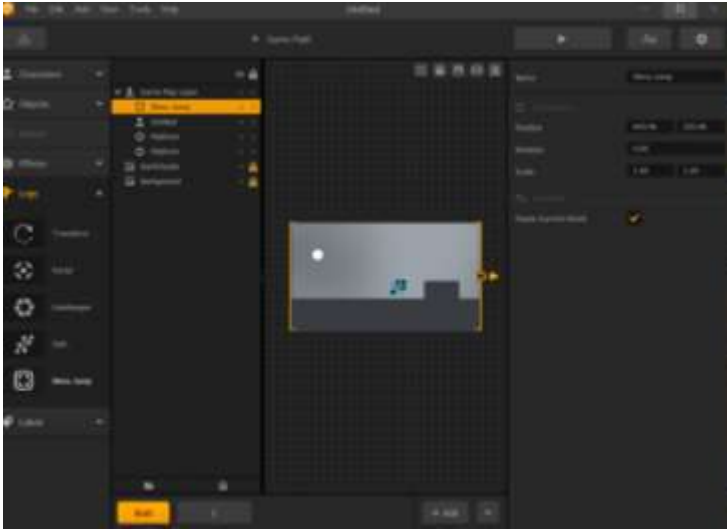
Place one of these in a Scene, and a new OUT selector will appear in the Mind Map. You can use that to take the player to any other world.

The available settings are:

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated. Not normally used.

- **Scale (x,y):** How big the jump point will be.

- **Pause Current World:** Keep current world in a frozen state, so that when the Character returns to this world, they can continue from the current location and game state.

Timeline Editor

Default scenes view
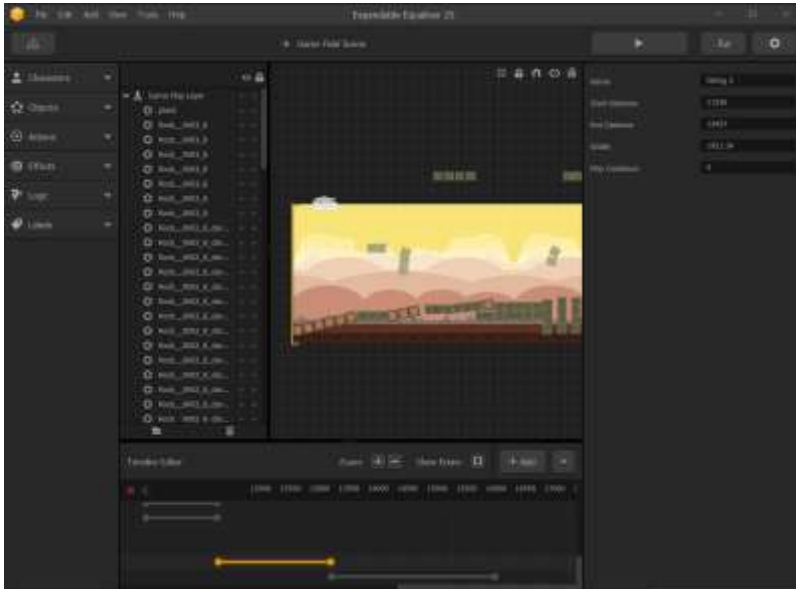
Opened scenes view

To open the Timeline Editor, press the down arrow highlighted, above. This will easily show you all the scenes and how they are arranged. To go back to the simplified view, press the same button again.

The **Start** scene is a special scene that will always be first. All other levels can be placed anywhere in the time line. Where two or more scenes overlap - as Scene 2 and 3 do in the example, Buildbox™ will randomly select from each of them at run-time. Each level will have an even chance of appearing.

Where only a single scene appears - no overlaps - then scenes will just appear in order.

If you look at the currently elected scene, above, you will see that the start and end position on the time line diagram, also appear on right. All numbers are in pixels.

For easier testing, you can deactivate or "solo" scenes. "Solo" means show the same scene repeatedly. Colors will change to indicate what will appear in the Preview.

To solo a scene, press the blue "S" next to the scene on the time line, or select the scene and press the "s" key on the keyboard.

To disable a scene, press the red "M" next to the scene on the time line, or select the scene and press the "m" key on the keyboard.

A scene in green will be soloed, and scenes in red ware disabled for the Preview.

Keyboard Shortcuts

Editor Controls

Game Mind Map

Right click	show add-new menu when the right-click is on the background only. The menu is rather self explanatory:- • Add New Scene Screen • Add New UI Screen
Mouse scroll	zoom in and out.
D	duplicate-right, currently selected object, character, action or effect.
S	duplicate-down, currently selected object, character, action or effect.
A	duplicate-left, currently selected object, character, action or effect.
W	duplicate-up, currently selected object, character, action or effect.
Drag and drop	move currently selected object, character, action or effect.
[ctrl]- + (PC) ⌘- + (Mac)	Zoom in
[ctrl]- - (PC) ⌘- - (Mac)	Zoom Out

Scene Editor

[space]	activate hand tool. When hand appears, you can use click and drag to move the level around.
[shift]	constrain aspect ratio when scaling anything.
[command]-Right click (Mac) [ctrl]-right click (PC)	show list of objects under the mouse (Mac only). Click an object in the menu to select it.

Right click	show layer menu with these options, which will be performed on the selected object, character, Action or effect:- • Bring to Front: move to the very front. • Bring Forward: move forward one layer. • Send Backward: send backward one layer. • Send to Back: send to the very back.
⌘-} (Mac) [ctrl]-} (PC)	Bring to Front: move to the very front.
⌘-] (Mac) [ctrl]-] (PC)	Bring Forward: move forward one layer.
⌘-[(Mac) [ctrl]-[(PC)	Send Backward: send backward one layer.
⌘-{ (Mac) [ctrl]-{ (PC)	Send to Back: send to the very back.
Mouse scroll	zoom in and out.
⌘-C (Mac) [ctrl]-C (PC)	copy currently selected object, character, action or effect.
[command]-V (Mac) [ctrl]-V (PC)	paste currently selected object, character, action or effect.
[command]-X (Mac) [ctrl]-X (PC)	cut currently selected object, character, action or effect.
D	duplicate-right, currently selected object, character, action or effect.
S	duplicate-down, currently selected object, character, action or effect.
A	duplicate-left, currently selected object, character, action or effect.
W	duplicate-up, currently selected object, character, action or effect.
⌘-[shift]-4 (Mac)	tool for calculating number of pixels in a specific area and make a screenshot of an area.
Drag and drop	move currently selected object, character, action or effect.

[shift]-Drag and drop	move currently selected object, character, action or effect horizontally or vertically only.
←↑→↓	move currently selected object, character, action or effect.
[shift]- ←↑→↓	move currently selected object, character, action or effect by 5 pixels
[alt] (PC) [option] (Mac)	Show connections
[ctrl]- + (PC) ⌘- + (Mac)	Zoom in
[ctrl]- - (PC) ⌘- - (Mac)	Zoom Out
[space] – Drag and drop	Scroll the scene around

With an object selected, change the Object Type

[ctrl]-1 (PC) ⌘-1 (Mac)	Platform
[ctrl]-2 (PC) ⌘-2 (Mac)	Physics Object
[ctrl]-3 (PC) ⌘-3 (Mac)	Enemy
[ctrl]-4 (PC) ⌘-4 (Mac)	Enemy Bullet
[ctrl]-5 (PC) ⌘-5 (Mac)	Character Bullet
[ctrl]-6 (PC) ⌘-6 (Mac)	Decoration
[ctrl]-7 (PC) ⌘-7 (Mac)	Wheel

Scene management

D	duplicate currently selected scene.
[Delete]	delete currently selected scene.
M	mute scene for testing in play mode.
S	Solo scene for testing in play mode.

Menu Screens Editor

M	Mute or enable the currently selected screen

Game Playing Controls

These controls will work in the simulator – assuming the controls have not been customised - and on Mac and PC.

←	move character left.
→	move character right.
↑	move character up.
↓	move character down.
J	jump
<	(strictly speaking, it is ",") rotate character left
>	(strictly speaking, it is ".") rotate character right.
z	rotate motors backward.
x	rotate motors forward.
[space]	Shoot

Menus/Screens

Game Mind Map
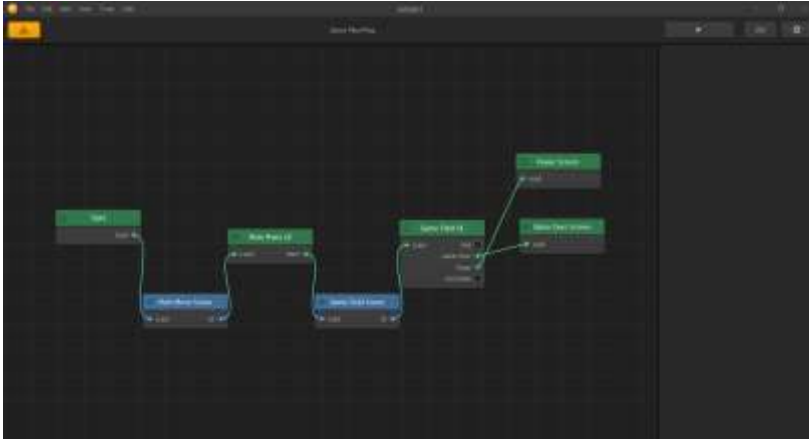
This screen shows an overview of the Buildbox™ menu structure for your game, including what screens allow you to navigate to what other screens. There are two types of Screens:

- **Scene Screen (blue):** In versions of Buildbox™ prior to version two, there was only one of these. This is a game-play screen. It houses the levels only - there are NO UI controls set here. To setup game play UI you need to create a UI Screen, and link it to the UI item in the Mind Map.

- **UI Screen (green):** Ads and UI elements.

- **Start (green):** a special screen that has only a single exit point – Load. For more details, see Start Screen.

A new game will have a 'standard' layout based on your choices on the Creator screen. You can add additional screens by right-clicking on the background. A popup menu will ask you what type of screen you want to create. You can also duplicate an existing screen by pressing 'D'. To remove existing screens, click on one to select it, and press the [delete] key.

Entry points are on the left, and exit points are on the right.

To link Actions on screens together, click and drag from one named element to another. There is a lot of flexibility, and it is up to the user to determine what connections make sense.

To edit a screen's options, simply click on one of the boxes, and look to the right, and it will highlight, as shown, below.
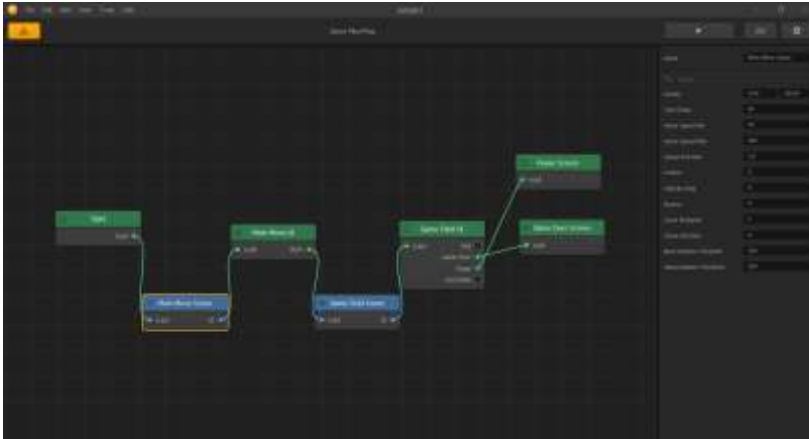
UI Screen Options

- **Name:** Name of the screen for your reference. This will appear on the mind map and in menus.

- **Ad Interstitial:** what ad network will show on this screen. Note: if you select an ad network that is not on the platform you are exporting, no ads will show on this screen (eg if you select Vungle and export to PC, there can be no ads shown as there is no such thing as Vungle on the PC at the time of writing). Select None, or any of the ad networks, listed, including the following special option:

    - **Custom** for iOS and OSX only – allows you to call custom code for this screen. You could use **this** to implement your own ad lib, or for antyign you want to happen for a particular screen. You need to export the game, then edit the following:

        - AppController.m file:

            - **-(void)showCustomFullscreenAd{}:** This will get called when Buildbox needs to display an interstitial.

            - **- (void)loadingDidComplete{}:**This is not strictly limited to custom ads, however it is called on game load, so you may wish to use it to initialise your code for use later (eg call startup ad caching for faster ad display).

- **Ad Banner:** for screens that support banners, what banner should show.

- **Ad Interstitials Freq.:** 0 means no ads. 1 is every time the screen is shown, 2 is shown every two times this screen is shown, and so on.

- **Ad Banner Freq.:** 0 means no ads. 1 is every time the screen is shown, 2 is shown every two times this screen is shown, and so on.

- **Music:** drag and drop an mp3 file for music to any music to play on this screen. If you use an identical mp3 file for music on several screens, Buildbox™ will currently store that several times. Note that there is no need to do this, given that many of screens are overlays (like the pause screen for example) so you would only put in another sound file if you wanted to override the music already playing.

- **Loop Music:** if ticked, the music will play continuously. If not ticked it will play once and stop.

Game Screen Options

Prior to BuildBox™ 2.0, these were the World settings under the game setting, Advanced screen. With the advent of BuildBox's™ multiple game screens, we have multiple worlds ie entirely different game play. To provide that flexibility, the settings were moved, here. The World settings are:

- **Gravity (x and y):** continuous force acting on everything as gravity does in the real world.

- **Time Warp:** how long to wait before adding 'speed increase' to the current game speed.

- **Game Speed Min:** start speed at which game will move past the player (regardless of player movement).

- **Game Speed Max:** final maximum speed at which game will move past the player (regardless of player movement).

- **Speed Increase:** amount to add to speed increase.

- **Friction:** friction between all objects.

- **Velocity Drag:** force used to slow anything that is moving down. Directly opposes all movement unlike air-drag for character, though very similar.

- **Bounce:** bounce force between all objects

- **Score Multiplier:** Scoring is 1 point per pixel moved. A value here will increase that. Eg value of 10 will mean 10 points per pixel.

- **Game Direction:** In clock-wise degrees where 0 has the character moving left. -90 will be moving up, 90 will be moving down, and 180 will mean moving left.

- **Back Deletion Threshold:** How far off the back of screen – behind the Player – do objects need to be before being deleted.

- **Sides Deletion Threshold:** How far off the 'sides' of the screen – perpendicular to Game Direction – do objects need to be before they are deleted.

Screen Names

Prior to Buildbox™ 2.0, screens and screen names are fixed. If you have an old project that successfully upgrades, you should see most screens coming through with their old names. The list of old names is listed here to show you how you may use multiple UI Screens and Game Fields.

- **Start:** first thing user sees when game is launched. A special screen with no options on the Game Mind Map screen.

- **Main Menu:** as it says, the "Main" menu - the purpose for this to allow users to navigate to other screens.

- **Shop:** This is where you will sell IAPs if you have any, for platforms that support IAPs.

- **Game Field:** shows when the game is playing. It is where you will create any on-screen buttons of for movement, etc of the character.

- **Game Over Screen:** shows when the game has ended and the game lost.

- **End Screen:** this is the screen displayed when the game has been won – by collecting an Action with an *Action Type* of "End Game".

- **Pause Screen:** shows when the game has been paused.

Game End Screen

As of Buildbox™ 2.0 there is no Game End screen. You can choose to send the player to a UI Screen or another Game Field Screen, depending on what you link the *End Game* action to.

Start Screen

This is a special screen that only appears on game-load. Once set up, you can see if only very briefly from the Preview screen (because the assets load very fast on a Mac/PC). It optionally has the starting image, and a loading bar and labels.
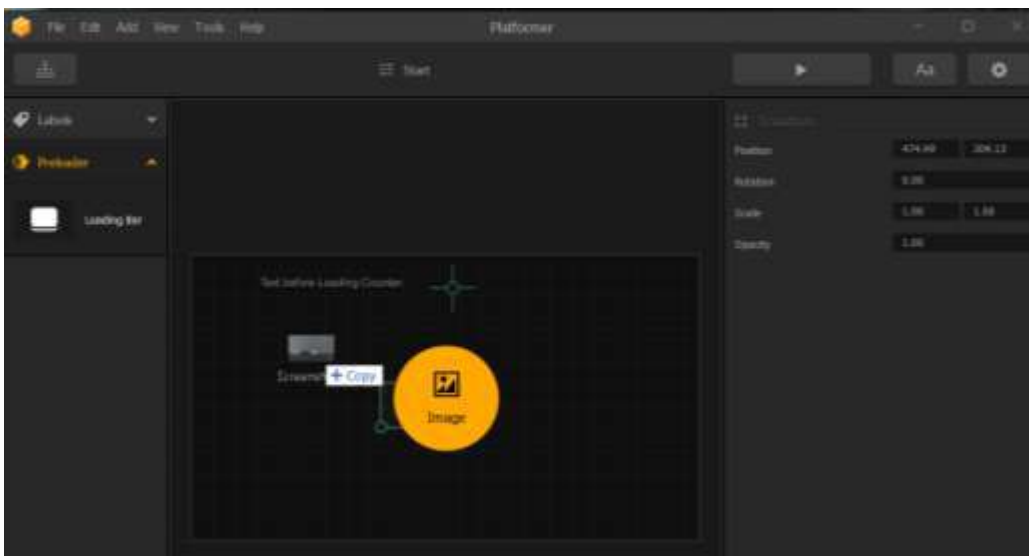
Change Background Image

Simply drag and drop an image onto the editor – as you can see in the screenshot. There is only one option – Image. Keep in mind this image will need to make sense on various aspect ratios for different screens.

Loading Bar

To add a horizontal loading bar, simply drag this from the left, onto wherever you want it to be. You can have more than one, should you wish. The settings are:

- **Position (x,y):** as for any other placed item.

Labels

This is for display of fixed text, and a starting counter. The available settings are:

- **Name:** Internal reference only.

- **Position (x,y):** as for any other placed item.

- **Rotation:** angle in degrees that this should be rotated.

- **Scale (x,y):** You would usually leave this at a value of 1,1.

- **Opacity:** how transparent the text will be.

- **Function:**

  o **User Text:** fixed text

  o **Loading Counter:** numeric indicator of Loading Bar.

- **Alignment (Loading Counter only):** text alignment.

  o **Left**

  o **Centre**

  o **Right**

- **Text (User Text only):** text to display.

- **Font:** Select from the list of Fonts you have defined.

- **Tracking (User Text only):** adjust the spacing between characters.

- **Stick To Edge:** Tick to position text by the edge, instead of floating in the middle.

- **Autohide:** if non-zero, the amount of seconds before the text is hidden.

UI Editor

The screen will look something like this the first time you open it:

Buttons and Image – Quick Start

Drag in a PNG file in as shown below, and you can create some UI elements.

Depending on what segment you drop it, you will get a different element. The more flexible, and common way of setting up UI elements is to use the left side bar and drag them onto the edit area. The next heading will detail all of those.

Buttons

This screenshot shows the palette of buttons available on the left, and an example of each button in use – as seen when no image has been set. The default name for each of is the button type.

Leaving the image unset allows you to have a transparent button in the game. These are commonly used for games controlled by pressing the screen – a single screen sized transparent button is setup and tied to the Jump action.

Action Button

These are used for triggering Actions you have setup in the game Worlds. The following settings are available:
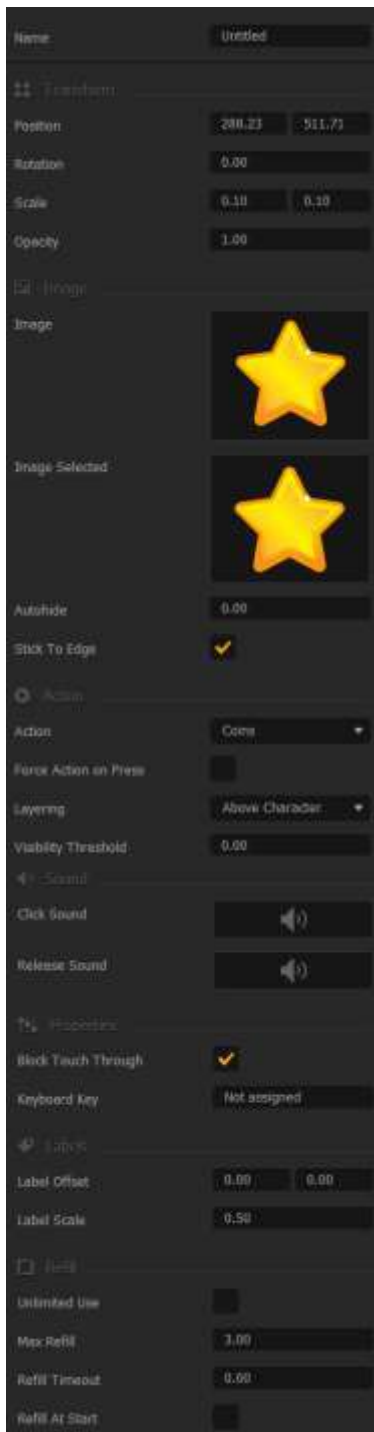
- **Name:** internal name for your reference only. Appears on the layout area so you can see what you are doing - especially useful for a hidden button.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** drop a PNG file here to change the image normally shown.

- **Image Selected:** the image shown when it is selected.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Action:** what the button does when pressed. List of options is made up of the Actions you have defined for the game World(s) this is attached to.

- **Force Action on Press:** Action will occur immediately when this is pressed.

- **Layering:** Where will the activation animation of the Action appear?

  o **Above Character:** animation is in front of the character – for things that you want to obscure your character slightly.

  o **Below Character:** animation behind character – would look good for a starburst to highlight the character.

- **Visibility Threshold:** the amount of uses (ie button presses) that need to be available to show this button. Otherwise it will be hidden.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind. This is useful if you have a screen-size hidden and you want your pause button to work without triggering the full screen button. Just be sure and tick this box on your pause button.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

- **Label Offset (x,y):** where to show the uses the player has left of the Action, if limited.

- **Label Scale:** numeric value from 0 to 1 which will scale down from the standard size font defined for labels, generally.

- **Unlimited Use:** will be unlimited if this is selected. If NOT selected, the next three options will be available.

- **Max Refix:** amount of Actions after game install or after resetting game, that you can earn through the timer.

- **Refill Timeout:** amount of time, in minutes, to wait before you can earn an additional Action.

- **Refill at Start:** refill Action to max refill each gameplay (instead of initial game install or reset).

Audio Button
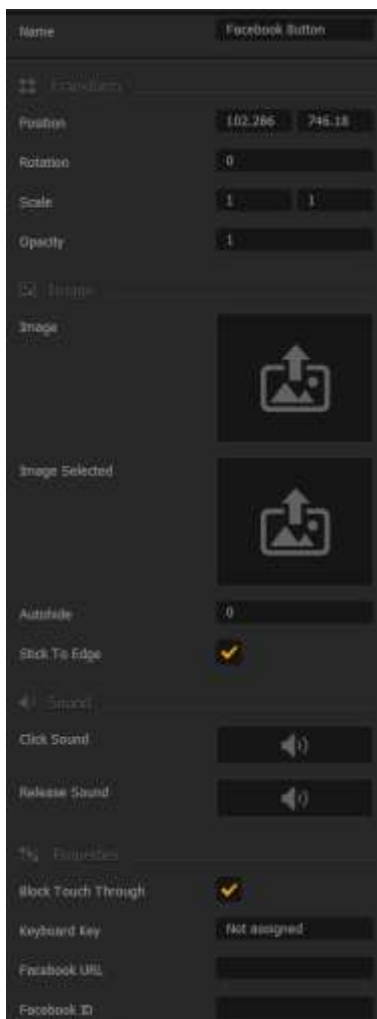These are used to turn the sound on and off. The available options are:

- **Name:** internal name for your reference only. Appears on the layout area so you can see what you are doing - especially useful for a hidden button.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Sound is ON:** drop a PNG file here to change the image shown when specified sounds are on.

- **Sound is OFF:** drop a PNG file here to change the image shown when specified sounds are off.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Function:** - what sound to turn on/off.

   o **All Sound**

   o **Sound Effects**

   o **Background Music**

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind. This is useful if you have a screen-size hidden and you want your pause button to work without triggering the full screen button. Just be sure and tick this box on your pause button.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)
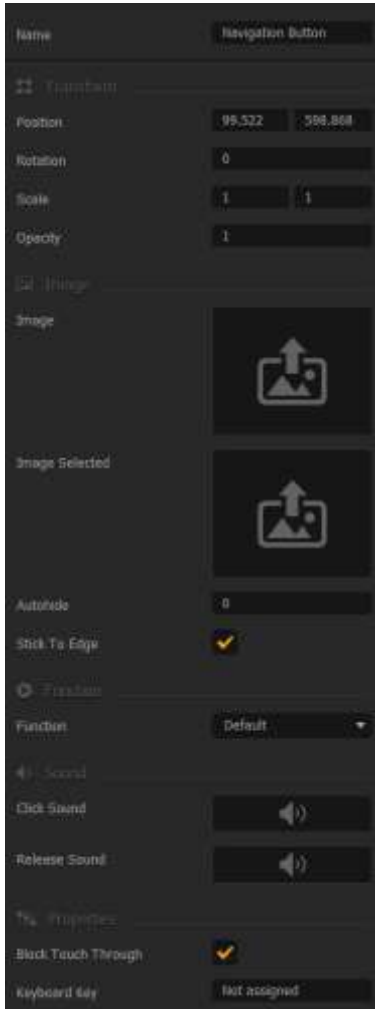
Facebook Button

The ubiquitous "Like" button has made it to Buildbox™ by popular request. Creates a like for the logged-in user, on their Facebook account. Drag the icon from the toolbar on the left, to the menu screen to create an instance. Has the following settings:

- **Name:** internal name for your reference only. Appears on the layout area so you can see what you are doing - especially useful for a hidden button.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind. This is useful if you have a screen-size hidden and you want your pause button to work without triggering the full screen button. Just be sure and tick this box on your pause button.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

- **Facebook URL:** This is the URL that will be liked by the user when they press the button.

- **Facebook ID:** A required id from Facebook to enable this feature. For more information, go here - https://developers.facebook.com/apps - and press [Create A New App].

Navigation Button

A big simplification in the update for Buildbox™ 2.0 is around navigation buttons. With the change to an unlimited number of UI Screens and Worlds, Navigation Buttons were born. Just create a Navigation button with the correct Function setting, and you will see an additional exit selector on the Game Mind Map for that UI – complete with the name you give it, here for ease of reference. The available settings are:
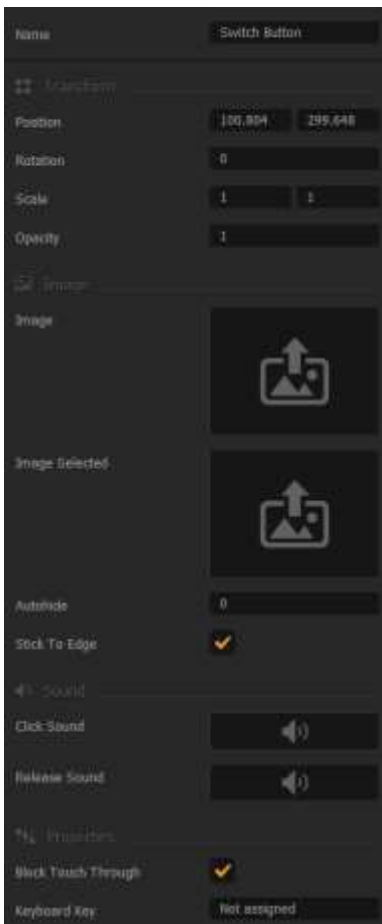
- **Name:** internal name for your reference only. Appears on the layout area and on the Game Mind Map for Default and Pause Function type.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Function:** set to allow navigation to another screen (setup on the Game Mind Map), or special function.

    o **Default:** exit to another screen.

    o **Pause:** exit to another screen.

    o **Back:** previous screen.

    o **Back Jump:** selecting this displays **Back To** setting.

    o **Restart:** Restart current World.

    o **Restart Checkpoint:** Restart game at last Checkpoint.

    o **Review:** take player to review link (see General in Project Settings) for the current App Store, when this game is built.

    o **Share:** allow player to share the game via email, sms, etc (depending on platform). Will use Share Message defined for the current App Store, when this game is built.

    o **Restore Purchase:** For App Stores that support it, this will restore non-consumable purchases (used when player has purchased IAPs and now loaded the game on a new device – so they can access what they have already paid for on the old device).

    o **Game Services:** Show Game Services (Apple App Store) or other App Store equivalent if supported.

o **Exit:** Quit game entirely.

o **Reset Settings:** Reset game to original state. Will display the warning configured in General Project Settings before proceeding.

- **Back Jump(appears when Back Jump selected):**

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind. This is useful if you have a screen-size hidden and you want your pause button to work without triggering the full screen button. Just be sure and tick this box on your pause button.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

Note: on Android devices, the system back button corresponds to the "Back" button Function . On game-play screen it will bring up the pause screen, and on pause screen it will go back to gameplay (this is to stop a double-tap of the button from exiting the game by mistake).
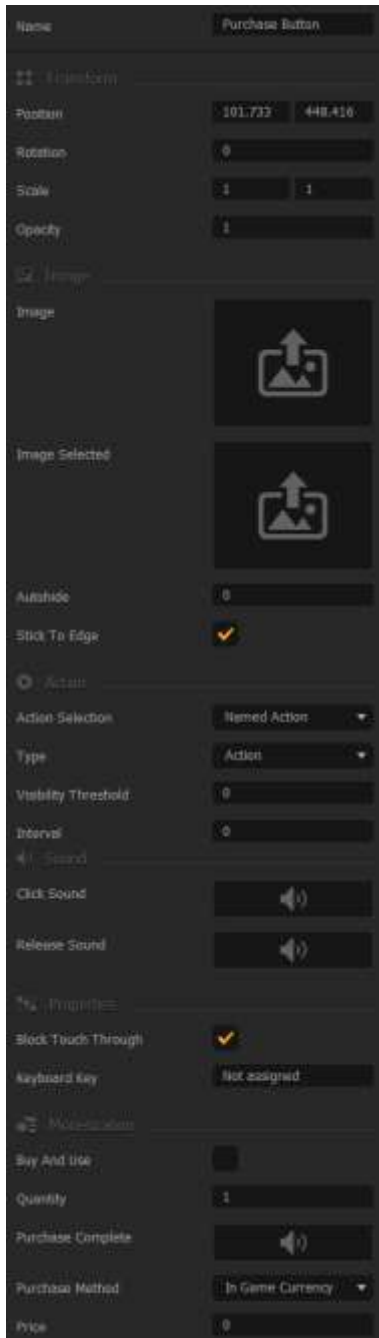
Switch Button

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

Purchase Button

Allow player to purchase items. Typically found in a game shop screen, or other areas where you want people to purchase. Available options are:

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Type:** type or purchase.

  o **Action:** allow purchase of an Action. Action Selection will appear, when selected.

  o **Remove Ads: a**llow ad suppression to be purchased.

- **Action Selection:** a list of all Actions you have defined in your Worlds. What you select will be purchased with this button.

- **Visibility Threshold:**

- **Interval:**

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind.

- Keyboard Key: This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

- **Buy and Use:** use Action item immediately when purchased.

- **Quantity:** number of Actions a purchase will purchase. Does not apply to **Remove Ads** type.

- **Purchase Complete:** an mp3 dropbox for a sound to play on purchase.

- **Purchase Method:** How the item can be purchased. Additional options will appear based on a selection from the following options:

  o **In Game Currency:**

    ▪ **Price:** cost in in-game currency (ie coins).

  o **In App Purchase:** in-app purchase (IAP) through an app store (purchase with real money.

    ▪ **Store Identifier:** Price is set on whatever company's servers are used to drive the purchase – Apple/Google/etc).

  o **Rewarded Video:** User will watch a video completely to earn this item.

    ▪ **Ad Network:** What ad network will be used to attempt to display the ads. Currently Heyzap is the only option.
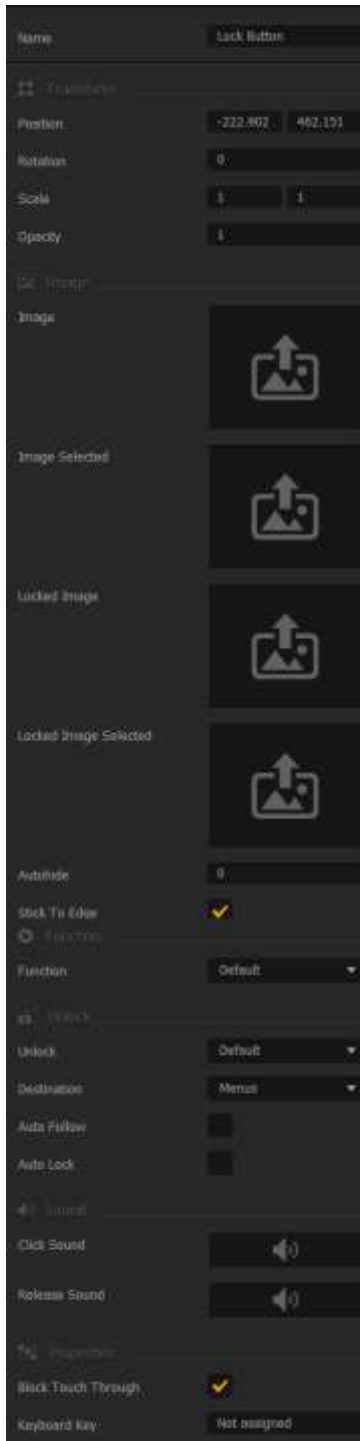
URL Button

Put this in and the user will open a new web browser window when the user clicks it, on platforms that support browsers.
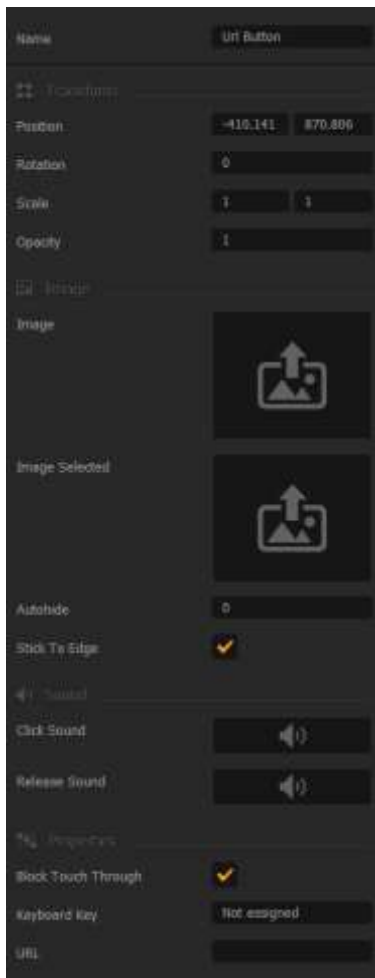
- **Name:** internal name for your reference only. Appears on the layout.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

- **URL:** standard URL that will be visited when use clicks it. Include the http:// (or other link prefix. This could be any protocol the local browser supports and not just web – eg a protocol defined by an app on the device, used to launch that app).

Lock Button

This is for buttons that show inactive at first, and need to be unlocked by purchase, game progress, etc. This is very flexible, and can be seen in Stage Clear starting game template. Available options are:
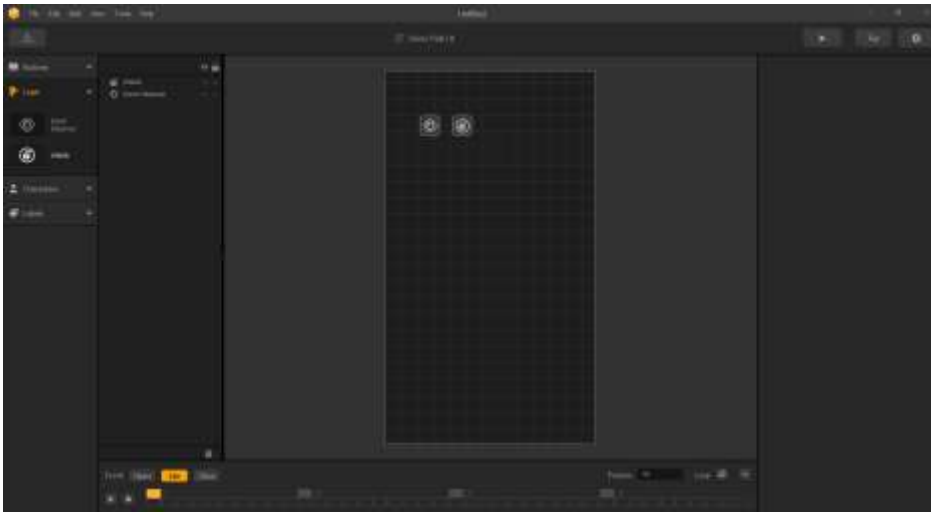
- **Name:** internal name for your reference only. Appears on the layout area and on the Game Mind Map for Default and Pause Function type.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Locked Image:** PNG dropbox for locked image.

- **Locked Image Selected:** PNG Dropbox for locked image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Function:** set to allow navigation to another screen (setup on the Game Mind Map), or special function.

  o **Default:** exit to another screen.

  o **Start Checkpoint:** exit to another screen at checkpoint.

- **Unlock:** methods of unlocking the button

  o **Default:** you must use an Unlock Logic asset.

  o **In Game Currency:** game currency will unlock.

    ▪ **Price:** in-game currency amount.

  o **In App Purchase:** player must purchase an IAP

    ▪ **Store Identifier:** for the app store item.

- **Destination:** set where this will take you once unlocked.

  o **Menus:** a menu screen ie UI you can attach on the Game Mind Map

 o **Scene:** shows two more options to set the Scene:

- **World:** the World where the Scene is found. You MUST attach this selected world in the Game Mind Map.

- **Scene:** scene in the above-selected World that is your destination.

  o **Next Scene:** used for stage-clear type games. Player will stay in a single level until it is complete (ie this is unlocked based on the rules you define) and then move to the next Scene in the World). This only works on UI screens attached to a World, of course.

- **Auto Follow:** as soon as you click the Button to unlock it (with currency or whatever you set it to) it will be activated and take you to the defined destination. Useful for a skip level button, or similar.

- **Auto Lock:** every time you see this button on a new level it will be locked, and require unlocking. This makes it easier to use the same UI for a collection of levels, while still starting each level with a "fresh" locked button.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind. This is useful if you have a screen-size hidden and you want your pause button to work without triggering the full screen button. Just be sure and tick this box on your pause button.

- **Keyboard Key:** This allows you to add keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

Logic (UI)

These are similar to Logic assets found in Worlds, however they are used for controlling the user interface.

Event Observer
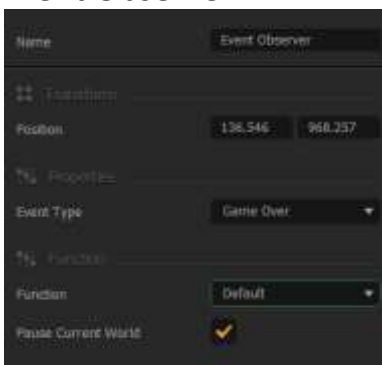
Use these to respond to game events and take action like loading another World or UI screen. They will only have events to react to when the UI screen is attached to a World, of course.

- **Name:** internal name for your reference only. Appears on the layout area and on the Game Mind Map

- **Position(x,y):** location on screen.

- **Event Type:** what event to listen for.

    o **Game Over**
    o **Session Distance**
    o **Session Points**
    o **Session Coins**
    o **Total Distance**
    o **Total Points**
    o **Total Coins**
    o **Best Distance**
    o **Best Points**
    o **Best Coins**
    o **Last Scene**
    o **Second Play**

- **Session Total:** available only for Event Types that require it – like Session Points.

- **Function:** how to start the screen (really only has an effect if loading a World)

    o **Default:** load World or UI at the beginning.

    o **Start Checkpoint:** load World at checkpoint.

- **Pause Current World:** freeze current World so player can come back to it in its current state.

Unlock (Logic)

These are used in conjunction with Lock Buttons that are set to Default Unlock, or to unlock Characters

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Unlock Type:** what to unlock

  o **Unlock Character**

  o **Unlock Button**

- **Object:** a list of all Character Buttons or all Lock Buttons defined in any UI screen (depending on what sort of Lock Type you selected, above) that you can link this to. With Unlock Button it will also show these special options in the list typically used for a menu similar in style to Rovio's Angry Birds:

  o **Unlock Next Scene Button:** unlock the next Scene – without having to configure an Unlock logic for each button.

  o **Unlock Current Scene Button:** unlock the current Scene – without having to configure an Unlock logic for each button.
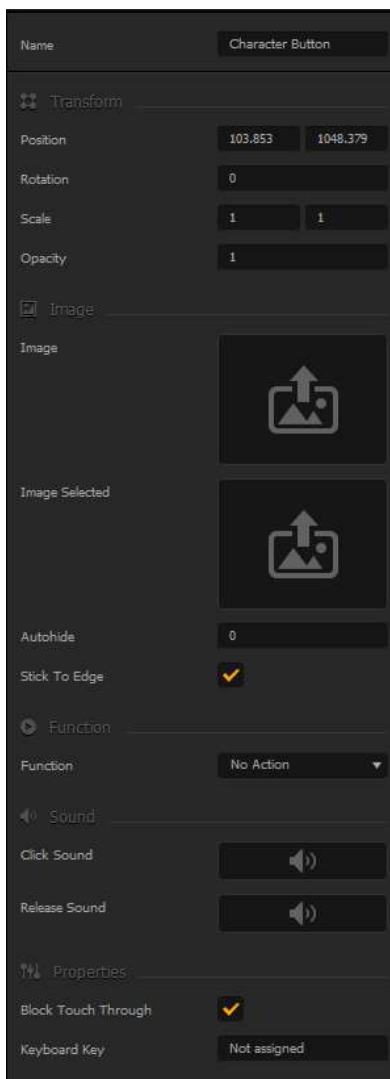
Characters (UI)
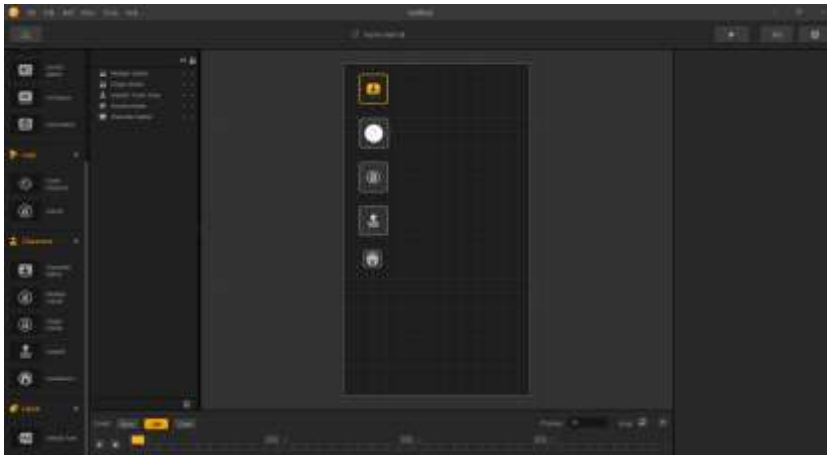
These are all about selecting and controlling the Characters. You can see all of the available assets on the screenshot, above.

Character Button

This is for defining button and or redefining keyboard controls of the game Character. Options are:
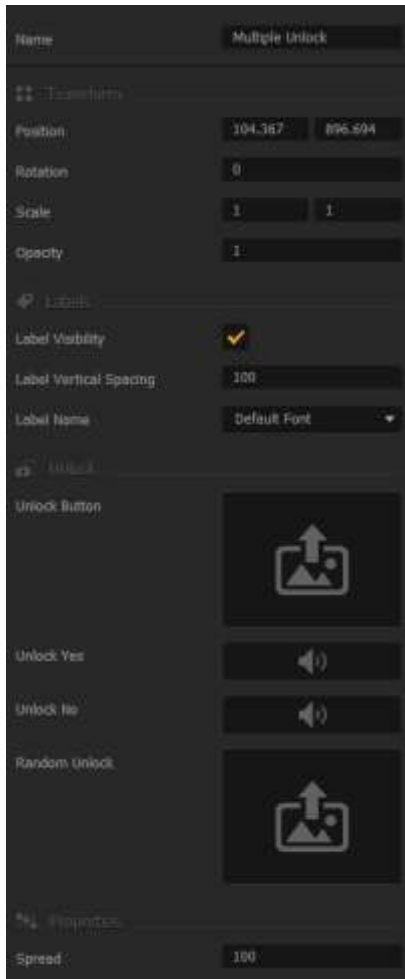

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Image:** PNG dropbox for standard image.

- **Image Selected:** PNG Dropbox for image when pressed.

- **Locked Image:** PNG dropbox for locked image.

- **Locked Image Selected:** PNG Dropbox for locked image when pressed.

- **Autohide:** number of seconds to show before hiding.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Function:**

    o **No Action**

    o **Motors Spin CW:** clockwise.

    o **Motors Spin CCW:** counter clockwise.

- o **Jump**

- o **Shoot**

- o **Rotate Left**

- o **Rotate Right**

- o **Move Right**

- o **Move Left**

- o **Move Up**

- o **Move Down**

- o **Unlock Random Character**

- o **Component Action**

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind.

- **Keyboard Key:** This allows you to add/redefine a keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)

Multiple Unlock

This used to be called Character Selector. This shows a carousel as you add more characters so you will never run out of space. Place this on a screen and resize till you are happy with it.   These options are available:

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Label Visibilty:** tick to display the label.

- **Label Vertical Spacing:** how much space between Character and label.

- **Label Name:** Which of your defined fonts to use.

- **Unlock Button:** a PNG dropbox for what to show as an overlay for a locked character.

- **Unlock Yes:** mp3 dropbox for selection when the character is unlocked

- **Unlock No:** mp3 dropbox for attempted selection when character is still locked.

- **Random Unlock:** Image to show if randomly unlocked.

- **Spread:** what is the space between characters to show.

Single Unlock

This is the Character unlocker for a single character only. This gives you the highest level of control, but may not be practical for games with large amounts of Characters.

- **Name:** internal name for your reference only. Appears on the layout area.

- **Position(x,y):** location on screen.

- **Rotation:** the angle the button is on.

- **Scale(x,y):** how the image is scaled.

- **Opacity:** how transparent the button is.

- **Stick to Edge:** button will stay on the edge of the screen when scaling is done for different sized screens.

- **Click Sound:** sound made when pressed.

- **Release Sound:** sound made when button released.

- **Block Touch Through:** If this button is in front of another button, do NOT pass the touch through to the buttons behind.

- **Keyboard Key:** This allows you to add/redefine a keyboard control. It works the same way as the game-wide keyboard controls: refer (page 82)
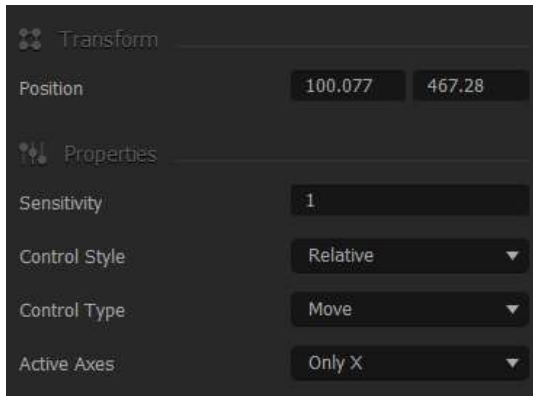
Joystick
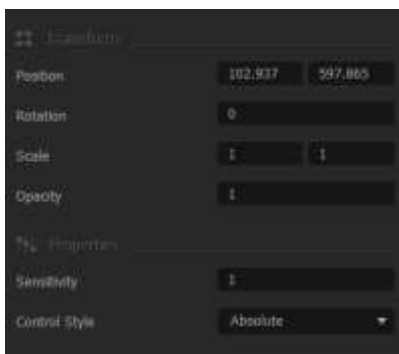
This allows you to have an on-screen joystick. To create one, drag the Joystick button from the left onto the game-field.

Resize and place as appropriate to whatever portion of the screen you want the joystick control to be available. The following options are available for Joystick:

- **Position(x,y):** where on screen to place it.

- **Rotation:** has no effect.

- **Scale(x,y):** determines how big the joystick is.

- **Opacity:** how transparent the button is.

- **Sensitivity:** a value of 1 will mean the character will move at the same speed as your finger. A smaller value will see less movement. Eg 0.5 will move at half speed, and 2 will move at double your finger speed.

- **Control Style:** How the joystick will work. Has the following values:

  o **Absolute:** you drag your character around the screen.

  o **Relative:** Works like a standard analogue joystick. The more you move from the center position in any direction, the faster you will change to move in that direction.

Accelerometer

This control works quite similarly to the on-screen joystick.
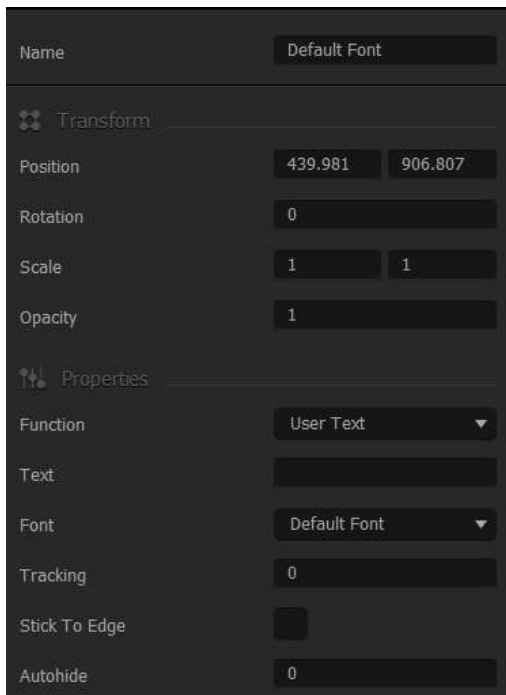

The following options are available:

- **Position:** Has no effect.
- **Sensitivity:** Default is 1. Lower values are less sensitive and higher values are more sensitive – in the say way as for the joystick control.
- **Control Style:** has these options:
    - **Absolute:** similar to absolute on the joystick control. Moving the device off center will move the Character to a given location on screen.
    - **Relative:** The more you tilt from the center position in any direction, the faster the character will change to move in that direction.
- **Control Type:**
    - **Move:** tilting will move the Character.
    - **Rotate:** tilting will rotate the Character. Tiling up or right will rotate right, and tilting left or back will rotate left.
- **Active Axes:**
    - **Only X:** constrained to only X axis (left to right)
    - **Only Y:** constrained to only Y axis (up and down)
    - **Both:** normal function.

Labels

To add a label, drag the Aa button down the left of screen. You will see the following options:

Name	Default Font
Transform	
Position	439.981 906.807
Rotation	0
Scale	1 1
Opacity	1
Properties	
Function	User Text ▼
Text	
Font	Default Font ▼
Tracking	0
Stick To Edge	
Autohide	0

- **Name:** for your reference, only.
- **Position(x,y):** where on screen to place it.
- **Rotation:** has no effect.
- **Scale(x,y):** determines how big the text is.
- **Opacity:** how transparent the text is.
- **Function:**
  - **User Text:** fixed text.
  - **Score**
  - **Start Scene Name**
- **Text (if User Text selected):** text to show.
- **World (if Score selected):** world for score.

- **Score Type (if Score selected):**
  - **Distance**
  - **Points**
  - **Coins**
- **Amount (if Score selected):**
  - **Best**
  - **Current**
  - **Total**
- **Alignment (if Score or Start Scene Name selected):** text alignment
  - **Left**
  - **Centre**
  - **Right**
- **Font :** which defined font to use.
- **Tracking (User Text only):** adjust the spacing between characters.
- **Stick To Edge:** Tick to position text by the edge, instead of floating in the middle.
- **Autohide:** if non-zero, the amount of seconds before the text is hidden.

Font Editor

This screen is where you set the font used in labels throughout your game. The fonts available are all fonts installed on your computer, so if you want more fonts for Buildbox™, just install them as per normal on your computer.

After selecting your font, you can then set the following options:

- **Name:** internal name for your reference only.

- **Font Name:** as above.

- **Font Size:** how big the font is.

- **Shadow Offset(x and y)**: if either value is not zero, and **Draw Shadow** is selected, a shadow will appear.

- **Canvas size:** size of the sprite sheet used for font.

- **Padding:** how much space in pixels between each letter.

- **Stroke Width:** The edge of the fonts will be drawn with a pen of this width, if **Draw Stroke** is selected.

- **Main Color:** main color of the font.

- **Gradient Color:** color to fade to from main, if **Draw Gradient** is selected.

- **Stroke Color:** Color if the font outline if **Draw Stroke** is selected.

- **Shadow Color:** what color the shadow should be, if **Draw Shadow** is selected.

- **Draw Shadow:** if selected a drop shadow will be drawn.

- **Draw Gradient:** if selected a gradient will be drawn on the font.

Collision Shape Editor
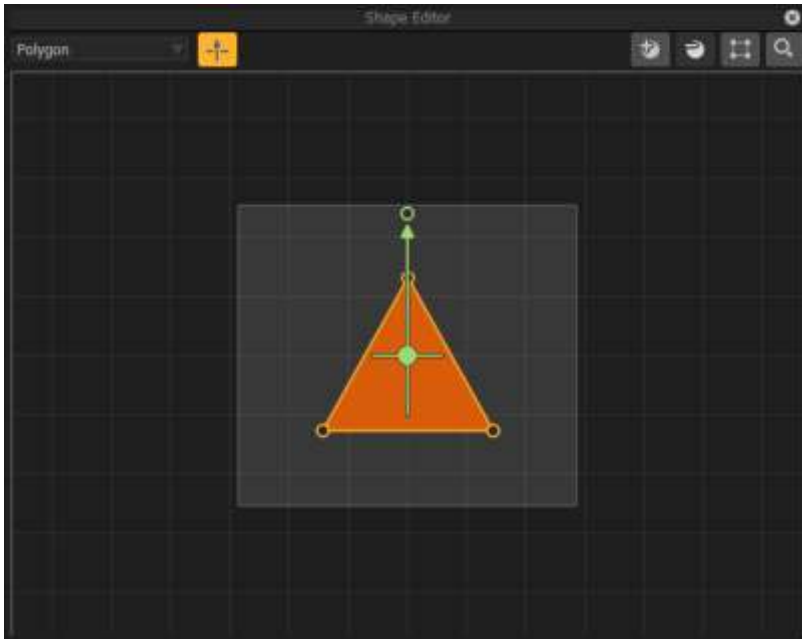
The collision shape could be said to the most important part of your game, as without them, you came cannot function. They are used every time your game kills an object, bounces an object, or even just stops an object. All these things happen when two game element's collision boxes collide, or when an object's collision box collides with the game boundary.

What is NOT used for any of these things is the sprite image itself - this gives great flexibility to you as the game designer.

Altering an object's collision box is very similar to editing your game's boundary, though it is more flexible.   You can change a collision box by clicking and dragging and dropping any of the points (shown as small circles) that define it.

At the top of screen are several elements to help you:

- **Shape:**

    o **Polygon:** collision box can have effectively unlimited points.

    o **Circle:** the simplest collision box to define. You can make this bigger and smaller and it will remain a circle.

- **(One-way collision):** turn on or off the one-way collision option – which turns on a green pointer. This can be seen in the above screen-shot. To change the direction of the one-way collision click and drag on the end of the arrow.   The direction the arrow is pointing, is the direction that the player's Character will be able to pass through the collision box.

- **(Add):** select a point, and click this button, and a new point will be added next to your selected point.

- **(Remove):** select a point, and click this button to delete the selected point.

- ▣**(Reset):** reset the collision box to the original "best guess" that Buildbox™ made when importing the PNG file.

- 🔍**(Zoom):** change the zoom level of the viewer.

Notes:

- You cannot have a convex collision box. Don't worry if you are unsure what this means, as Buildbox™ will not let you create a convex box, and will turn the box red to warn you if you do so by accident. If you need a convex box, you should first consider if you really need one. If you are still convinced, you may want to look at Connections to create a composite object.

- Collision boxes for Characters will usually be a bit smaller than the visible boundary of the character for better playability. If too big you run the risk of the user feeling cheated when they get killed and think your game unfair.

- For lower CPU usage, keep your collision boxes as simple as you can.

Project Settings

General

Here is where you configure the details needed for submitting your game to the various app stores.

The very top of the window is a menu selection bar that highlight the current - General - screen in yellow. Click on Controllers or Advertisements to go to those screens.

At the top of screen, below the menu selection bar, you will a drop-area for the game icon. To set your game icon, drag and drop a PNG file. The file will be resized automatically for each platform. The advanced image options screen is accessible from this box.
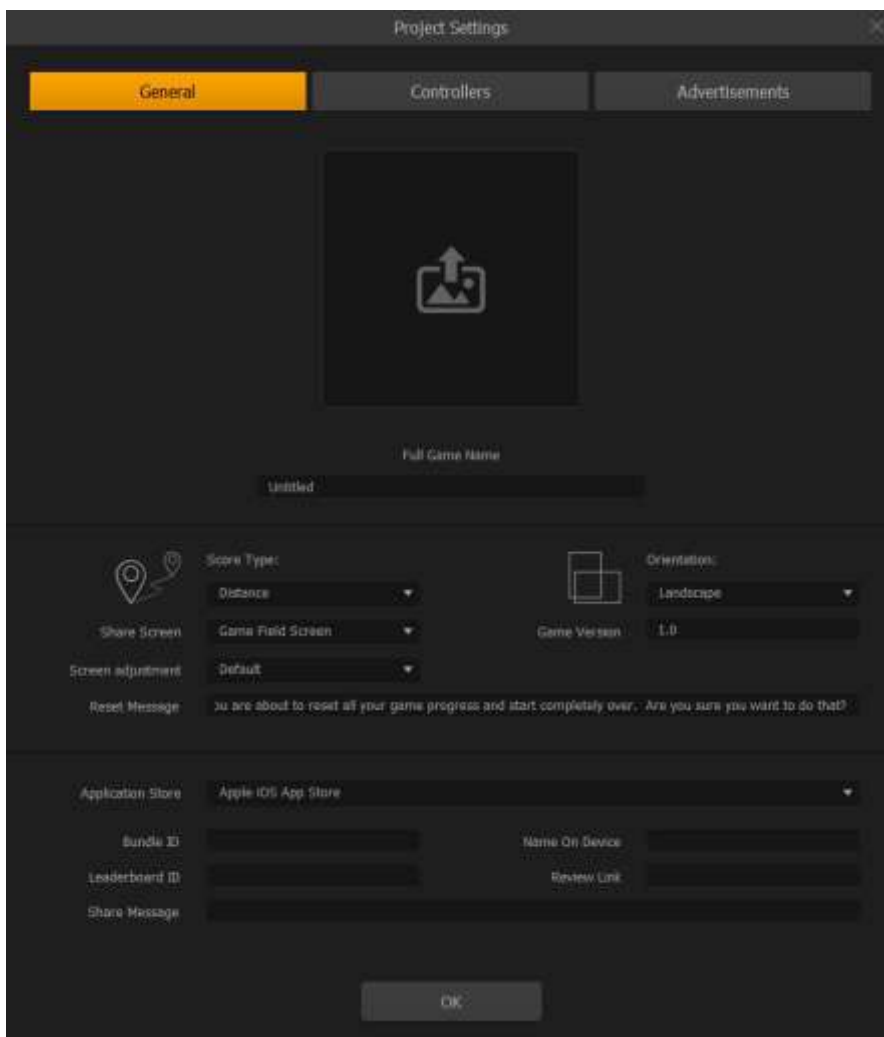
The following settings are available:

- **Full Game Name:** the name of the game.

- **Score Type:**

    o **Distance:** The default value. Scoring is based on distance travelled, only.

- **Coins:** Scoring is based on collecting Action it**em**s that have a coins value. Ie your scoring is the same as your coin collection.

- **Orientation:** which way is "up"

    - **Landscape**

    - **Portrait**

- **Game Version:** the game version. This is used for version control when submitting updates to app stores.
- **Share Screen:** What will be used for social sharing features, if used.

    - **Game Field Screen:** share the game field image

    - **Game Over Screen:** share image of the game-over screen

- **Screen Adjustment:**

    - **Default:** classic Buildbox™ scaling.

    - **Fit to Height:** scale image to fit height of screen. Sides will crop or show more as needed to deal with aspect ratio changes.

    - **Fit to Width:** scale image to fit width of screen. Top and bottom will crop or show more as needed to deal with aspect ratio changes.

- **Reset Message:** This is the confirmation message shown when somebody presses the Reset Game button.

The bottom part of the screen is designed in such a way that you save settings for multiple different app stores (/device types). To look at each of these, click the Application Store drop-down and change the displayed app store. When you export a game for an app store the corresponding settings are used.

- **Application Store:** displayed settings are for this app store. For available options, see list of Exportable App Stores.

- **Bundle ID:** A unique identifier that gets embedded into the app and must be unique within a single app store. Usually in the form of com.short-company-name.short-game-name.

- **Name on Device:** a short name that will be displayed on the device, once the game is installed. This will appear under the game icon. If the this is too long to display, the device will usually abbreviate the middle of it by using ".."

- **Leaderboard ID:** the code-name of the leaderboard. This is what you setup in Apple's Itunes Connect, or Play Developer Console, or Amazon, so the game can use it.

- **Review Link:** link within the app store that will take the player to a page on the app store where your game can be reviewed.

-

- **Share Message:** the message to show to the user when prompting to share the game. If you want to show the players' score in this message you can insert the special tag "[[SCORE]]" into your text. If you want this message to be useful, you will need to put in a link. If you are pushing to multiple platforms, you may want to make this link go to a web pages instead of an app store (perhaps their friend has a different kind of device).

- **Licence Key:** Google Play only – signing key from the game on Google Play.

- **Game Services ID:** Google Play only – ID for Google Game Services. For leaderboards, etc.

- **Google Play ID:** If using Google Game Services (leaderboard) then you grab the ID from setting the game up for Google Game Services.

- **Steam App ID:** ID for the game if you are releasing on Steam.

- **Version Code:** Used for Amazon Game Circle or Google Game Services

- **Amazon API Key:** Used for Amazon Game Circle.

Controllers

The very top of the window is a menu selection bar that highlight the current - General - screen in yellow. Click on General or Advertisements to go to those screens.

Non-click/touch game controls are set here. There are currently two control methods supported – selectable from the **Controller** dropdown.

Keyboard
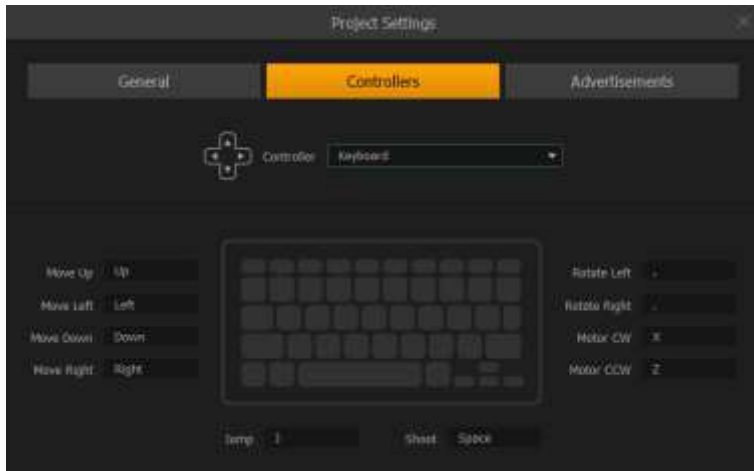
Gameplay

This Buildbox™ setting screen has been removed with version 2.0. Gameplay Presets have been moved to the Creator screen.

Advertisements

The very top of this window is a menu selection bar that highlight the current - Advertisements - screen in yellow. Click on General or Gameplay to go to those screens.

There are several sets of options set to set on this screen, which you get to by changing the value of **Application Store.** Screen-shots of each of these are to follow.

Note: the ad networks compiled into your game are NOT determined by these screens. They are determined by the ad networks you select for the game screens - see Menu Overview.

Best Practice For Heyzap

If you wish to utilise Heyzap – which is a network aggregation service – then don't configure the ad networks here. The correct place to configure the networks is in your account within the Heyzap website. If you configure them within Buildbox then the networks will be initialised twice – once by buildbox and once by Heyzap. Performance will suffer as Heyzap's optimisation system will not know about this.

When using Heyzap for iOS or Android games, after exporting the game manually check the ad networks you have chosen with your Heyzap account for this game. There are a lot of networks you may use – or choose not to – and Heyzap add to this list from time to time. You may decide to add additional networks – or even remove some you decided not to use.

Hence you generally only want the networks you are actually going to use or your game will be too big – to no benefit. (for example if you are not using Facebook ads then it would be silly to add the Facebook ads library.)

Buildbox ships with support for a lot of the supported networks – though not some commonly unused.

Banners/Interstitials Priority List

There is the ability within Buildbox™ to define a priority list – sometimes referred to as a waterfall. A waterfall is a list, in order, of ad networks. If the first network has no ads to display, the second is used, and if it has no ads, then the next is used, until you display an ad or reach the end of the list.

Currently, Buildbox™ supports a waterfall for some ad networks on Android and iOS platforms. Hence you can define a waterfall for banners and interstitials on:

- Apple iOS App Store
- Google Play Store
- Amazon App Store

Not all networks can be used in waterfall at this time. Heyzap for example is not part of this list – though it is available for Google Play and Apple iOS App Store.

The ad network used is still determined by the individual screen settings settings. To use the waterfall you define here, just pick an ad network that is in your waterfall list. The ad network does NOT have to be the first network in that list.
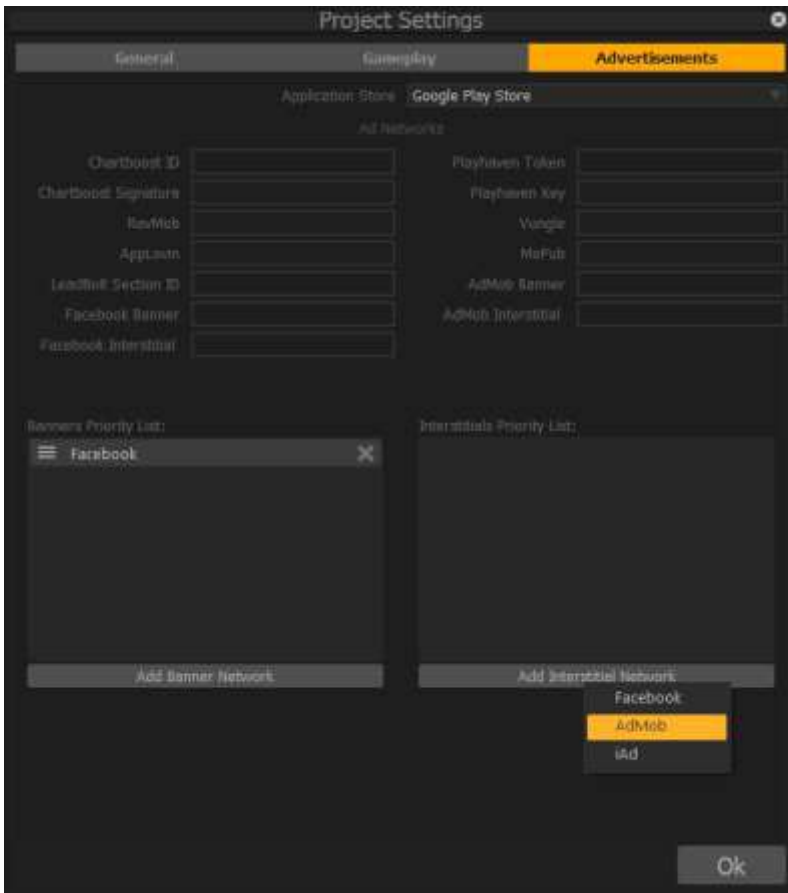
Apple iOS App Store
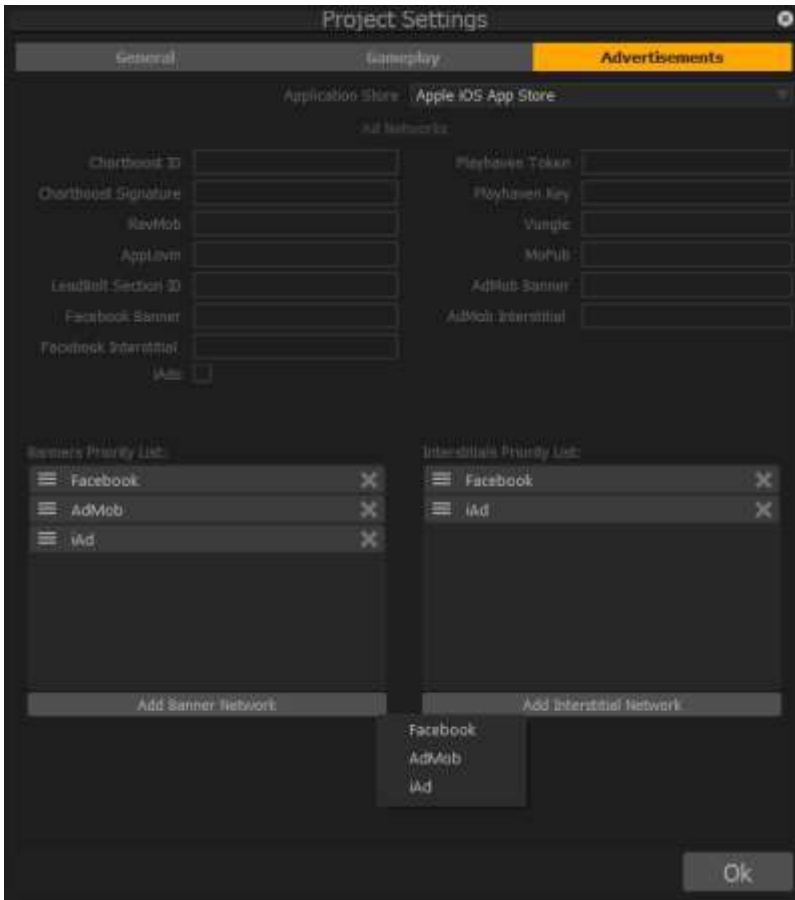
Fill out details for whatever advertising company you will be using. As you can see, there is a wide selection.
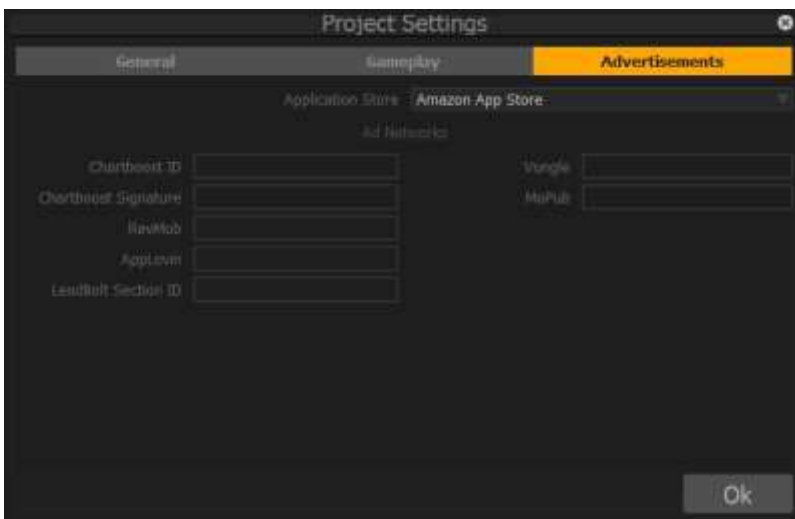
Apple Mac App Store
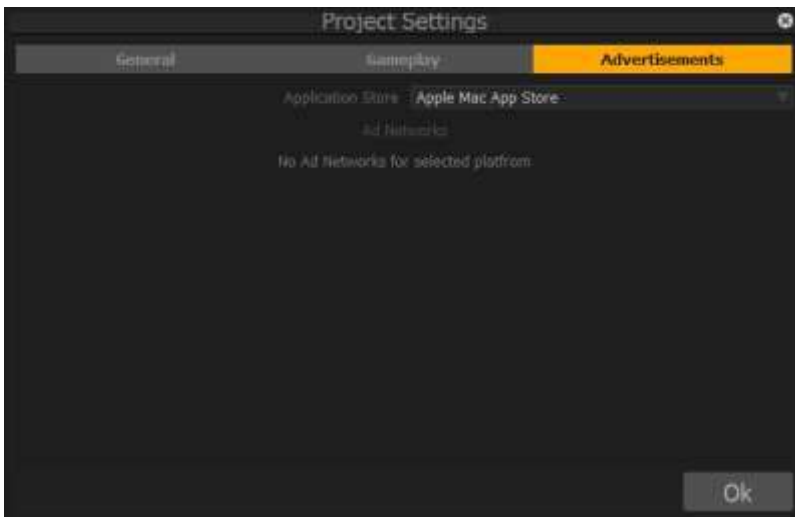
There are no ad networks supported on Mac desktops.
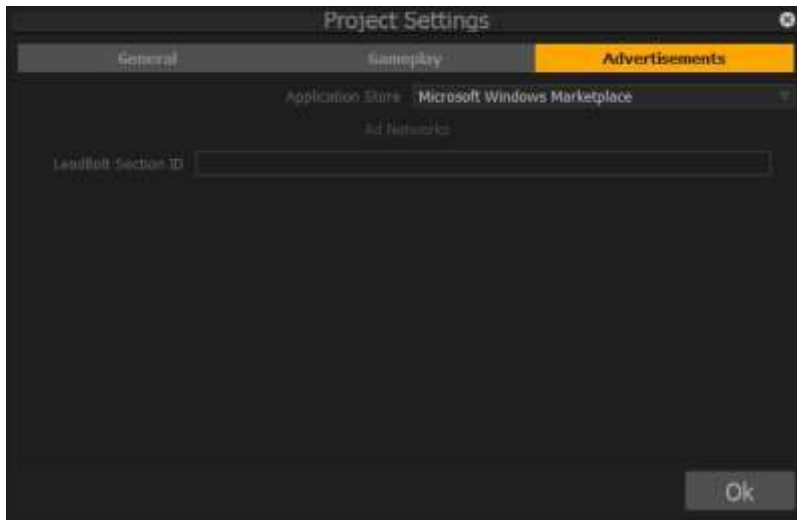
Google Play Store

Fill out details for whatever advertising company you will be using.
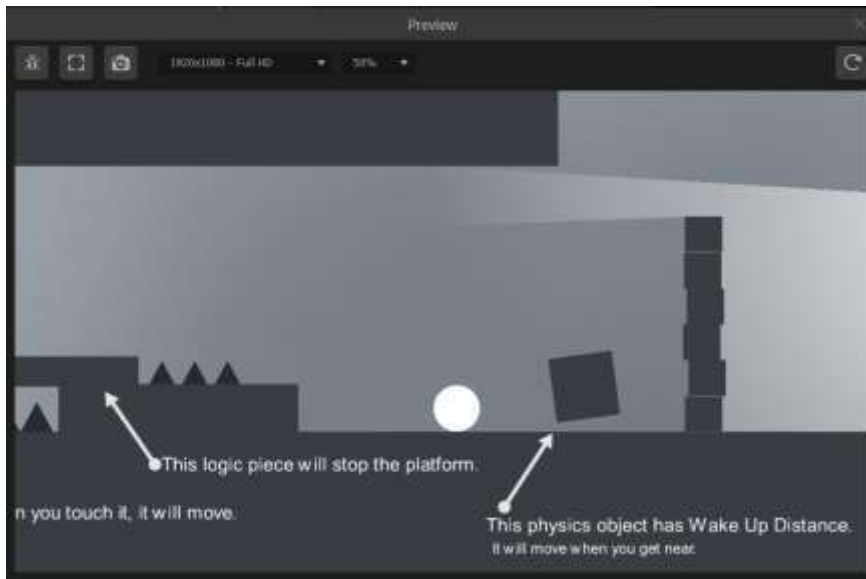
Amazon App Store

Fill out details for whatever advertising company you will be using.

Microsoft Windows Marketplace

The only advertising company supported on Windows (desktop) is Leadbolt.

Preview

There are only a few controls for this screen. The buttons/options from left to right are:

- **Debug mode:** show collision boxes in various colors (like the screen shot, above).

- **Show Fullscreen:** set to full screen. Works well if you have two monitors - once for the editor and one for preview.

- **Take Screenshot:** save an image from the simulator and save to desktop.

- **Set specific screen size: (self explanatory)**

  o **1136x640 - iPhone5**

  o **960x640 - iPhone4**

  o **1024x768 - iPad2**

  o **1920x1080 - Full HD**

- **Zoom:** take set size and scale it for display. This is handy for when you want to do HD screenshots, but without the window taking up the entire screen. Options are:

  o **100%**

  o **75%**

  o **50%**

And on the top right is the restart button. Pressing this is like closing the window and opening it again.

Atlas Editor
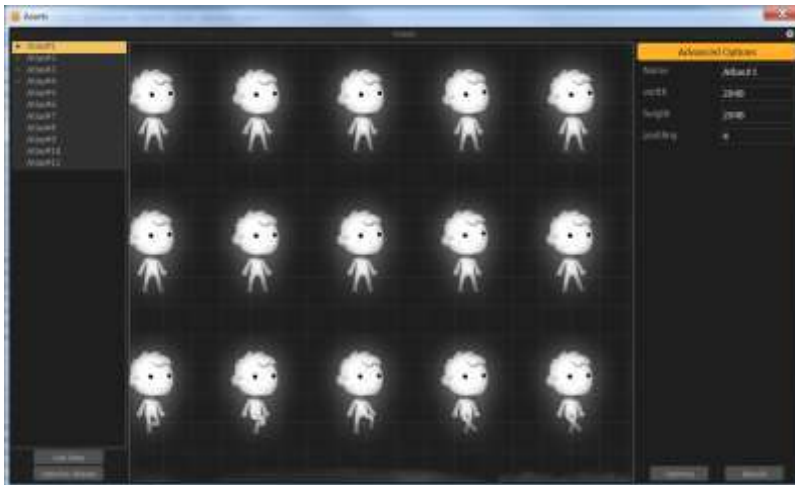
This is an advanced screen provided to edit the atlas/sprite sheet files that will let you tweak them.

The most important thing you can see on this screen - for the non-advanced user - is how many Atlas sheets you can see. The more of them, the bigger your exported games will be. In general, smaller is better - for memory use and download size of the app.

On the left is the list of atlas files. When you click on them, you can edit the advanced options on the right. These are:

- **Name:** name of the atlas file.

- **Width:** width in pixels. This must be a base-2 number (eg 256,52,1024,2048...)

- **Height:** height in pixels. This must be a base-2 number (eg 256,52,1024,2048...)

- **Padding:** how many pixels space between individual sprites. A value of 0, may generate artefacts in your sprites edges is not recommended

If you want to move sprites to another atlas file, just click and drag the image to another atlas on the left side of screen. To create a new atlas file, press the [Add Atlas] button on the bottom left.

To optimise or rebuild the currently selected atlas file (after removing or adding some images press the [Optimise] and [Rebuild] buttons respectively, on the bottom right.

Buildbox™ Menu Summary

Most menu functions are available from buttons elsewhere in Buildbox™. Here is quick summary in case you are feeling a bit lost on some of them.

File

- **New:** abandon the current project (BBDOC file) and create a new one. Show Creator Screen.

- **Open:** abandon the current project and load a new one.

- **Open Recent:** show a list of recent project (BBDOC files)

- **Save:** save current project with its existing file name.

- **Save As:** save current project with a new file name.

- **Export:** export current project. This is what you need to do when you a ready to submit a game to an app store. When you click on one of the export options listed below, you will be asked for a location.

  - **IOS**   (mobile)

  - **Android**   (mobile)

  - **Windows Desktop**   (desktop)

  - **Windows Desktop EXE**   (desktop – needs no compile; unsigned; no ads)

  - **Windows Store**   (mobile, etc)

  - **Amazon**   (mobile)

  - **OSX**   (desktop)

- **About:** show summary of your current Buildbox™ install, including current version installed.

Note: exported projects will then need to be compiled in XCode, Visual Studio, or similar – with the exception of Windows EXE.

Edit

- **Undo:** standard application undo last operation.

- **Redo:** standard application redo last operations.

- **Align Scenes:** layout all scenes in the time-line such that they will play one after the other.

- **Randomize Scenes:** layout all scenes in the time-line such that they will are all in parallel. On play one of these will be randomly selected to show.

Add

- **New Image:** add Image object (only available on UI Editor)

- **New Button:** add Button object (only available on UI Editor)

- **New Action Button:** add Action Button object (only available on UI Editor)

- **New Purchase Button:** add Purchase Button object (only available on UI Editor)

- **New Character:** create new Character.

- **New Object:** create a new Object.

- **New Background:** create a new Background. (only available on Scene Editor)

- **New Action:** create a new Action.

- **New Effect:** create a new effect (light, trail or flag)

View

- **Atlasses:** Show the Atlas editor.

- **Actual Size:** change zoom level to actual size - neither zoomed in or out.

- **Show All:** change zoom to show entire scene.

Following only shown on Game Field editor:

- **Show Simulation:** open Preview screen – as if you had pressed the play button on the right.

- **Debug Mode:** show collision boxes in various colors.

- **Linker Mode:** show linker mode so connections can be set up.

- **Snap Mode:** turn on snap mode so that objects snap to locations on screen (easier to position objects).

- **Lock Background:** stop background from being selectable - preventing accidental background changes.

Tools

- **Remove Unused Level Objects:** delete unused objects, Actions and effects.

- **Remove Unused Image Objects:** delete unused PNG files.

- **Remove All Sound Objects:** remove all sounds from project.

- **Convert to Clones:** see Convert to Clones in Advanced Section.

Help

- **Buildbox Documentation:** Open a web browser and show the main documentation page.

- **Welcome Screen:** Show Welcome screen that also appears when you open the program.

- **Check for Updates:** check to see if there is a new version of Buildbox™ available for download.

- **Get Logs:** support may as you to send them logs. Press this and you will find a log file on the desktop, ready to send to support.

- **Activate Licence:** use this to turn your demo software into a fully licensed product.

Advanced

Motors

Motors are a setting that relates to Connections. If you want the background on this, you should read that first.

Click on the line between two points, and you can set the options for the join. These options let you get objects moving as you can see from these:

- **Speed:** top speed of object around the parent pivot point. Positive numbers will move clockwise.

- **Torque:** how much circular force is used around the parent pivot-point.

- **Button Activated:** if ticked, this will only activate when the motor rotate buttons are pressed. Usually this is used on objects attached to the player character, however they can be used anywhere.

Convert to Clones

This is a high level game optimization. No need to mess with this unless you have already optimized other things like image sizes, number of atlases, etc, and are still looking to make things even smaller and more efficient.

If you have many of the same objects in your game, all with the same properties (ie same size, etc) you can select one of them and select **Convert to Clones** from the tools menu.

It will remove a ton of duplicate data and just store the duplicate objects as clone of the root object. Only the position, rotation and scale will be saved.

The "root" object will contain the other settings which will affect all the clones automatically.

Advanced Game Settings

In Buildbox 2.0, these settings have been moved. Character settings are to be found in in the Character section. For World settings, see Game Mind Map.

Advanced Image Options

Move your mouse over a image box, and you will see a pencil and "X". To delete the image currently in the box, press the "x" however to access the advanced image settings screen click on the pencil. You will see a screen like this:
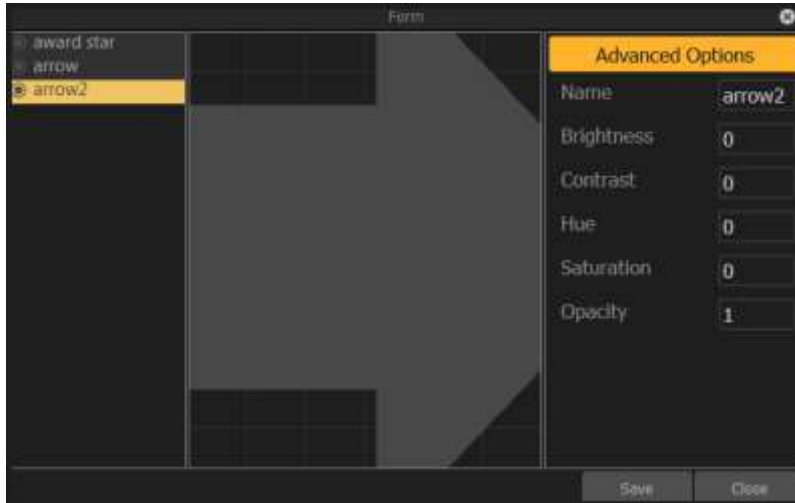
The first thing to note, is the top-left, which allows you to select what image you wish to alter. By default the image that you came from will be selected, but you can edit any image currently loaded into the project (Note: changing the selected image and making an edit will NOT change the image you came from - it will edit whatever image you changed to).

All instances of the selected image will be changed by what you do on this screen.

You can edit the following image properties:

- **Name:** name used internally within Buildbox™.

- **Brightness:** brightness of the image, from -6 to 1 where 1 is the brightest(/white).

- **Contrast:** contrast of the image, from -1 to 1 where 1 has the most contrast and -1 as the least.

- **Hue:** If not 0, will put a color-cast to the image. Values between -2 and 1.5.

- **Saturation:** Sets the richness of the color. Values between -2 and 1, where -1 is completely washed out and 1 is completely saturated.

- **Opacity:** Sets how transparent the image is. Values between 0 and 1 where 0 is completely transparent and 1 is completely opaque.

By default all the values are 0 except for Opacity.

To save changes press [Save] and to leave the screen - without saving - press [Close]

How Do I...

The biggest question the team in support get starts with "How Do I," so we wanted to devote a little time to answering some of these, and if possible give you all ideas on how to answer this question yourself. After all, when you are "in the zone" you don't really want to wait for anything - let alone for support to reply.

Buildbox™ is a very flexible system, in which case there may be more than one way to do what you are trying to do. You may find flicking through this manual may give you an idea. Probably the biggest tip that can be offered is to try NOT to look at items in Buildbox™ as they are defined.

Think laterally. What happens when you take an item out of its usual operating location and put it somewhere else. An example of this are links and attachment motors. Now they are designed for use in creating vehicles that are driven by wheels. If you look up the wheel motors in this manual you will see the example given is completely diverse from that. This was done with the purpose of helping the reader look at things more flexibly.

So to get you going, what are some things you can do with these links and motors, aside from a motorcross bike?

- **unicycle:** attach one wheel to the character.

- **Launcher:** set up two rows of wheels, each side set to turn in opposite directions. Put each row evenly apart, and put an object to be launched in the barrel created. Make all the motors button triggered. When the user presses the button down, the projectile will be launched.

- **Walker:** make a stick with its pivot up one end (ie not in the center). Attach two of these to the character, with opposite turn directions. The player can move the character around by pressing both turn buttons one after the other.

- **Gripper arms:** give your character gripper arms by using two sticks as above, and mounting them parallel to each other, and set them to turn in opposite directions. Now one turn button will cause the gripper arms to grip, and the other button will cause the arms to move part.

Trigger an Event on the Other Side of the Screen?

Put in a transparent prop that holds up whatever it is on the other side of the screen, but which the player can run into. Set it to move at high speed off the screen when touched.

Enemy That Dies on Player Contact

This is here to show you an alternative way of doing things. Now of course you can kill enemies on contact by using Strike, however you can achieve a similar effect by doing the following:

Drag in enemy sprite as an Action, with type coin and a value of 0.

Set **Start animation** as the enemy death animation.

Make a Swinging Object

This was alluded to in the linked objects section. This is what you need to create:

- Create a fixed object "A"

- Create a physics object "B". Recommended set to collision type - no-collide (or shape will stop free movement as it is at the elbow)

- Draw a connection from object "A" to "B"

- Create a physics object "C"

- Add a connection from object "B" to "C"

Object "C" will swing if you have it start high enough, as acceleration due to gravity kicks in.
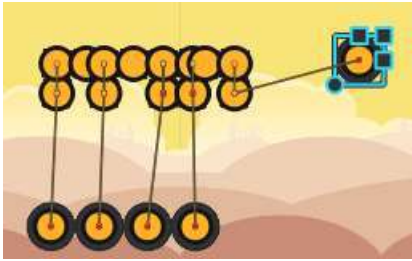
Design Mode

Preview mode

Action with Physical Behaviour

You cannot set an Action to have physical properties or movement. You can, however, attach the Action to an Object that has those properties.

1. Create an object.

2. Give it whatever physical and/or movement properties you want.

3. Optionally make the object invisible (by using a transparent PNG for the object image)

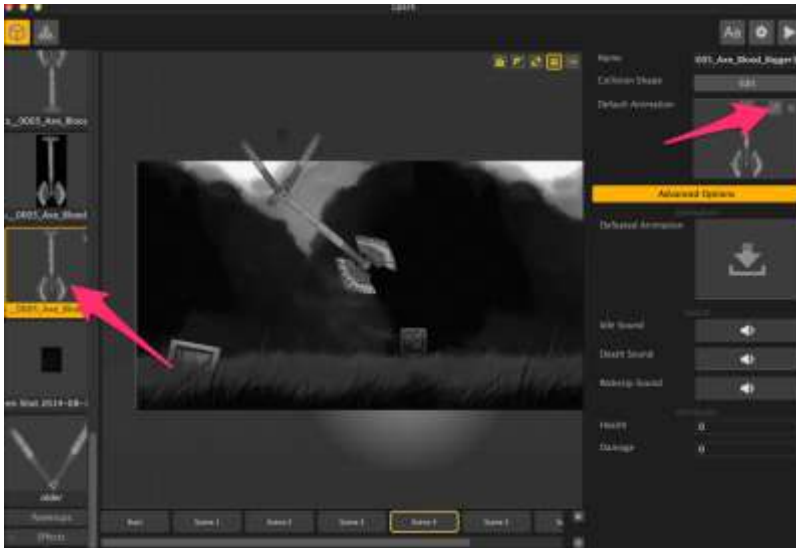4. Add the Action.

5. Join the Object to the Action.

Note: you may make the Action impossible to pick up as the object cn be in the way. Hence you may wish to offset the Action from the object.
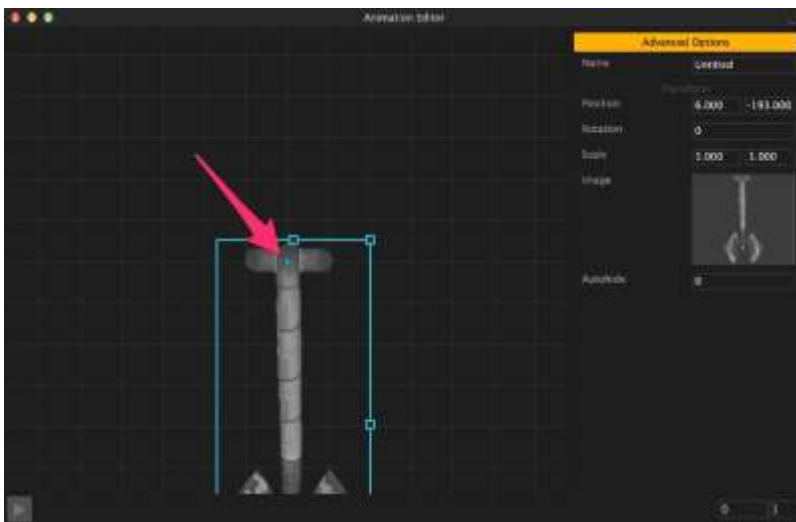
Make a Swinging Axe

An example of this is found in one of the sample games.

To do this you just need to adjust the anchor point in the animation editor. Open the animation editor like this:

Then drag the image around to adjust the anchor point:

Kill Enemies by Jumping on Their Heads

For every enemy that you want to be able to kill by the character jumping on it:

- place a Strike Action - with a transparent PNG.

- Any animation should be single-play.

- Use a link to attach the Action to the enemy.

- Short duration - definitely less than a second.

So what this means is as soon as the character lands on the enemy's head, it picks up and triggers the Strike which kills the enemy.

Online Training Videos

This section has been removed as a more up to date list is always to be found here:

https://www.buildbox.com/tutorials/

The Basics

Drag and Drop

Perform a drag and drop by having both the source and destination visible on screen (for an image file, the source would be a Finder/Explorer window). Move the mouse over the source, press the left mouse button down and hold it. Now drag the mouse to the destination, and let go of the mouse button.

PNG File

A type of image file. Supports transparency and various amounts of colors. Lower color settings will mean a smaller PNG file.

PNG Drop Box

A box where you can drag and drop a PNG file. Once an image is added, you can remove it by pressing the **x** and access the Advanced Image Options by clicking the pencil. You can drop multiple PNG files at once onto some PNG Drop Boxes, creating an animated sprite(/image).

MP3 File

An audio file that uses lossy compression. Lowering the quality settings of the file before import will make the file smaller.

Exportable App Stores

The following app stores are supported:

- Apple iOS App Store - Mac mobile devices including iPhones and iPads. IAP supported.

- Apple Mac App Store - Mac OSX desktop.

- Google Play Store - Default Android app store. IAP supported.

- Amazon App Store - Amazon Kindle and Kindle TV.

- Windows desktop.

- Microsoft Windows App store (previously known as Windows Metro). This is the new universal app all windows devices are moving to – including phones, pads, desktops and the next release of Xbox.

Image Scaling

Scaling an image in Buildbox™ is very useful, however you should consider the following before using it too much:

Scaling an image down all the time means that your final game will be downloading, storing, and loading into memory, a large image on the device, for no benefit - the extra detail is being thrown away by shrinking the image on screen.

Scaling an image up too much will result in blurry edges which can make your game look bad. This is not always bad though depending on how you are using the image and you may want to consider it for optimising your stored atlas sizes. For example if you are using a 1x1 pixel grey image and scaling it up to completely cover the screen, no blurry edges will be seen.

Atlas / Sprite Sheet

A sprite sheet or atlas is a tool used behind the scenes in computer games to save both run-time memory use on a device, and storage space. An abbreviated, non-technical description follows.

Storage Space

Every PNG file has "header" information about how big the image is and the like, and then the image data itself.   If we assume that every PNG file you have is 2% header information, and we have 25 images, then we are using 25.5 image worth of storage to store those files.

If we take those 25 images and stack them all next to each other (a bit like a photographer's contact-sheet) then we will have a single PNG file with almost no wasted "header" information, saving the half an image's worth of space.

You can manually create sprite sheets using a tool like Photoshop and a lot of manual work, however Buildbox™ does this for you automatically.

Device Memory

Without going into the reason why, any image stored in memory can only put into a box that is a size that is a power of 2 (2,4,8,16,32,64,128,256,512...). Now if we have an image that is 300 pixels wide and 100 high, it would be stored in a box of size 512 pixels wide and 512 pixels high which would mean 88% wasted memory for just that one image.

Start using a lot of those single images and your game may perform terribly, or not run due to lack of memory. Sprite sheets fill the entire storage box in memory, and inside IT you can cram as mans images as possible, resulting in very little wasted memory. Sometimes - with more technical work that we will go into here – none at all.

Color-Picker
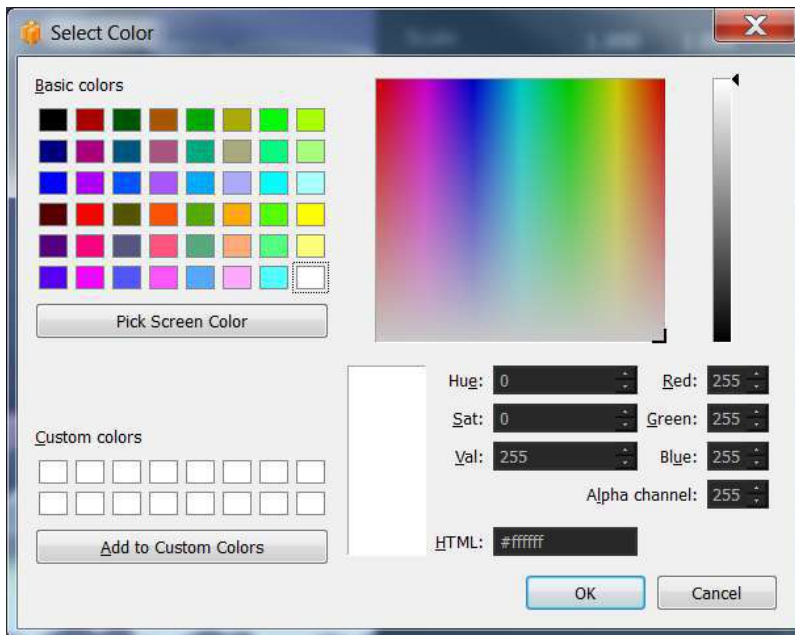
This screen is rather self explanatory. The easiest way to use this screen is to click in the box on the top-right, and then set the color brightness/darkness with the slier on the far right. If you want a transparent color, set the Alpha Channel to a value less than 255.

Animation Editor
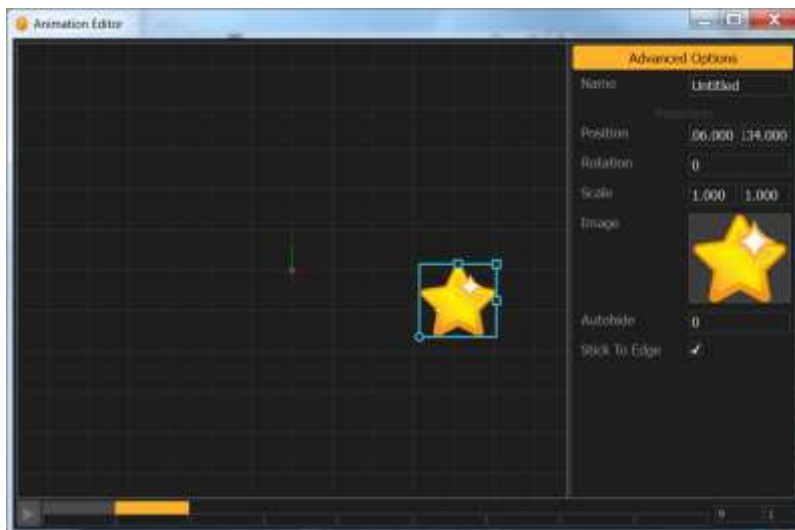
The screen has several components. The main image area, where images can be moved around the sprite pivot point (the dot in the very centre), and the advanced options panel on the right - which are defined per each image you have.

On the bottom you can see the animation panel itself, with these buttons/fields from left to right:

- **Play:** start and stop the animations

- **Frames:** number of animation frames. (the screenshot shows 9) If you have animated sprites that have less frames than this value, they will repeat as required.

- **Animation Speed:** how fast to play animations. A value of 1 - the default - I about 30 frames per second, with values between 1 and 0 being slower, and values above 1 being faster.

The options - per image/image set - are:

- **Name:** for internal reference only.

- **Position:** where the images are in relationship to the centre point. You will see these change as you drag the image around the window.

- **Rotation:** what angle the image is rotated to.

- **Image:** this is a PNG Drop Box. If you want to use an animated image, you can drop several PNG files at once, here.

- **Autohide:** hide after how many seconds.

- **Stick to Edge:** if this object is scaled, having this object ticked will mean the object stays where it is relative to the side, instead of the centre.

Notes:

- moving images around in a multi-framed animation will not allow individual movement per frame. Wherever you move them to, is where they will stay for the entire animation.

- if you edit images on this screen, it will not alter the objects'(/characters'/etc) collision box.

Drag-Drop New Image/Animation
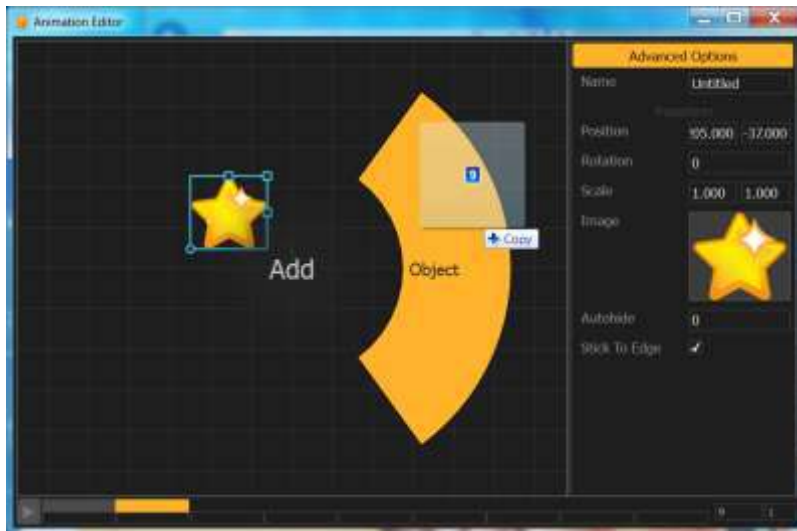
If you want to add another image - possibly animated - then just drag and drop a png file (or collection of png files) onto the main editing area, as you can see, above. When you press the animation play button down the bottom-left, you will see them both animate.

Advanced: Physics Objects
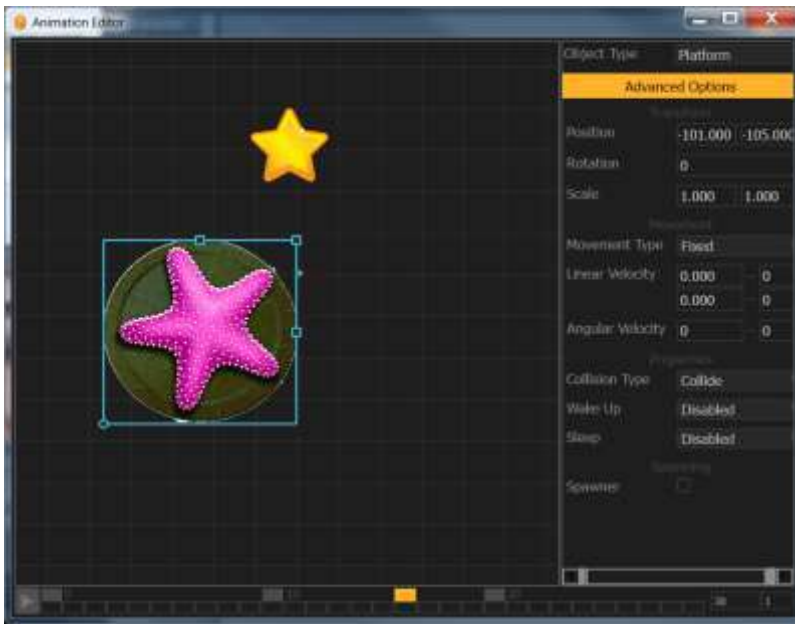
You can drag and drop Objects from the main Buildbox™ window.

This will bring in all the standard object settings, instead of just an animated or fixed image. The screenshot shows an example of an object, and the options available on the right. For details of what you can do you should look at the section on Objects settings.

Here are some ideas however to get you started:

- Character death animation as per the game, Phases: put in several separate "chunks" of the character, and give them physics properties including velocities.

- Objects that spawn: could be interesting weapon effect if added to shooting animation or bullet animation - providing a different look or additional objects that kill a bit like a cluster bomb.

- Exploding objects on jump: A rocket particle-like effect when the character jumps.

- ... your imagination is the best idea source!

Export/Compile Checklist

Do you have:

1. An app icon?
2. A bundle ID for your app store?
3. An app full-name and device-name?
4. Leaderboard ID (if using Game Center)?
5. Google Play ID (if using Google Game Services)?
6. A share message (if using Share feature)?
7. A review Link (if using a Review button)?
8. Ad network settings (if showing ads)?
9. Checked that ad networks selected on screens are available for your selected app store (most ad companies are only available on some platforms)?
10. Do you have a licence key? (for Google Play or similar)