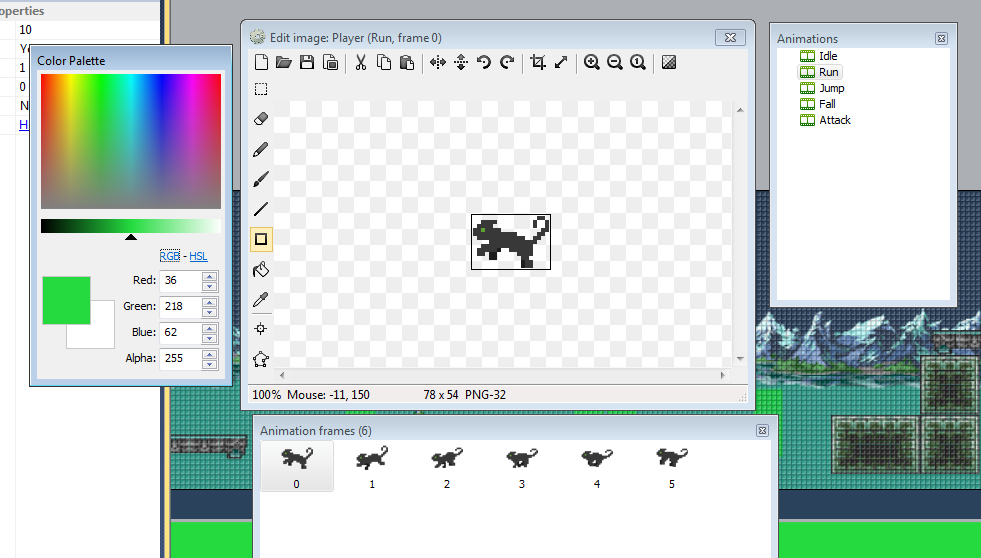
First step of developing this game was to set up some global variables. First was “level”, which is used to track which level the player is currently on.

Before creating the player character, we create a new Sprite and name it PlayerBox. Then create another sprite for the player character.

In this player sprite, import an animation for each type of movement (idle, moving, jumping). This can be done by accessing the Sprite Editing window, and right clicking in the Animations window to Add new animation.

Once here, right click in the animation frames window below, and select 'import sprite strip'. Select your character's sprite strip and chop it up into the appropriate number of horizontal and vertical cells. You can also import animation frames from an animated .gif file if you have one.

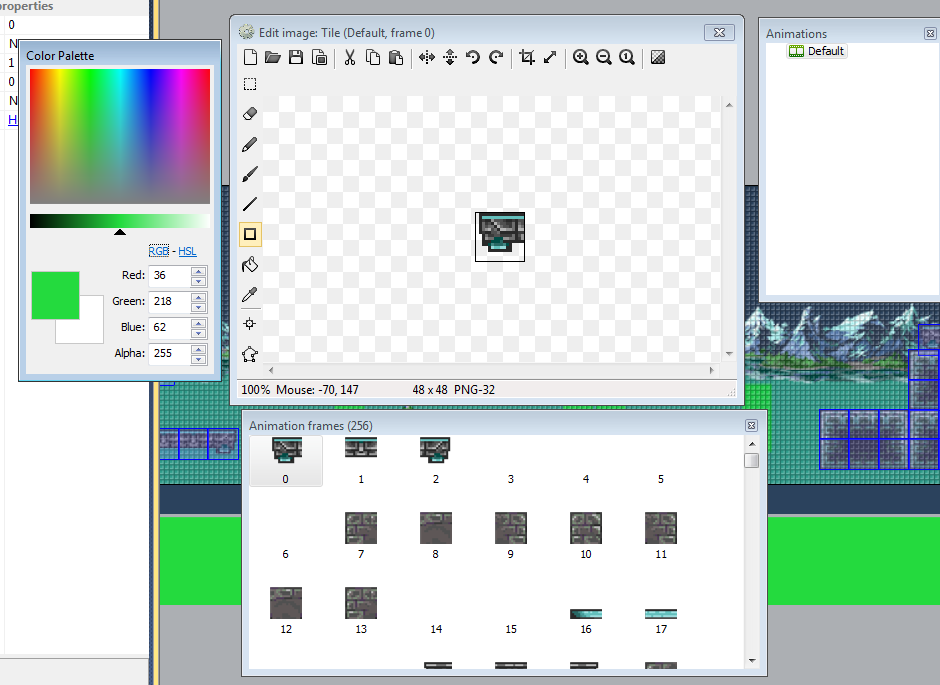
Once the frames are imported, select one and hold Shift while clicking the “Trim edges” button. This will reduce the size of the sprite frames to only the non-transparent content.



Make sure to name the player character Sprite something useful like “Player”, and the PlayerBox to “PlayerBox”.

Create a new sprite named Tiles. Similarly to the Player sprite, you can import a tilesheet using the 'import spritesheet' option and slice it into the appropriate number of pieces.

Select the Tiles you created, and on the left hand side you will find a property for Behavior. Click Add behavior and add the Jump-Thru behaviour. This means that when your player character jumps from below the tile they will jump up through it, but they will not fall through from the top.



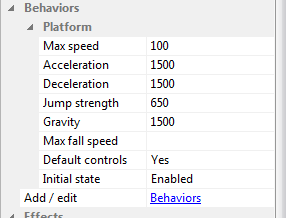
Similarly, add the 'Platform' behaviour to the PlayerBox. This means you can control the player box as a platform character, with jumping and movement.

Resize the PlayerBox to be roughly the same size as the Player sprite.

From here you can create some platforms for the level. You can use the property option on the left with the Tile selected to select which tile you wish to place. The animation frame will match with the animation frame number in the sprite editor. You may also wish to right click on the Tile animation in the animation editing screen to turn off Looping, so the tiles stay at the initial frame you set.

You can Ctrl+drag a tile to create a new one of the same type and use this to construct a basic level.

Give the PlayerBox the Platform behaviour

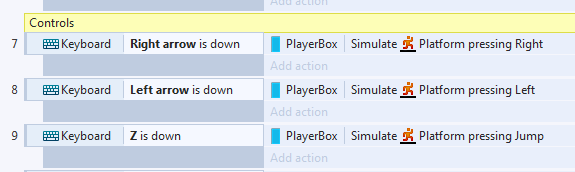


Now, open the Event sheet.

From here we can set up the platforming controls.

Firstly, we need to set some things to happen every tick. Click Add Event, select System and then Every tick. Add an Action to Set Player sprite position to PlayerBox. This will mean that the player sprite follows the PlayerBox, and the PlayerBox can later be used as a collision detector. This is used because the sprite for the Player is irregular and can cause unreliable collision detection.

To set up the controls, create a new Event. Keyboard object>On Right Arrow pressed. Add an action for this event with the PlayerBox object, to Simulate Platform pressing Right. This will make the player move when Right is held on the keyboard.

Create similar events for Left Arrow to move left, and Up Arrow (or any other key of your choosing) to simulate Jump.

To make sure that appropriate animations play when moving, you can set up some events which change animation based on current status.

Create some new events based on the PlayerBox object, Platform on moved, Platform on Stopped, platform on Jump, platform on Fall, platform on Landed and platforkm is Moving.

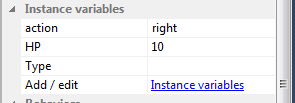
Create some animations in the Player sprite, and name them according to the action they will represent (idle, running, jump, fall etc). For each of the events we set up, create an action which Sets the Player sprite's animation to the appropriate animation for the action.

Now we can create an Enemy sprite. Create a new sprite and import a sprite strip like you did with your Player. You can create new animations in the Enemy Sprite which will correspond to different enemies if you wish. Name each of these animations Enemy1, Enemy2 and so on.

Place the Enemy sprite on a platform, and give it the Platform behaviour.

While the Enemy is selected, in the left properties bar there is an option at add instance variable, click this. Add an instance variable called “action”, make it a string and have it's default value be “left”. This will govern the direction that enemies move in.

Create another instance variable, a number named Type.



Back in the event sheet, create an event On start of layout, with an extra condition For each (Enemy). Add the action to this event as Set animation to “”Enemy” & Enemy.Type”.

This means that whatever value is selected for Type will determine which type of enemy appears in the game for that particular enemy sprite.

Now to set the movement of the enemies.

Create an Event to Compare Instance Variable in Enemy, action. Create one event where Enemy.action is “left”. As actions for this event create 'Simulate Platform pressing left” and Set Mirrored for Enemy. This will make enemies move left when their action is “left”.

Create a similar event for “right”, changing out the Mirrored option to “Not Mirrored” and “right” for movement.

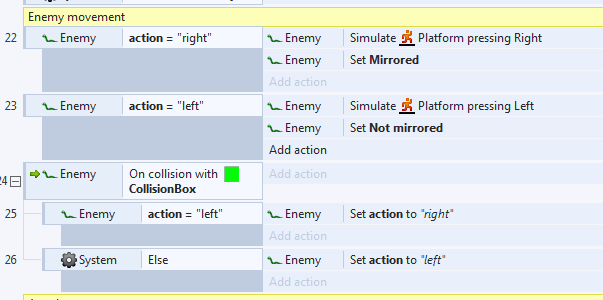
Next, create a sprite which is just a small box of a certain colour, and call it CollisionBox.

In it's properties set it to 'invisible' so that players do not see it while playing the game.

Now create two of these Collisionboxes at the edges of the area you want your enemy to patrol. In your Event sheet create a new event Enemy>On collision with CollisionBox, and a sub event Enemy.action = “left”. Set the action for ths sub event as Set Enemy.action to “right”.

Add a sub event to the “action = left” sub event, and have this condition set to System>Else, and the action for this sub-event be to Set Enemy.action = “left”.

This means that when an enemy collides with the CollisionBox, they will reverse their direction. This will make the enemy patrol left and right between CollisionBoxes.



Click Play to test the game. You should be able to move your Player character with Right, Left and Jump buttons, and the Enemy should move left and right between the CollisionBoxes (which will not be visible while playing).