Sign in here!

fsu.devlup.org/signin

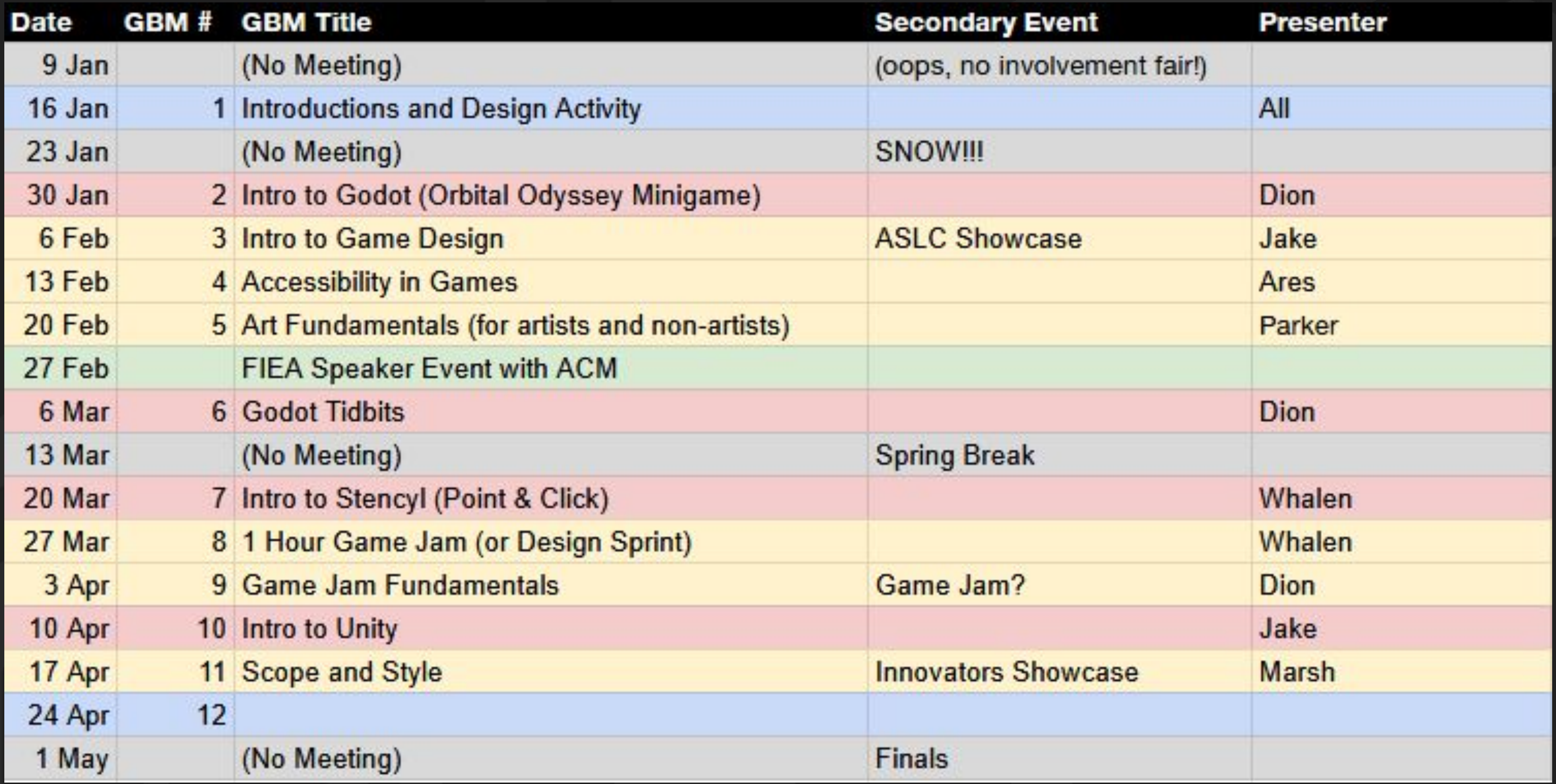
DevLUp FSU

GBM #7

Intro to Stencyl

March 20th, 2025

Meeting Schedule

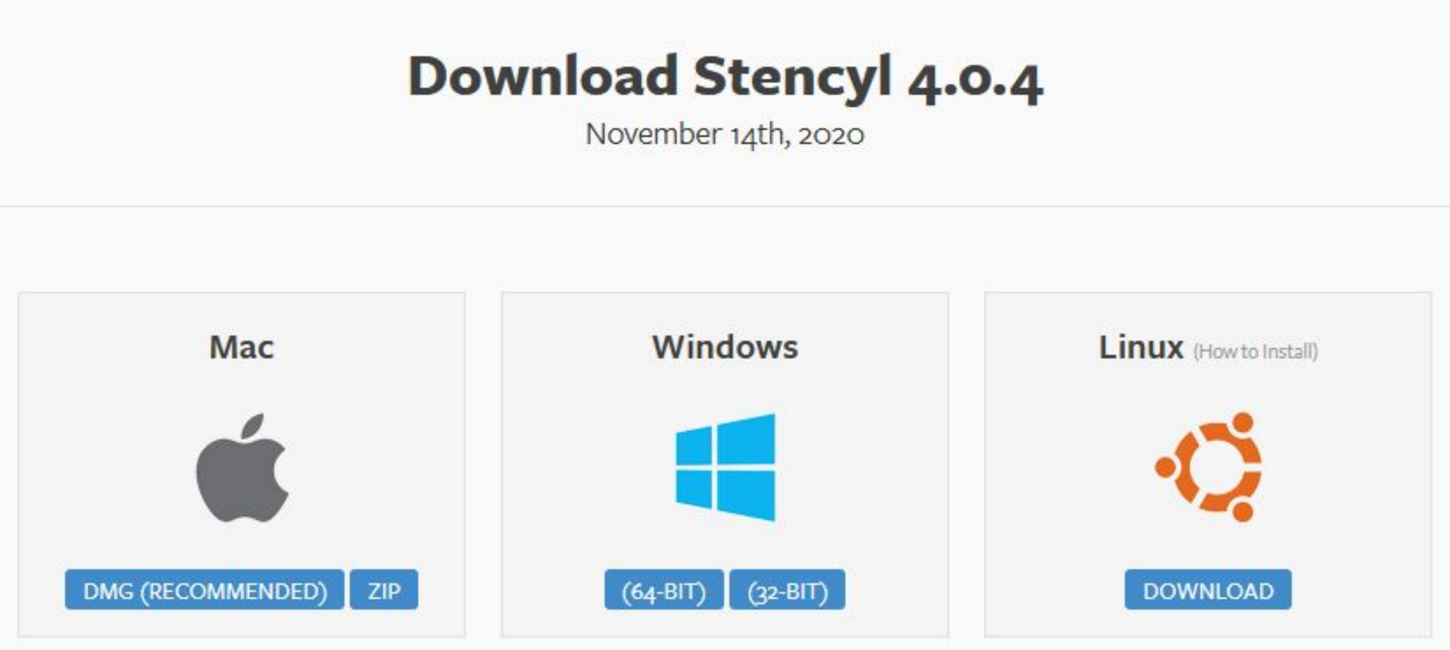


Date	GBM #	GBM Title	Secondary Event	Presenter
9 Jan		(No Meeting)	(oops, no involvement fair!)	
16 Jan	1	Introductions and Design Activity		All
23 Jan		(No Meeting)	SNOW!!!	
30 Jan	2	Intro to Godot (Orbital Odyssey Minigame)		Dion
6 Feb	3	Intro to Game Design	ASLC Showcase	Jake
13 Feb	4	Accessibility in Games		Ares
20 Feb	5	Art Fundamentals (for artists and non-artists)		Parker
27 Feb		FIEA Speaker Event with ACM		
6 Mar	6	Godot Tidbits		Dion
13 Mar		(No Meeting)	Spring Break	
20 Mar	7	Intro to Stencyl (Point & Click)		Whalen
27 Mar	8	1 Hour Game Jam (or Design Sprint)		Whalen
3 Apr	9	Game Jam Fundamentals	Game Jam?	Dion
10 Apr	10	Intro to Unity		Jake
17 Apr	11	Scope and Style	Innovators Showcase	Marsh
24 Apr	12			
1 May		(No Meeting)	Finals	

#👀showoff

Download Stencyl

stencyl.com/download -- it'll bug you for an account, but you don't need one today

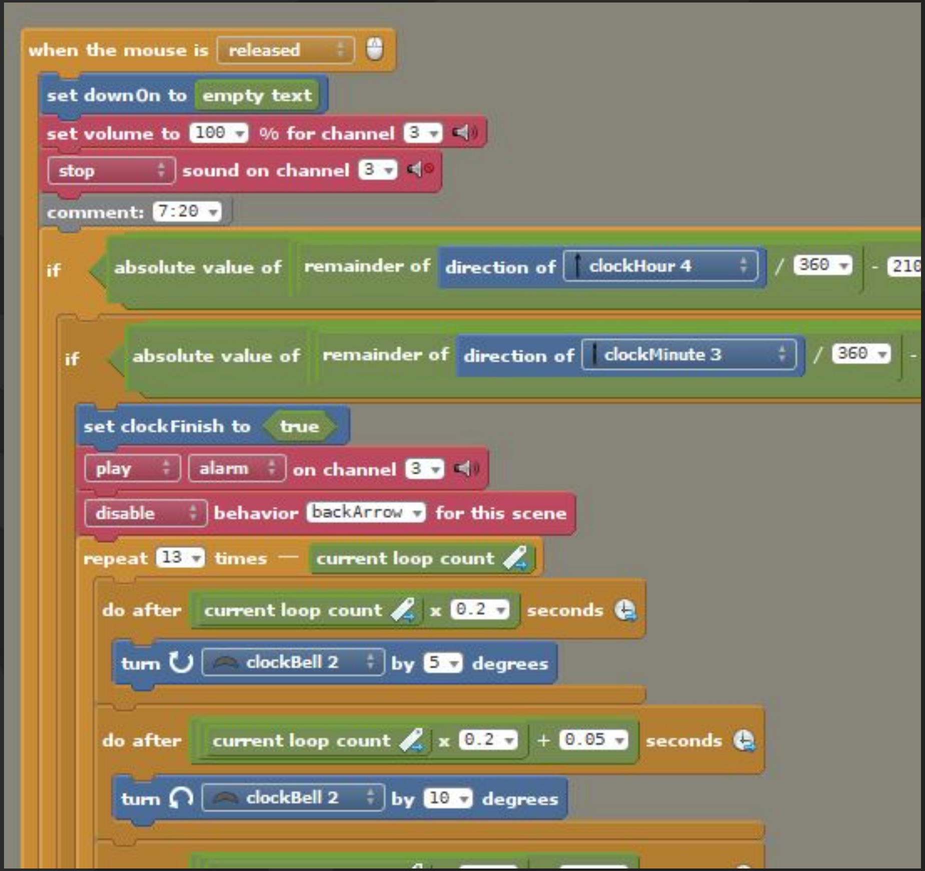


Download Stencyl 4.0.4

November 14th, 2020

What is Stencyl?

- Block-based game development engine
- Like Scratch or Code.org
- More railroaded--less freedom, but you’ll get where you want to go faster



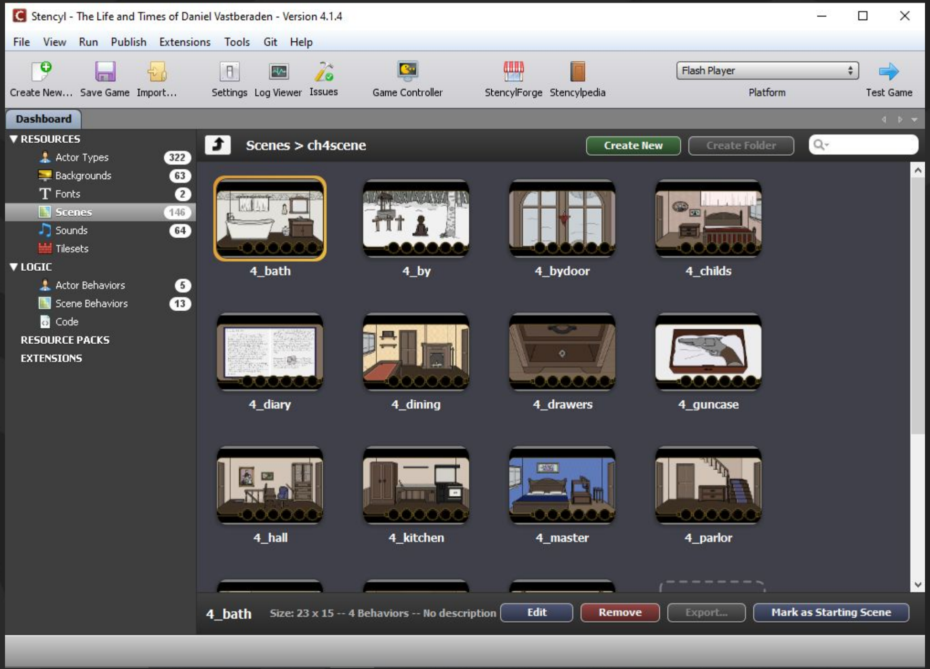
```
when the mouse is released
  set downOn to empty text
  set volume to 100 % for channel 3
  stop sound on channel 3
  comment: 7:20
  if absolute value of remainder of direction of clockHour 4 / 360 - 21
    if absolute value of remainder of direction of clockMinute 3 / 360 -
      set clockFinish to true
      play alarm on channel 3
      disable behavior backArrow for this scene
      repeat 13 times — current loop count
        do after current loop count x 0.2 seconds
          turn clockBell 2 by 5 degrees
        do after current loop count x 0.2 + 0.05 seconds
          turn clockBell 2 by 10 degrees
```

Why use Stencyl?

- Easier to think visually
- Engine UI distinguishes object types
- Less stuff to worry about for a basic game

Personally,

- Regions and mouse controls are easily definable
- Animations, scales, etc. made more sense for nebulous reasons
- I got to Steam with it, so it's not just for babies



Today, we're recreating a room from MOTAS

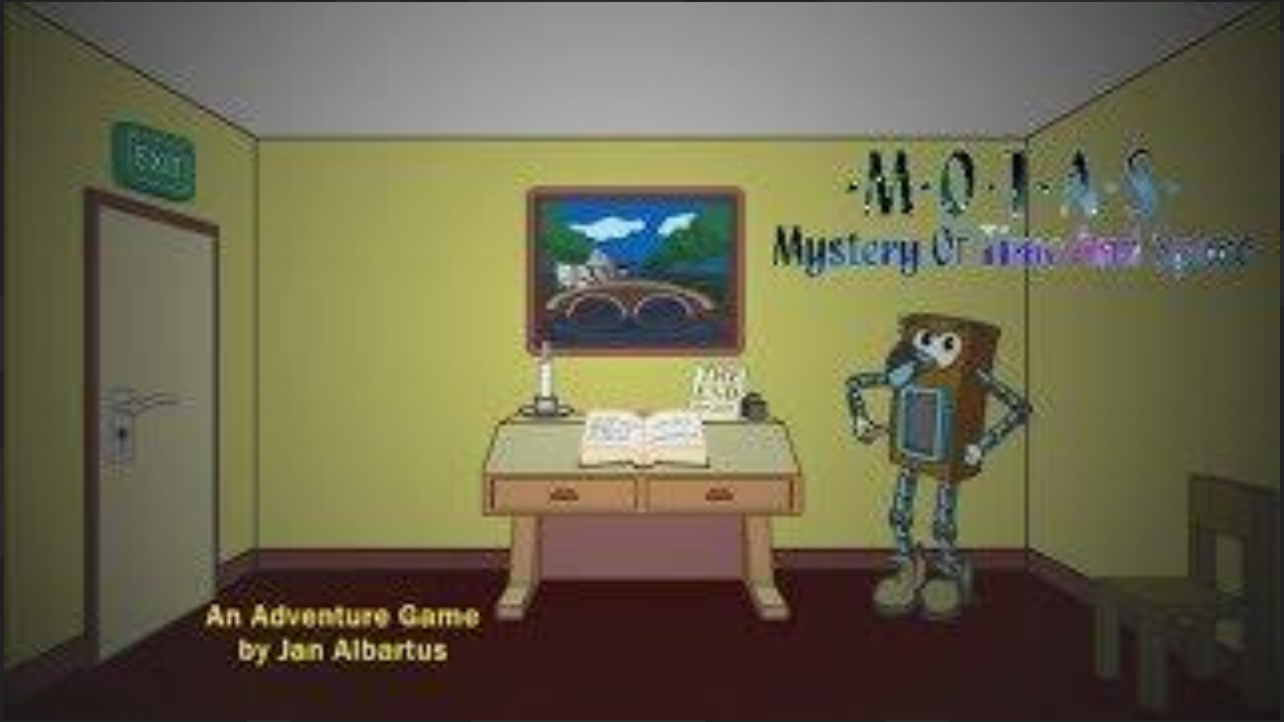
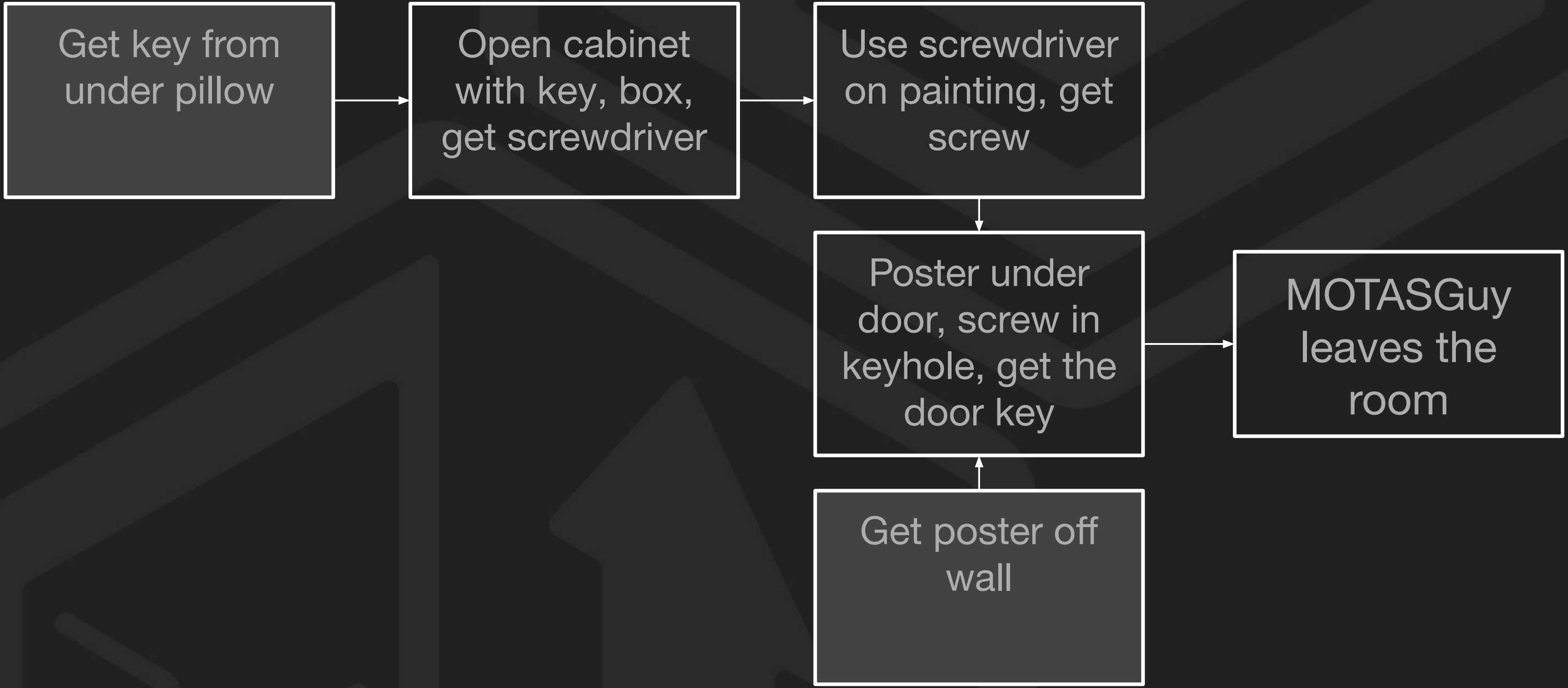


Fig 4.1. If the embed doesn't work, stop the video before he enters the next room. The suspense!

First, we plan:



Get key from under pillow

Open cabinet with key, box, get screwdriver

Use screwdriver on painting, get screw

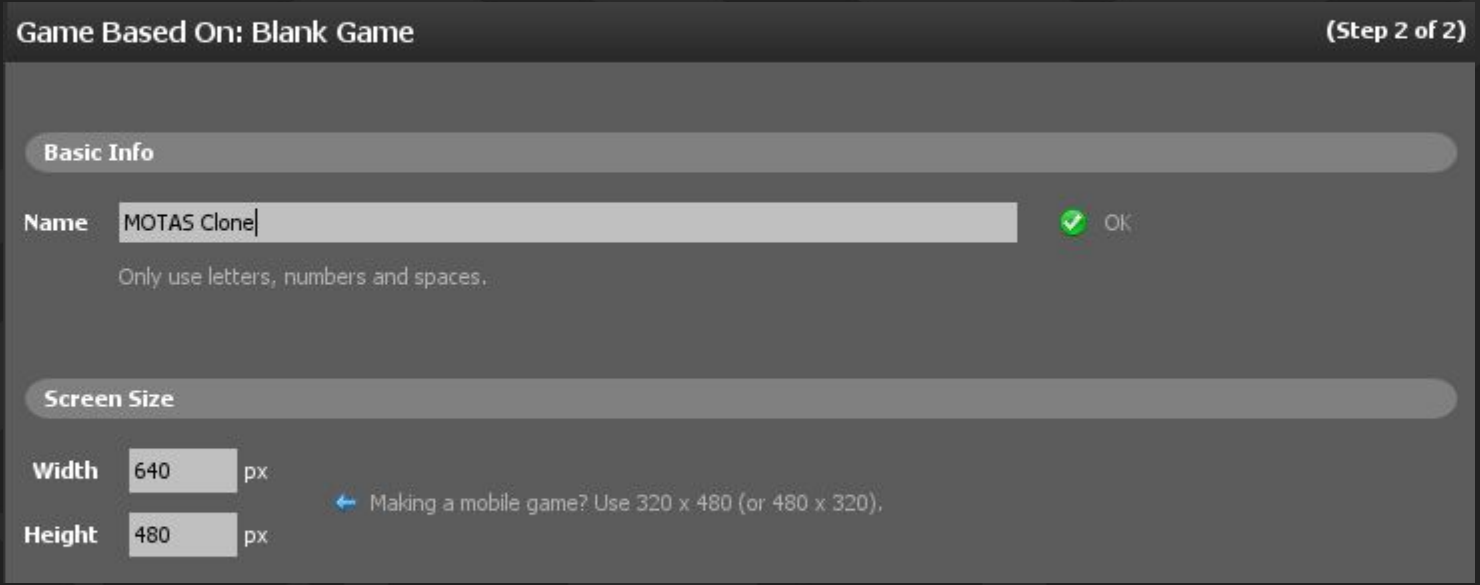
Poster under door, screw in keyhole, get the door key

Get poster off wall

MOTASGuy leaves the room

Make the game file

- Create a New Game…
- Title it whatever you want



Game Based On: Blank Game

(Step 2 of 2)

Basic Info

Name: MOTAS Clone

✓ OK

Only use letters, numbers and spaces.

Screen Size

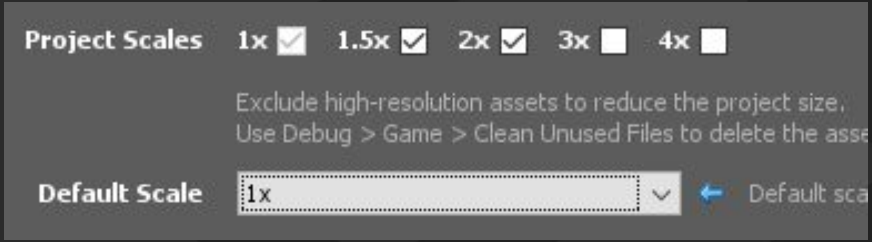
Width: 640 px

Height: 480 px

← Making a mobile game? Use 320 x 480 (or 480 x 320).

Settings for Point’n’Click

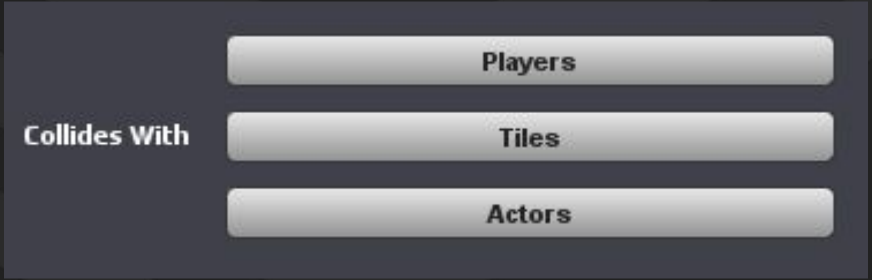
- Settings -> Advanced -> Project scales -> Uncheck 4x and 3x
- Groups -> Uncheck all ‘collides with’ boxes
- Groups -> Create new -> name ‘Items’

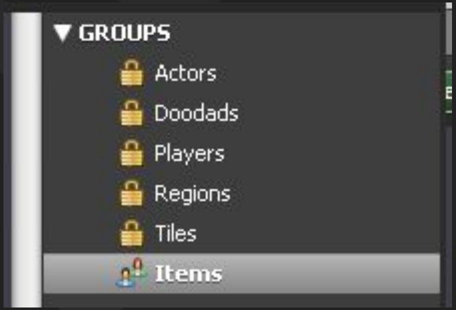


Project Scales 1x ☑ 1.5x ☑ 2x ☑ 3x □ 4x □

Exclude high-resolution assets to reduce the project size.
Use Debug > Game > Clean Unused Files to delete the asse

Default Scale 1x

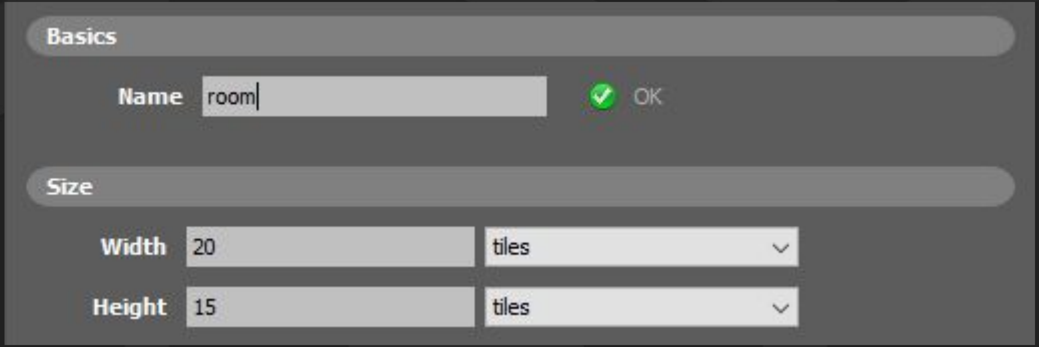




Scenes

Where the ‘game’ of the game happens

- Scenes
- Create New…
- Name it ‘room’



Name: room

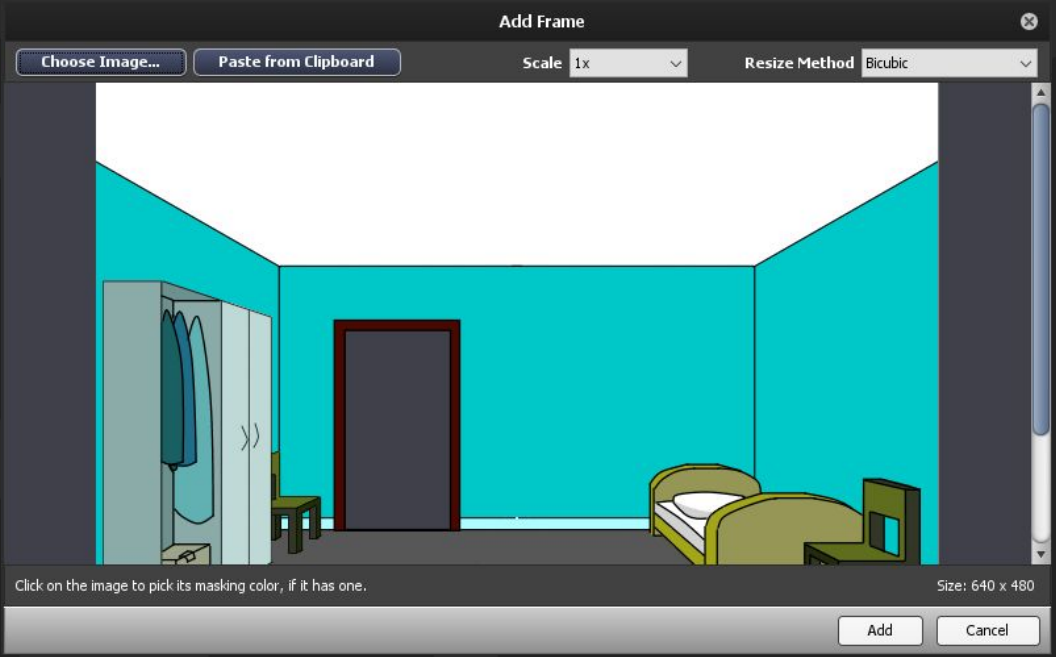
Width: 20 tiles

Height: 15 tiles

Backgrounds

Static images that usually fill the screen

- Backgrounds
- Create New…
- Name it ‘background’
- Add a frame…
- Select ‘background.png’



Scale 1x

Resize Method Bicubic

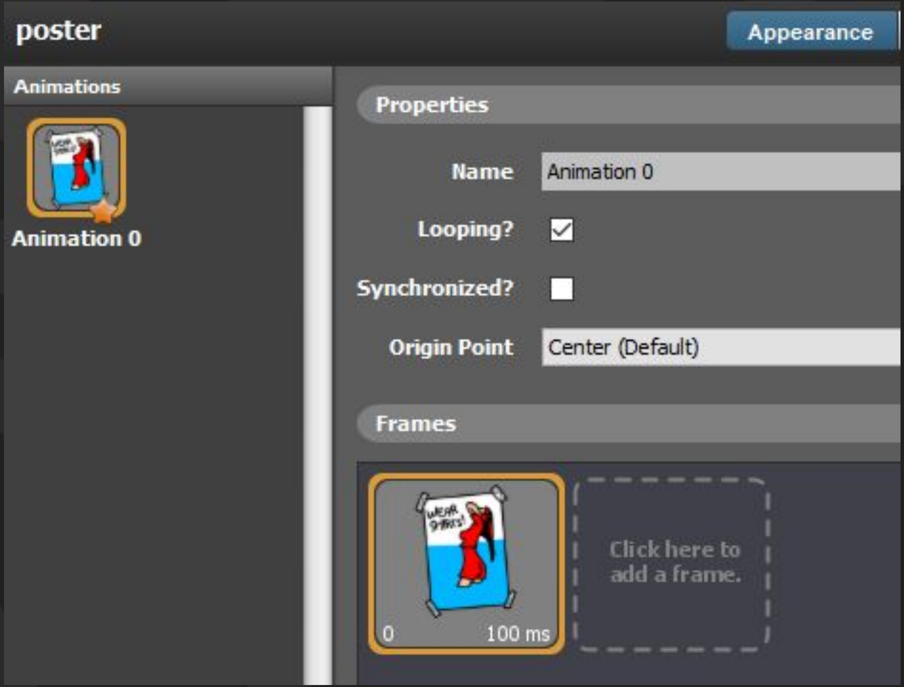
Click on the image to pick its masking color, if it has one.

Size: 640 x 480

Actors

Things that move (and you can click on)

- Actors
- Create New…
- Name it ‘poster’
- Add a frame…
- Select ‘poster.png’
- Repeat for posterground.png, cabinetdoor-L.png, cabinetdoor-R.png exitdoor.png, and painting.png



poster

Name: Animation 0

Looping? ☑

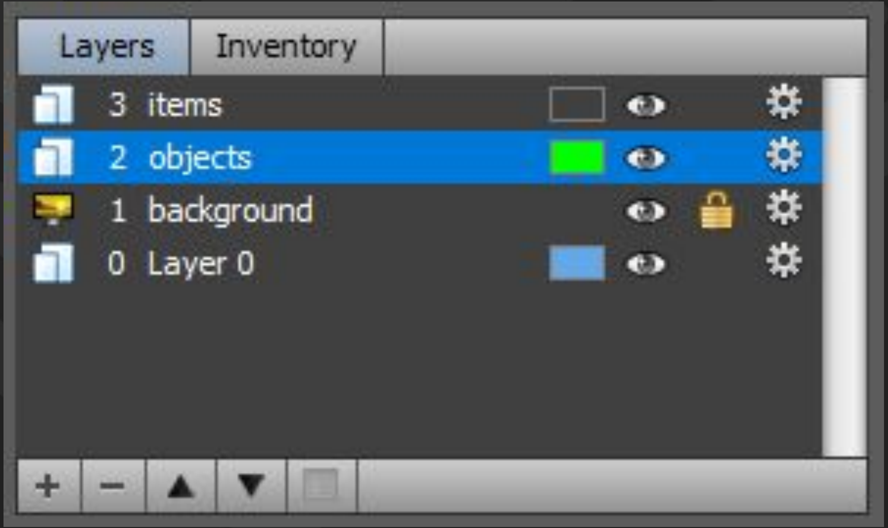
Synchronized? □

Origin Point: Center (Default)

Build the Scene

See ‘layers’ in the corner? That’s the magic.

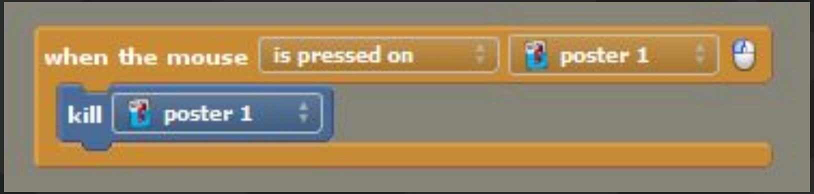
- Add ‘background’ as new background layer
- Add ‘objects’ as new tile layer
- Add ‘items’ as new tile layer
- Select the ‘objects’ layer
- Go to ‘Palette’ -> ‘Actors’
- Select each actor from earlier and place them accordingly



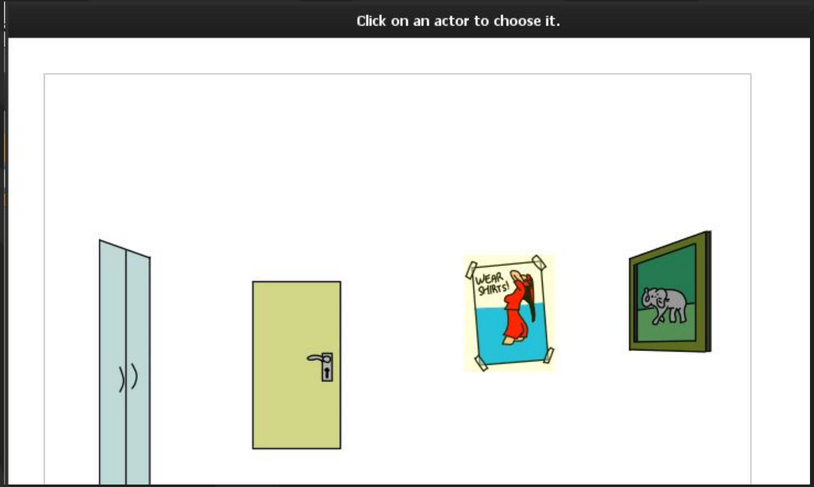
Let’s click on stuff!

Finally, some code. I thought this was art class for a minute.

- Go to ‘Events’ in ‘room’
- Add event -> Input -> On actor
- Change ‘entered’ to ‘is pressed on’
- Change ‘actor’ to ‘choose actor’ -> click on the poster
- Add block Actors -> Properties -> Kill actor -> click on the poster again



```
when the mouse is pressed on poster 1
  kill poster 1
```

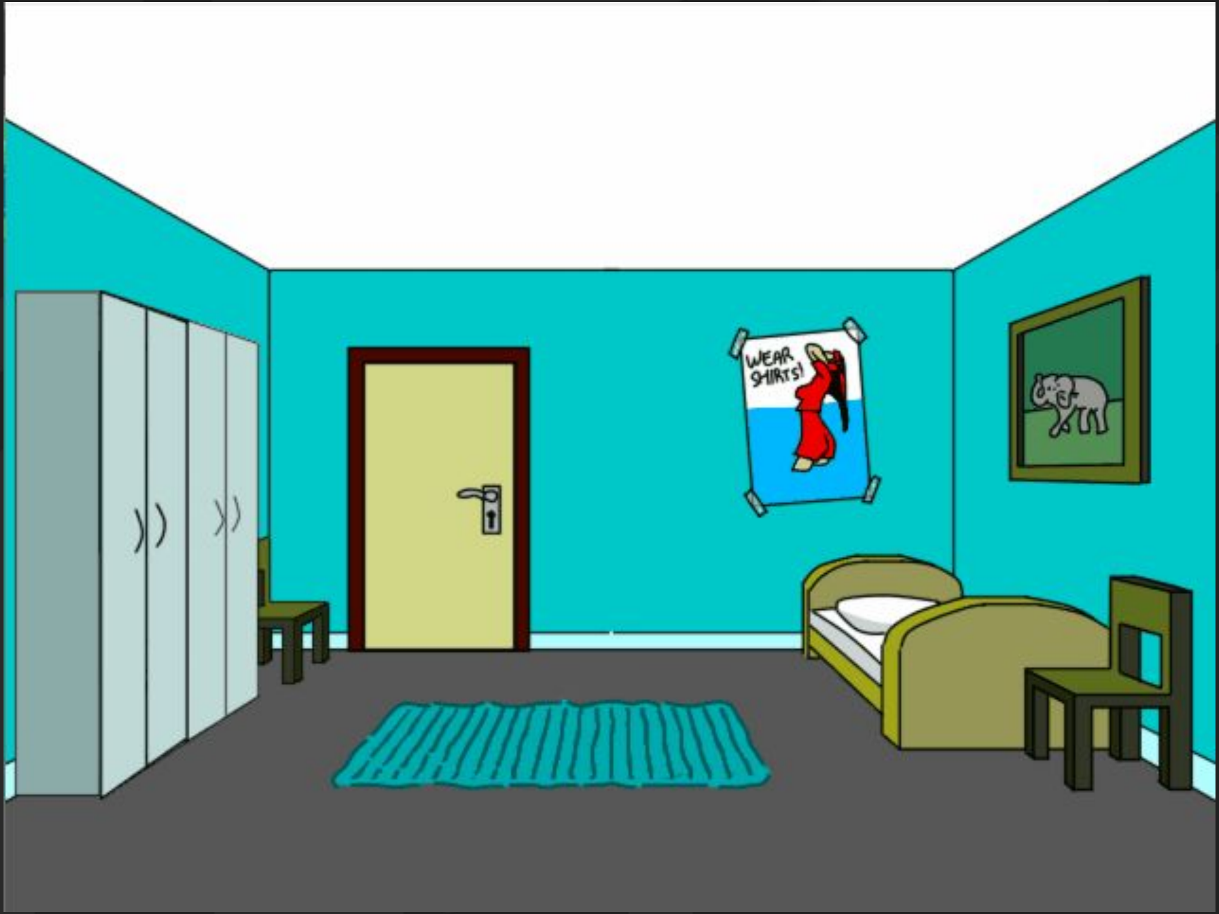


Click on an actor to choose it.

Run the game

(Click on the poster and it vanishes!)

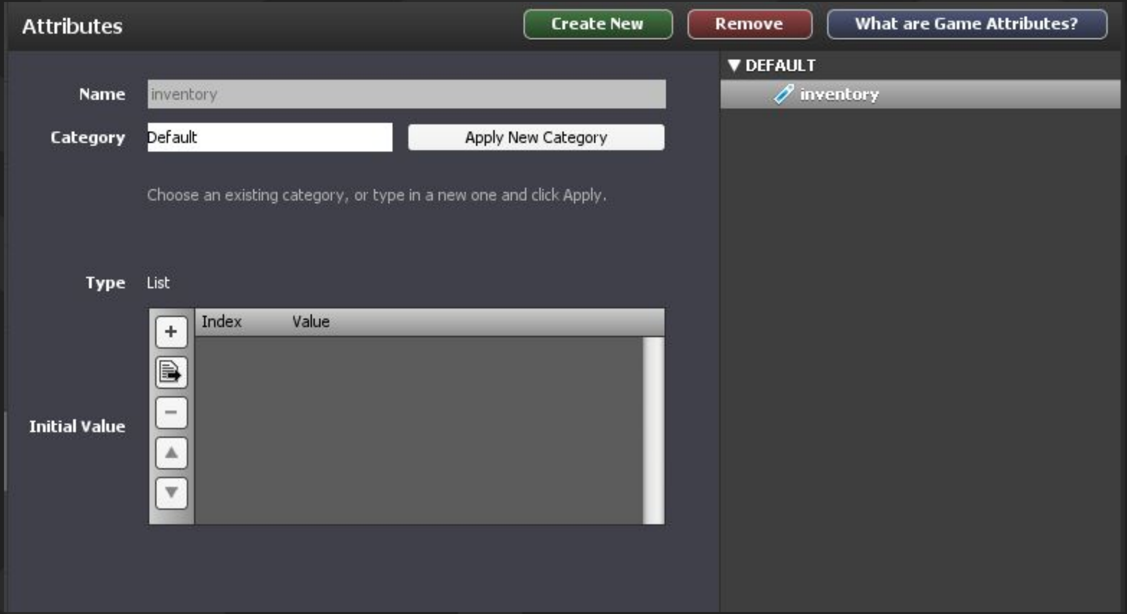
(But we didn't pick it up yet.)



Game Attributes

Basically global variables.

- Add block Game Attributes -> Create new game attribute
- Name it ‘inventory’ and make it a list



Name inventory

Category Default

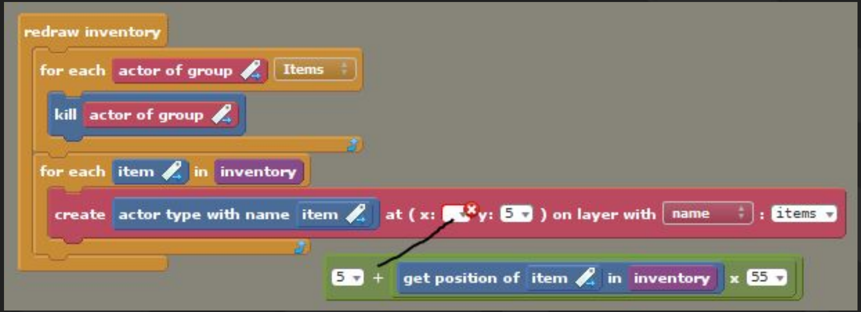
Choose an existing category, or type in a new one and click Apply.

Type List

Initial Value

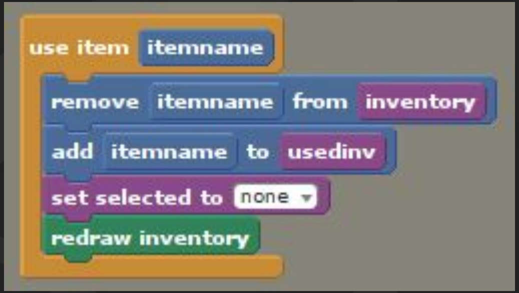
Custom Blocks

Global functions! Yippee!

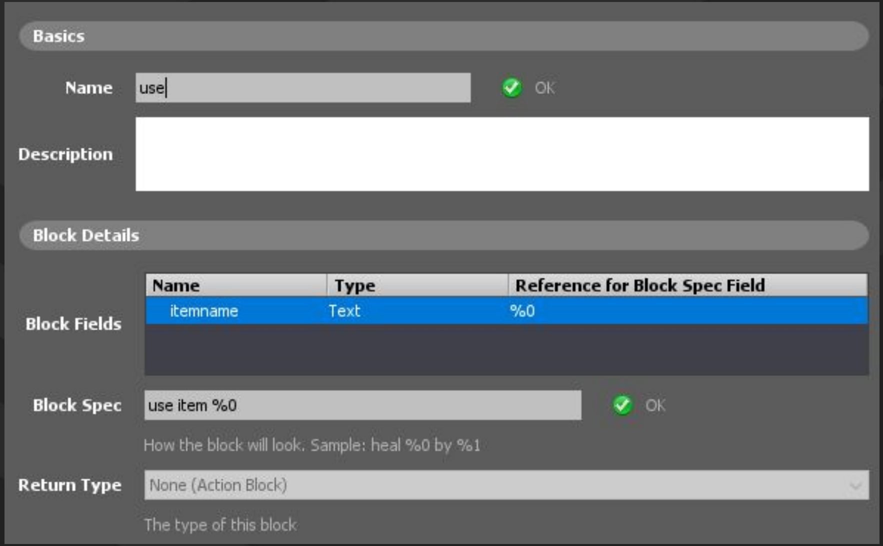


```
redraw inventory
    for each actor of group  Items
        kill actor of group
    for each item in inventory
        create actor type with name item at ( x:  y: 5 ) on layer with name : items

5 + get position of item in inventory x 55
```



```
use item itemname
    remove itemname from inventory
    add itemname to usedinv
    set selected to none
    redraw inventory
```



Basics

Name: use ✓ OK

Description:

Block Details

Block Fields:

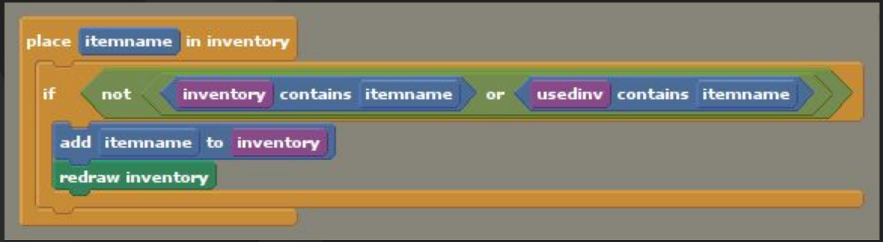
Name	Type	Reference for Block Spec Field
itemname	Text	%0

Block Spec: use item %0 ✓ OK

How the block will look. Sample: heal %0 by %1

Return Type: None (Action Block)

The type of this block

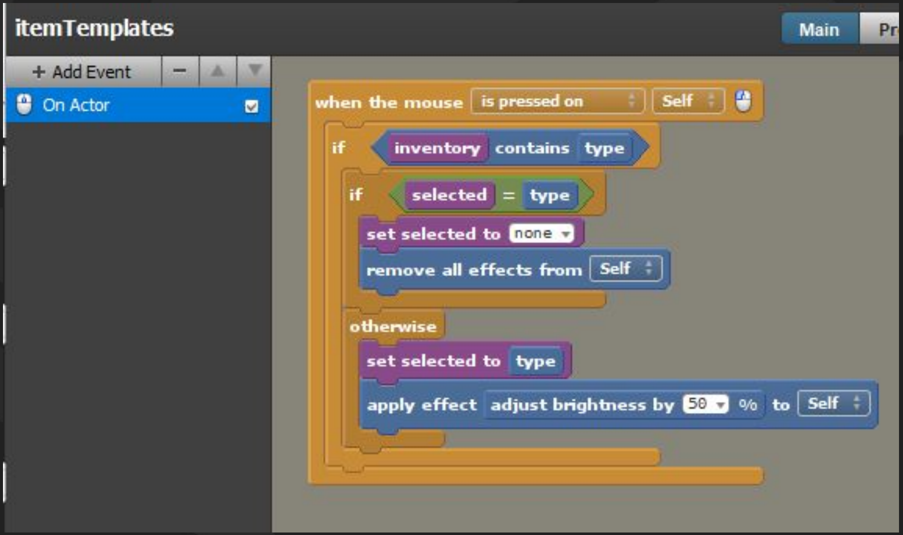


```
place itemname in inventory
    if not < inventory contains itemname or usedinv contains itemname >
        add itemname to inventory
        redraw inventory
```

Behaviors

Also kinda global functions, but you attach these to actors and scenes specifically.

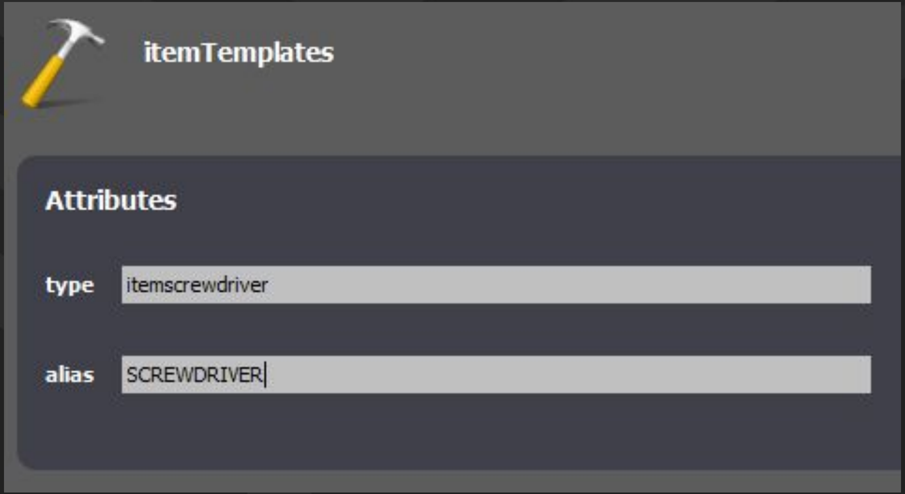
- Actor Behaviors
- Create new - ‘itemTemplates’
- More selection code yay
- Create two attributes - ‘type’ and ‘alias’
- We’ll only use ‘type’ today -- ‘alias’ is for later if we want to write the items’ names on screen



```
when the mouse is pressed on Self
  if inventory contains type
    if selected = type
      set selected to none
      remove all effects from Self
    otherwise
      set selected to type
      apply effect adjust brightness by 50 % to Self
```

Actors, p.2

- Make an actor named ‘item-poster’
- Give it the frame ‘item-poster.png’
- Go to Behaviors and attach itemTemplates
- Make its type ‘item-poster’ and its alias ‘POSTER’
- Go to Properties and put it in the Items group
- Repeat for item-cabinetkey, item-screwdriver, item-screw, and item-exitkey.



itemTemplates

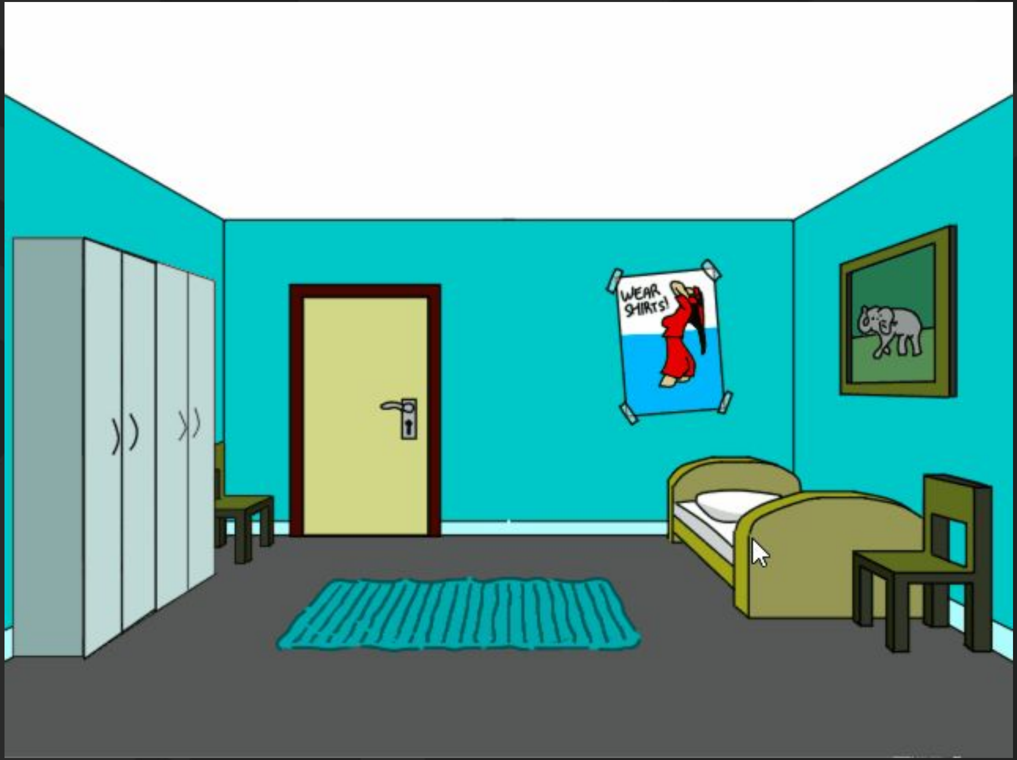
Attributes

type: itemscrewdriver

alias: SCREWDRIVER

Update that poster code

- Add custom block -> place ‘poster’ in inventory
- Run the game! It definitely works!
- (editor’s note: make sure it works)

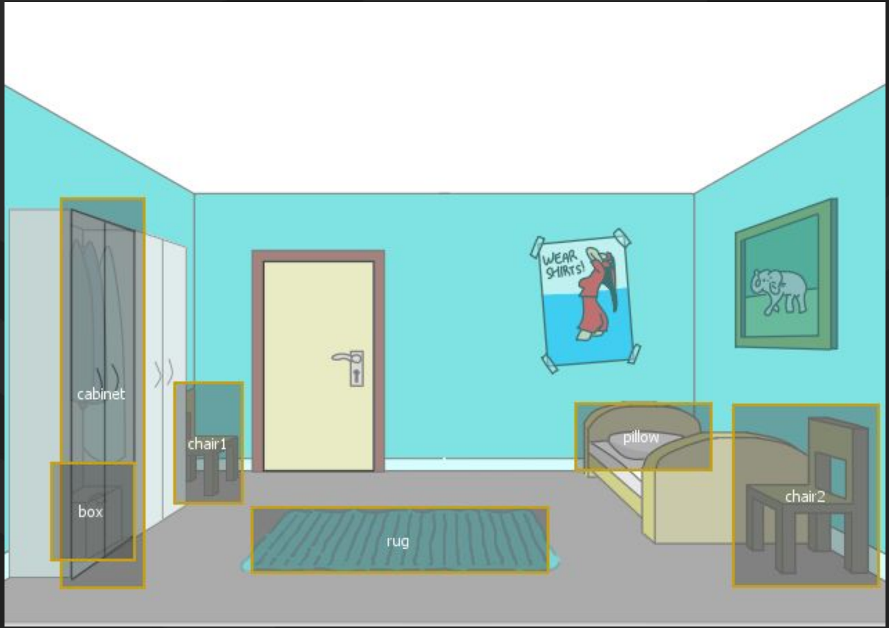


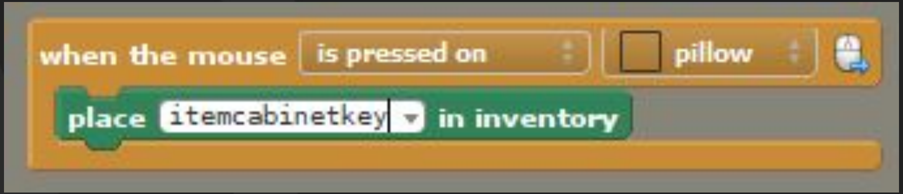
Regions

Stuff you can click on that's not an actor

- Go to Scene in room
- Click that blue rectangle with the plus
- Drag to make a rectangle on your screen--that's a region!
- Put regions on the pillow, rug, box, cabinet doors, and both chairs

Now we can go back and copy the poster code for the pillow, making it give us the key



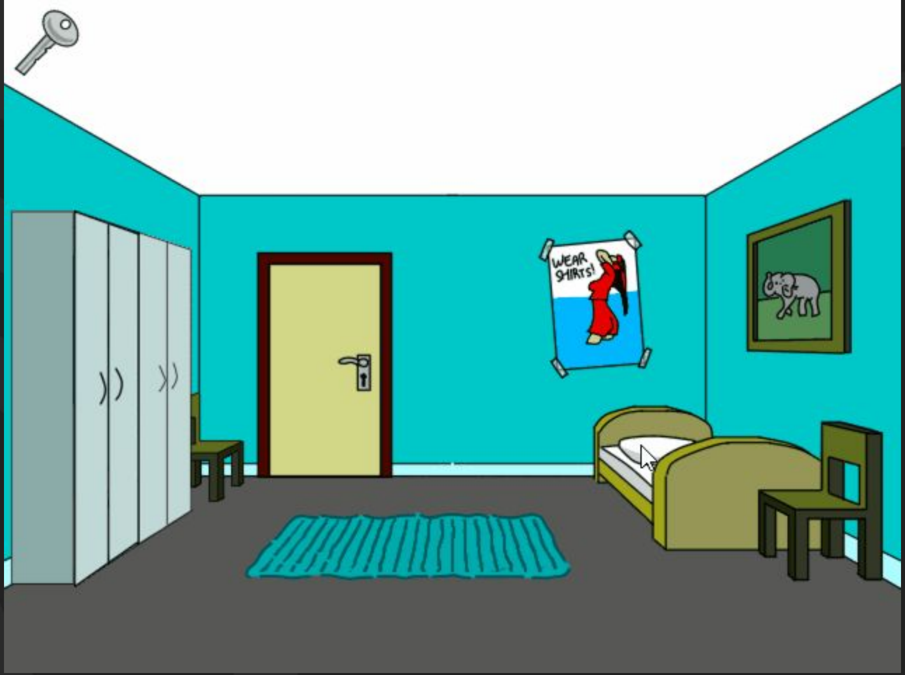


```
when the mouse [is pressed on] [pillow]
    place [itemcabinetkey] in inventory
```

Tweening

Slide stuff around on the screen -- animation on the cheap

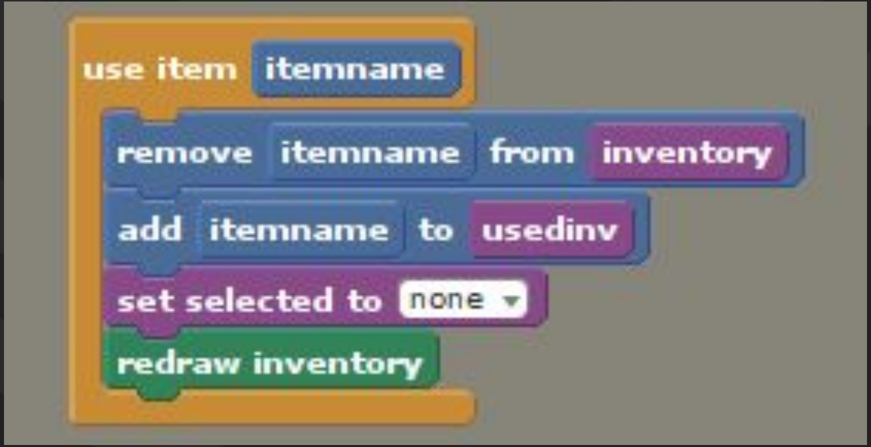
- New event, Input -> On region, select the cabinet doors
- Flow -> if selected = itemcabinetkey
- Custom -> Use itemcabinetkey
- Actors -> slide
- (Slide the left door left and grow it, slide the right door right and shrink it. Give them numbers)



Used Inventory

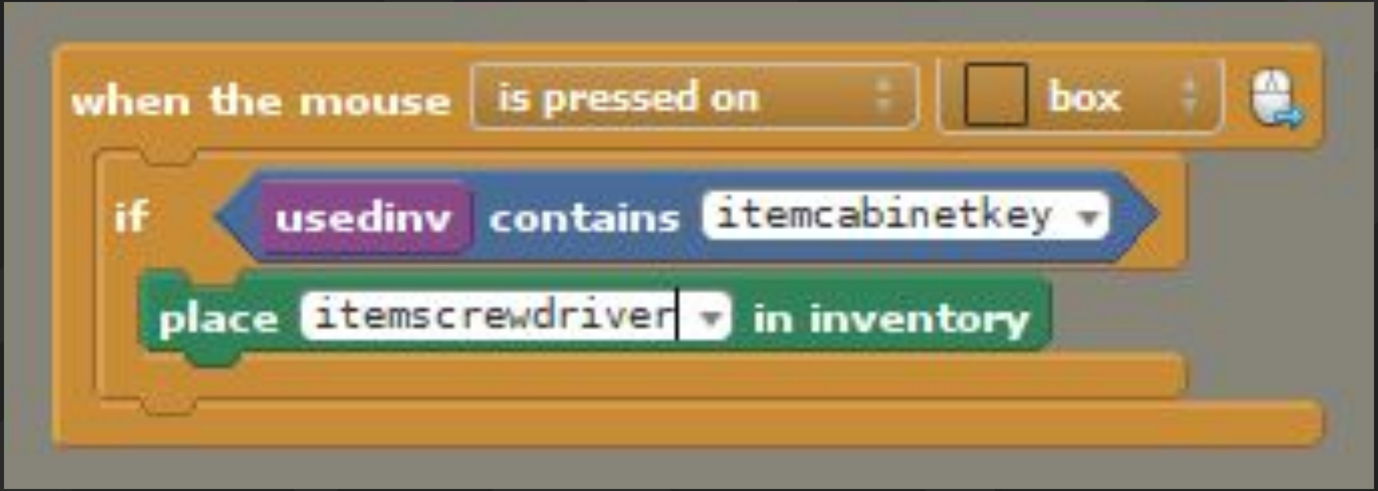
So we can keep track of where we are on that map

- Add new game attribute, ‘usedinv’
- Make it a list
- Go back to the ‘use’ code
- Add block Attributes -> Lists -> Add ‘type’ to ‘usedinv’



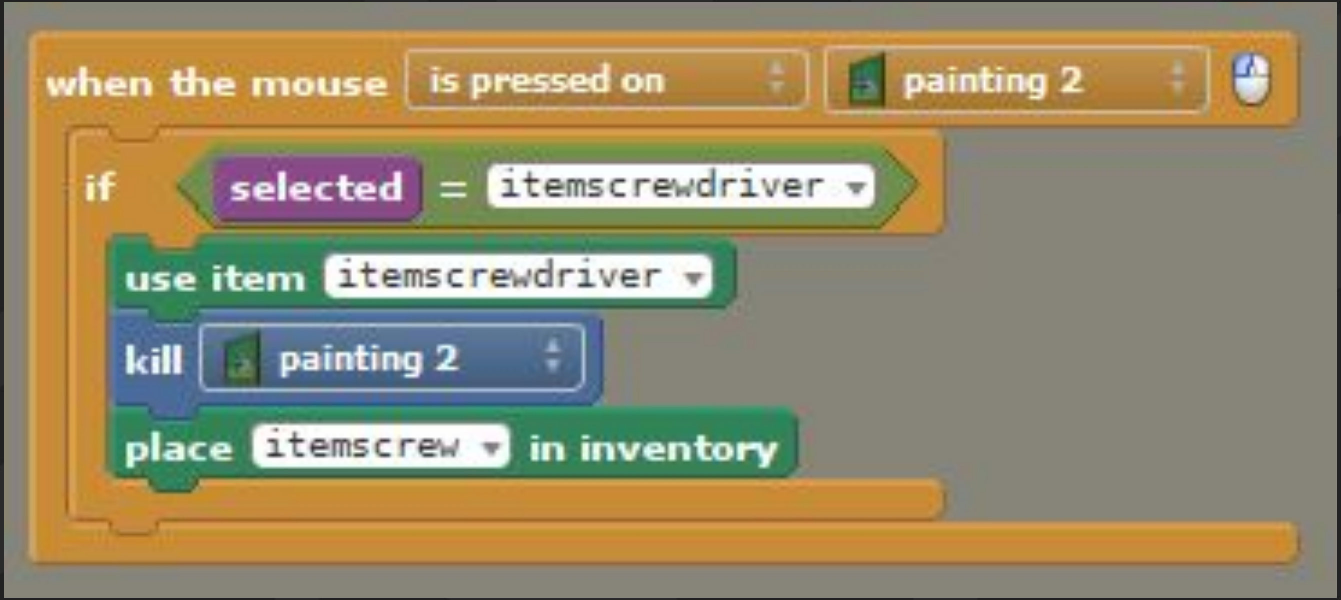
```
use item itemname
    remove itemname from inventory
    add itemname to usedinv
    set selected to none
    redraw inventory
```

Getting the Screwdriver



```
when the mouse is pressed on box
  if usedinv contains itemcabinetkey
    place itemscrewdriver in inventory
```

Use the Screwdriver

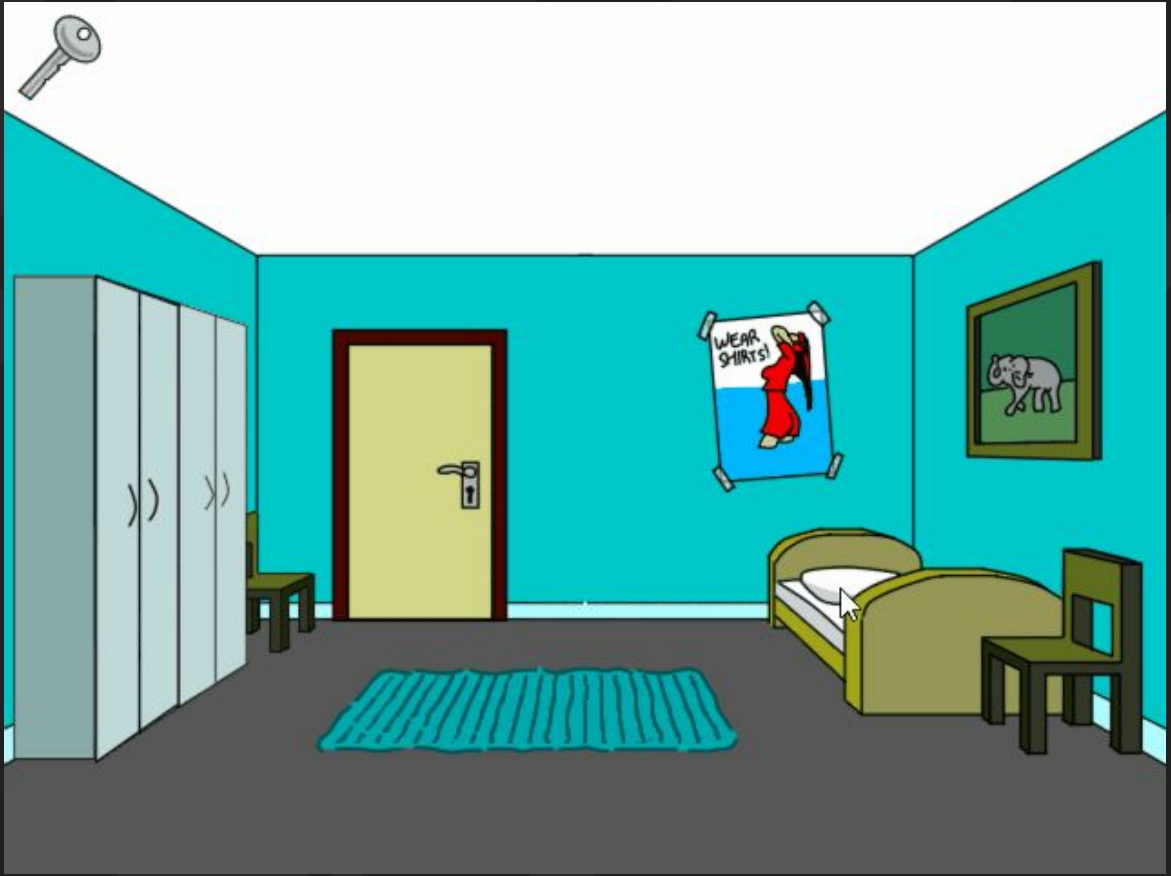


```
when the mouse  is pressed on  painting 2
  if  selected = itemscrewdriver
    use item itemscrewdriver
    kill painting 2
    place itemscrew in inventory
```

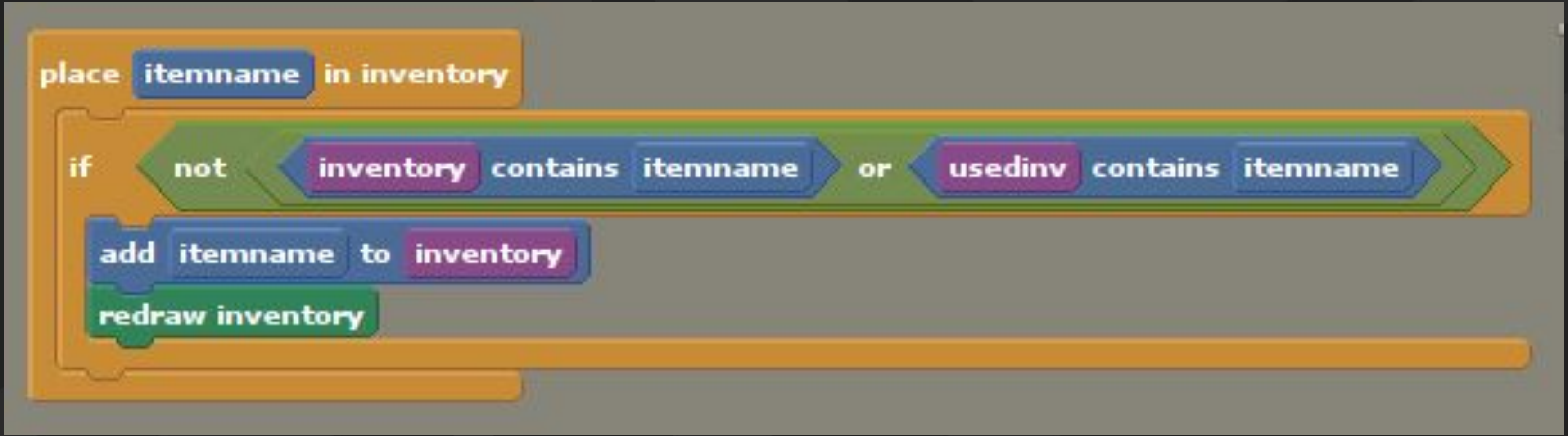
Run the game

(Now we can get the poster, get the cabinet key, and open the cabinet)

(There’s a problem--we can get multiple screwdrivers)



Getting Only One Screwdriver



```
place itemname in inventory
    if not inventory contains itemname or usedinv contains itemname
        add itemname to inventory
        redraw inventory
```

usedinv checks if we’ve ever had the item in our inventory, so we can only pick it up once

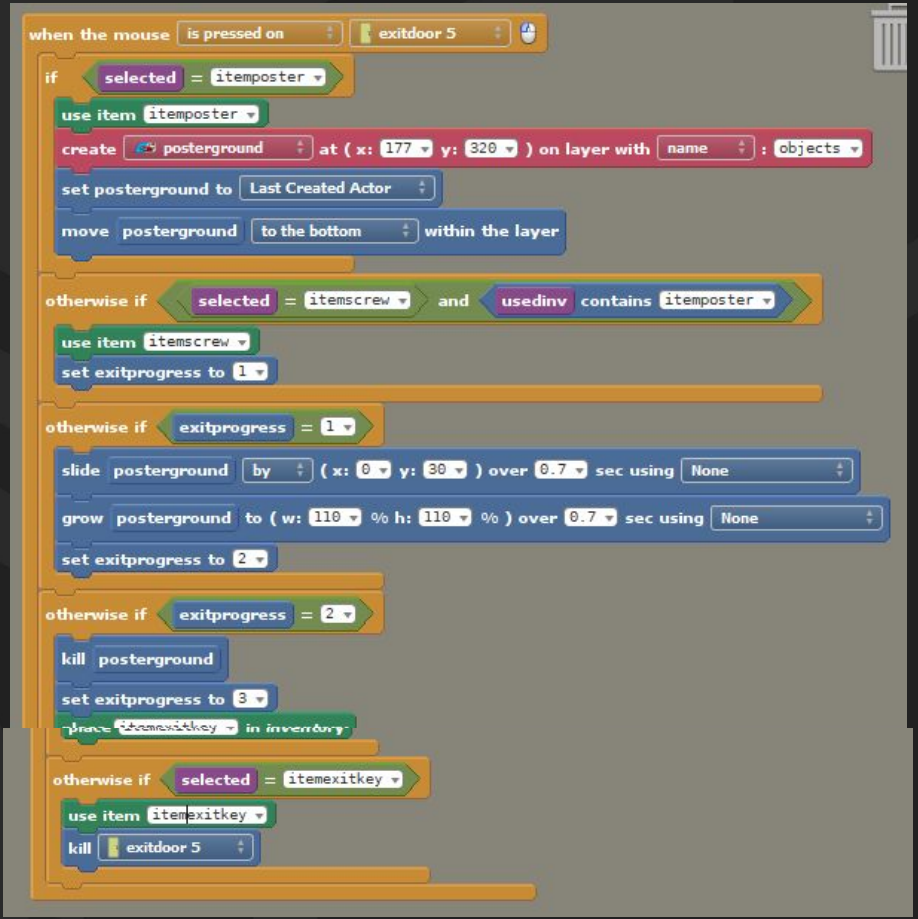
THE DOOR

(Use the screw and poster on the door)

(Gotta figure something out for the sliding)

(MAKE SURE TO CHECK ORDER--poster first)

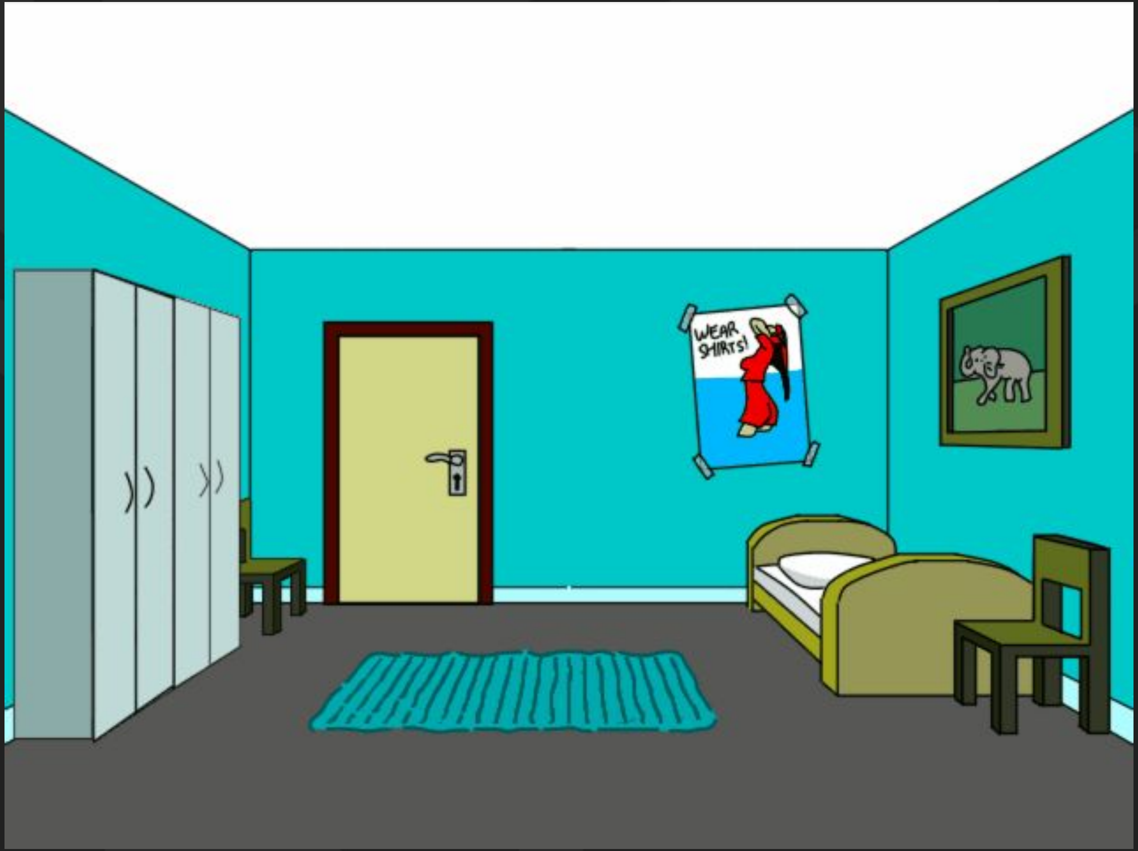
(get key, use key, yippee)



```
when the mouse is pressed on exitdoor 5
  if selected = itemposter
    use item itemposter
    create posterground at ( x: 177 y: 320 ) on layer with name : objects
    set posterground to Last Created Actor
    move posterground to the bottom within the layer
  otherwise if selected = itemscrew and usedinv contains itemposter
    use item itemscrew
    set exitprogress to 1
  otherwise if exitprogress = 1
    slide posterground by ( x: 0 y: 30 ) over 0.7 sec using None
    grow posterground to ( w: 110 % h: 110 % ) over 0.7 sec using None
    set exitprogress to 2
  otherwise if exitprogress = 2
    kill posterground
    set exitprogress to 3
    place itemexitkey in inventory
  otherwise if selected = itemexitkey
    use item itemexitkey
    kill exitdoor 5
```

Run The Game

- (Should be fully functional by now, if a bit boring)



Sign in here!

fsu.devlup.org/signin

DevLUp FSU
GBM #7

Intro to Stencyl

March 20th, 2025