# Catch the Clown – Using GameMaker Studio 2

Design Documentation & Game Resources

Design Document

Catch the Clown is a little action game. In this game a clown moves around a playing field. The goal of the player is to catch the clown by clicking with the mouse on him. If the player progresses through the game the clown starts moving faster and it becomes more difficult to catch the clown. For each catch the score is raised and the goal is to get the highest possible score. Expected playing time is just a few minutes.
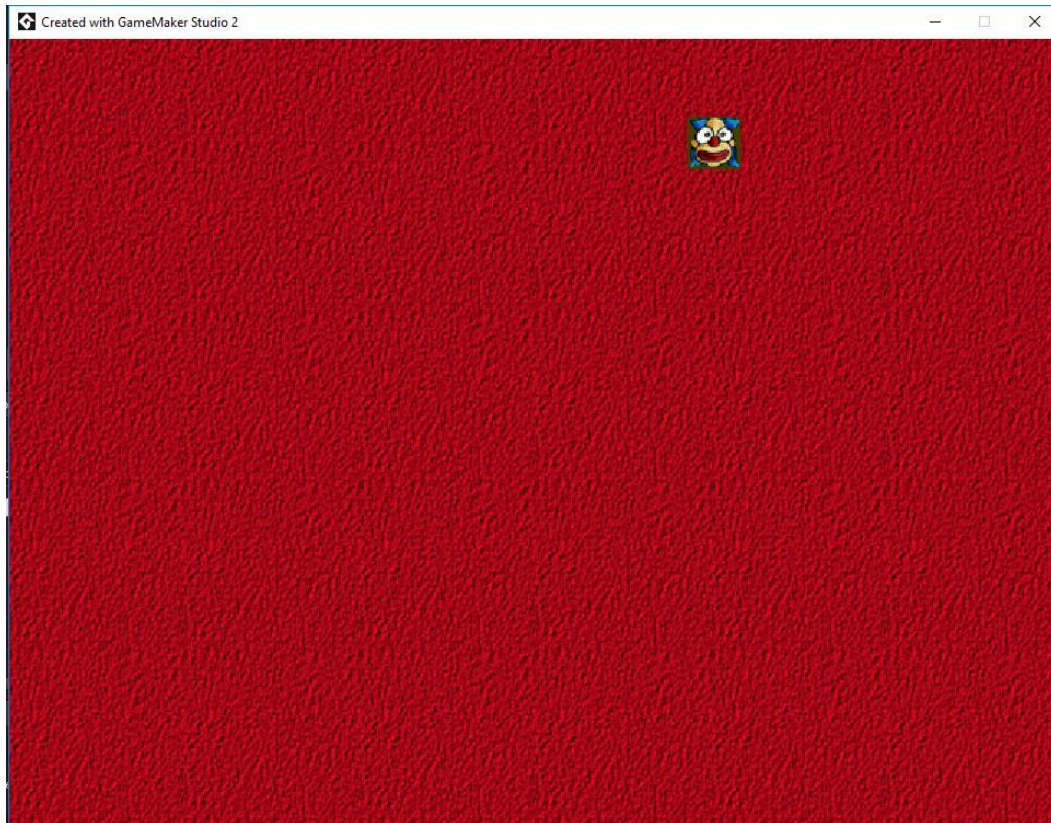
Game Objects

There will be just one game object: the clown. The clown object has the image of a clown face. It moves with a fixed speed. Whenever it falls of the screen, it warps. When the player clicks on the clown with the mouse the score is raised with 10 points. The clown jumps to a random place and the speed is increased with a small amount.

Sounds

We will use two sounds in this game. A bounce sound that is used when the clown warps, and a click sound that is used when the player manages to click with the mouse on the clown.

Controls

The only control the player has is the mouse. Clicking with the left mouse button on the clown will catch it.

Game flow

At the start of the game the score is set to 0. The room with the moving clown is shown. The game immediately begins. When the player presses the <Esc> key the game ends.

Levels

There is just one level. The difficulty of the game increases because the speed of the clown increases after each successful catch.

Game Resources

All game resources are given to you electronically in a Zip Folder. Un-zip this folder to access the resources.

Getting Started

Before starting to make this game, make sure that you have all the sprites and sounds.

Creating Sprites

- You are going to create 2 sprites which will define one image for the **clown** object, and one image for the **background**.
- Click with the right mouse button on the **Sprites** folder on the right of the screen and choose **Create,** then **Sprite**.
- Call the sprite, **spr_clown** and import the clown image.

Add the other sprite and name it **spr_background.**

Creating Objects

Before we can use our sprites in the game, we need to create the game objects.

- Click with the right mouse button on the **Objects** folder on the right of the screen and choose **Create,** then **Object**.
- Call the object, **obj_clown** and choose the clown sprite from the list available.

Setting up The Room

All the action in the game will take place in a **Room**. In GameMaker, different rooms are used for different game screens and levels.
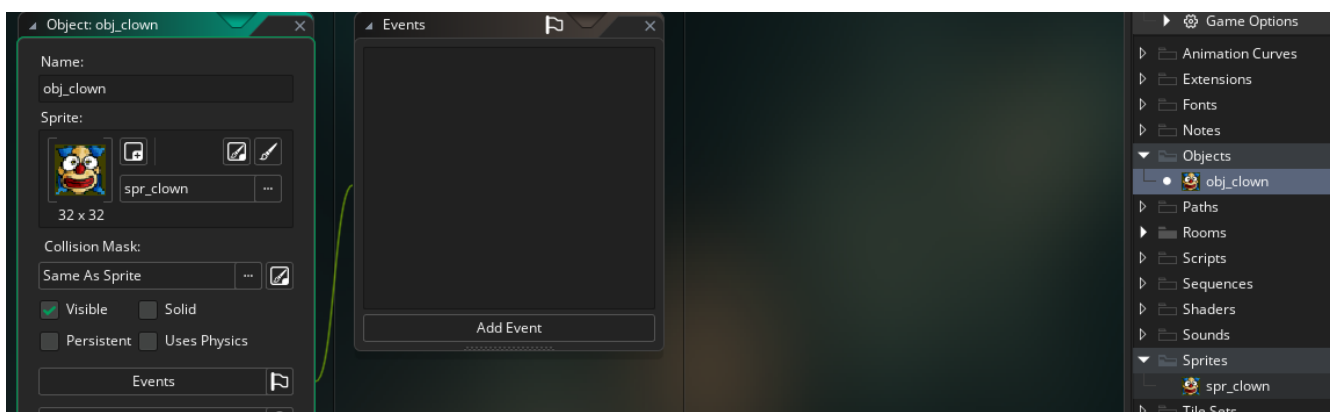
- Click with the left mouse button on the arrow next to the **Rooms** folder, or double click on the **Rooms** folder.
- Open up the **Room** by double clicking on **Room1**.
- In the bottom left, resize the room **Width** to **640**, and resize the room **Height** to **480**.
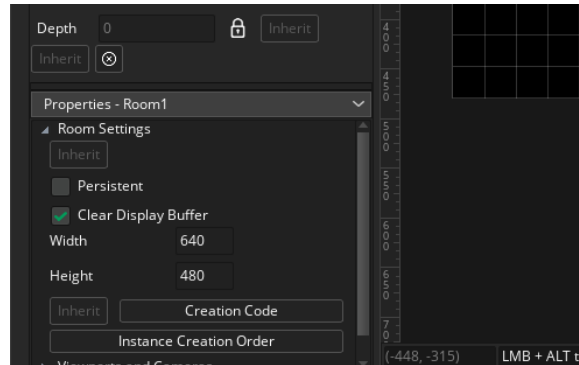
- Scroll down and click **Enable Viewports**. Click **Viewport 0** and set it to **Visible**.
- Set the **Camera Properties Width** to **640**, and the **Height** to **480**.
- Then Set the **Viewport Properties Width** to **960**, and the **Height** to **720**.
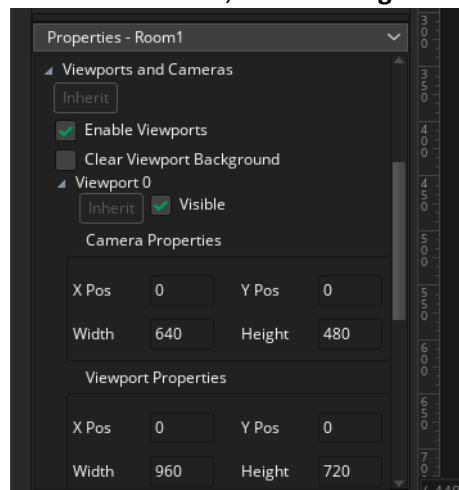

To set the room up, click on the **Instances** layer to the left and then drag in the **obj_clown Object** to place it in the room.

- You also want to add the background image to the room.
- Select the **Backgrounds** layer. And in the properties area, choose the **spr_background** sprite
- Then check the **Horizontal** and **Vertical Tile.**

Making the Clown Move

Actions in GameMaker take place after certain events.

- Double click on the **Clown Object** to open up the clown object window.
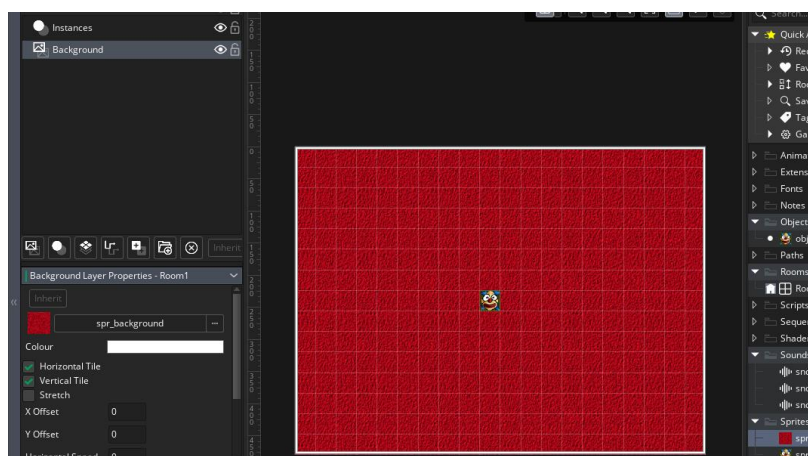
- Click on the **Add Event** button and add a **create event**. When the clown **first** appears on the screen we want to start it moving in a random direction.

- To get a random direction every time, drag the **Randomize** icon in. Then drag in the **Get Random Number** and set the **Type** to **Integer**. **Minimum** to **0**, **Maximum** to **359**, and **Target** to **dir**.

- Drag the **Set Direction Variable** icon into the action window. Then drag the **Set Speed** icon into the action window. Fill the window as shown below.
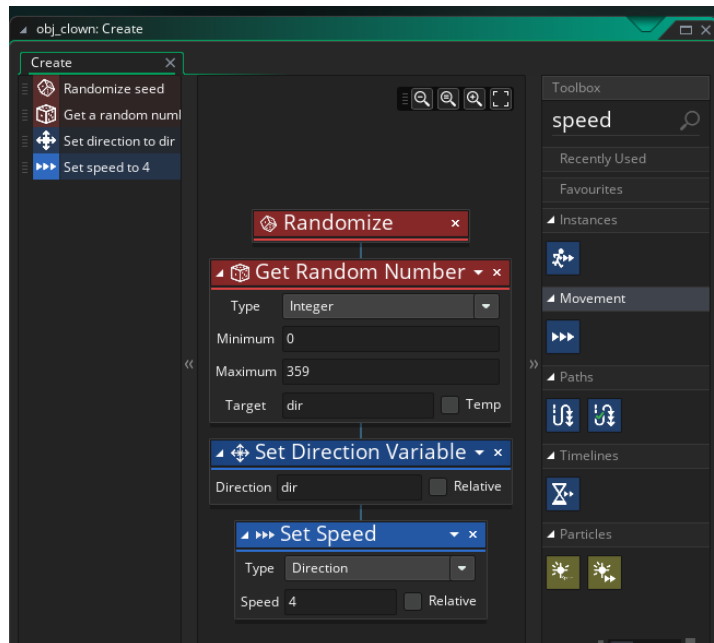
Making the Clown Wrap around the Room

When the clown falls of the screen, we want the clown to wrap to the opposite side of the room.

- Click on the **Add Event** button and add a **step event**.

- Drag in the **Wrap Around Room** icon, and set the margin to 16.

You can now press F5 to test your game so far

We now need to add events to deal with the user clicking on the clown. There are 3 actions to add,

- Add 10 points to the score
- Make the clown jump to a random position
- Make the clown start moving in a new direction and slightly faster.

Increasing the Score

- Open up the clown object window and add a new **event**. Click on **Mouse** and choose **Left pressed**.

- Drag in the **Set Score** icon.

- By clicking the **Relative** box we **add** 10 to the score. Otherwise the score will always be 10.

Moving To a Random Position

- Drag in the **Jump To Point** icon and set **X** to **irandom(576)** and set **Y** to **irandom(416)**.

Changing Direction & Increasing Speed

- Drag in the ⬛ **Get Random Number** and set the **Type** to **Integer**. **Minimum** to **0**, **Maximum** to **359**, and **Target** to **dir**.

- Drag the ✦ **Set Direction Variable** icon into the action window, set the **Target** to **dir**. Then drag the

  ▶▶▶ **Set Speed** icon into the action window. Set the **Speed** to **0.5** and **Relative**.
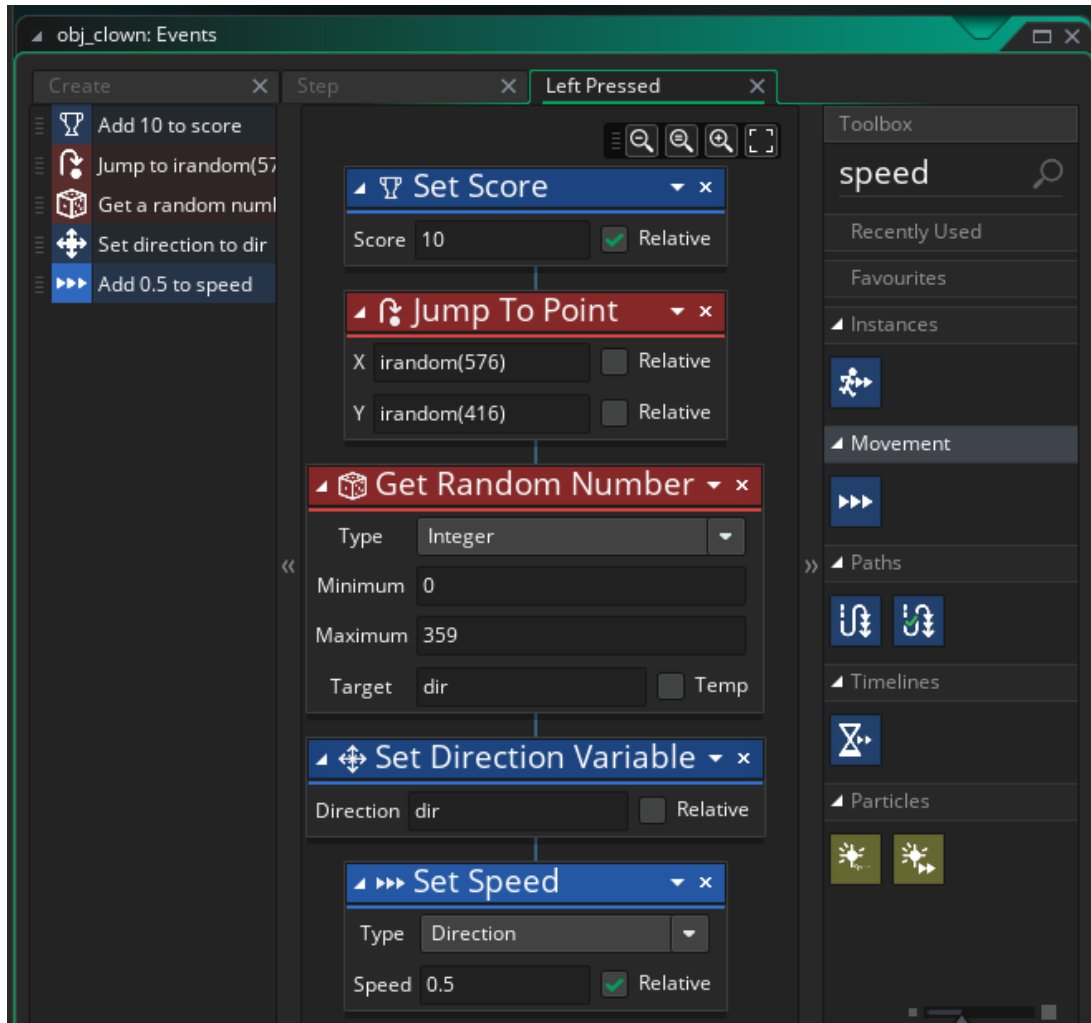
Notice how we are adding a small amount to the speed each time the user hits the clown. This makes the game more difficult the more successful the player. Save your work and test the game at this stage.

Sounds

Most games have sounds included that help make the game more interesting. Just like the sprites, we need to add sounds to the game before we can play them.

We have 3 sounds to add,

- Annoying background music
- A bouncing sound to play when the clown bounces off the walls
- A sound to play when the clown is successfully clicked

To add sounds

- Click with the right mouse button on the **Sounds** folder on the right of the screen and choose **Create,** then **Sound**.
- Call the object, **snd_bgmusic** and load the background music file

Add the other 2 sounds and name them **snd_bounce** and **snd_click**.

Making the Sounds Play

- The events that cause some of these sounds to be played have already been made. All we need to do is drag
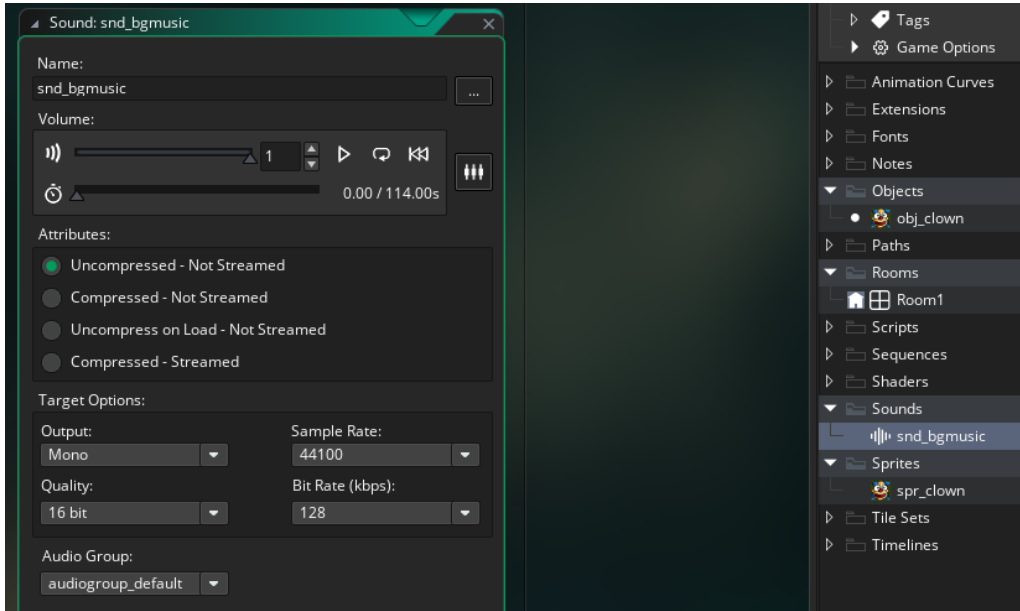
the [icon] **Play Audio** icon into the actions window.

- Start with the background music. We will start playing the music when the clown first appears on the screen. This is handled in the **Create Event** of the clown **object**. – Go to **obj_clown** now.

- Drag in the [icon] **Play Audio** icon underneath the actions that are there. Fill in the details as shown below.

Next we can add the bounce sound when the clown warps.

- Click on the **Add Event** button in the **obj_clown** window and click **Other**, then **Outside Room**.
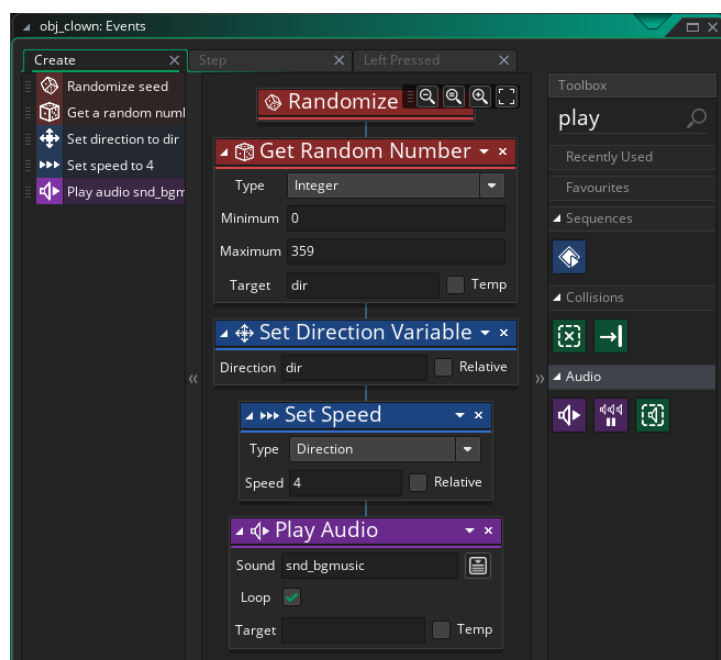
- Drag in the ◀▶ **Play Audio** and select the sound **snd_bounce**. Keep **loop** unchecked.
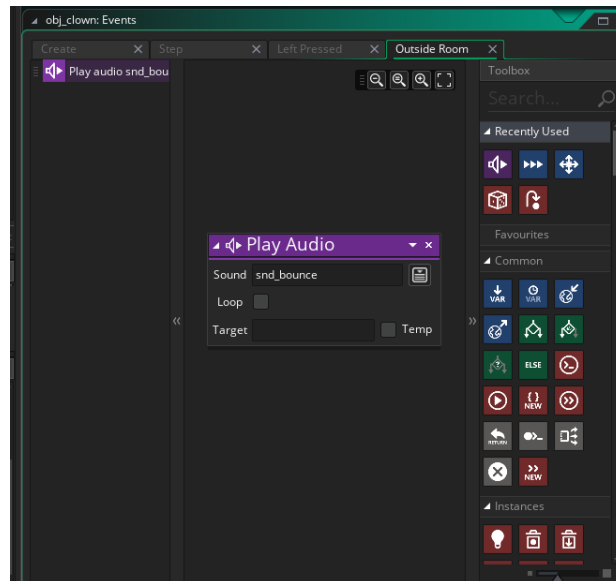
The final sound is to be added to the **Mouse pressed** event that takes place when the player clicks on the clown. Add **snd_click** in the same way as you did for the bounce sound.

Save and test your Game (F5). Your Game is finished, now if you want to think about changing or adding to your game, read on below.

Extension Exercises

Our game doesn't have much to it yet. This is an advantage for us at this stage. We can experiment by changing the sprites and sounds.

Sprites - Editing The Clown

The clown sprite is a picture that is 32 pixels wide and 32 pixels tall. You could replace it with a similar sized picture or design your own.

You can use most image editing packages to make sprites. Paint is more than good enough for the purpose. (Search Paint in the start menu)

Draw a sprite around the same size and try that out in your game. Remember that changing the size of the clown will make it easier or harder to click on.

To set Paint up to draw a 32x32 graphic, click on Image, Attributes on the main menu and fill in the boxes appropriately.

Sounds

There are only 3 sounds. But you could record some sounds using Audacity or other suitable software. How about if the clown said 'Ouch' in your voice when it was clicked? Or you can find free sounds online.

All the events and actions that we created are called the **Game Rules**. Our game has a number of variables. These are values that we can change to make our game play very differently.

Catch the Clown Variables

Clown Starting Position

Currently the game starts with the clown positioned wherever we added it to the room. The starting position could be changed to the centre of the screen or a random position.

The GameMaker **Jump To Point** icon could be added to the clown's **Create** event to change this.

Clown Speed

The clown **starts moving** at a speed of **4**. This number can be changed to make the game start off faster or slower.

The clown speeds up by 0.5 every time he is **clicked**. Change this number to make the game easier or harder.

Score

You score 10 points when the clown is clicked. This figure can be changed although that won't make a huge difference to the game. Perhaps there could be ways to lose points.

Our game is quite simple. There are not many variables for us to change. If we want to add any more interest to the game, we will have to add new objects.

There are a few ideas on this page, but that is no substitute for experimenting with your game to see what you can do.

Add More Objects

You could add a new object. This object will move around randomly just like the clown. There could be several of them. When you click on this object you lose some points.

Add New Game Rules

You could make the game work by giving the player a reason to try to hit the clown straight away. If the player gets more points for hitting the clown sooner.

The following steps would need to be followed,

- Clown Create Event - Set a variable called points to the value 100.
- Clown Step Event - Set the points variable to relative -1.
- Clown Left Pressed Event - Instead of increasing the score by 10, increase it by the value stored as points. You will also need to set points back to 100.

If you try this out, you should see the scoring change quite dramatically.

Over the coming months take a look at some of the games that have been made with GameMaker Studio 2 using this link: Made With GameMaker Showcase (yoyogames.com)