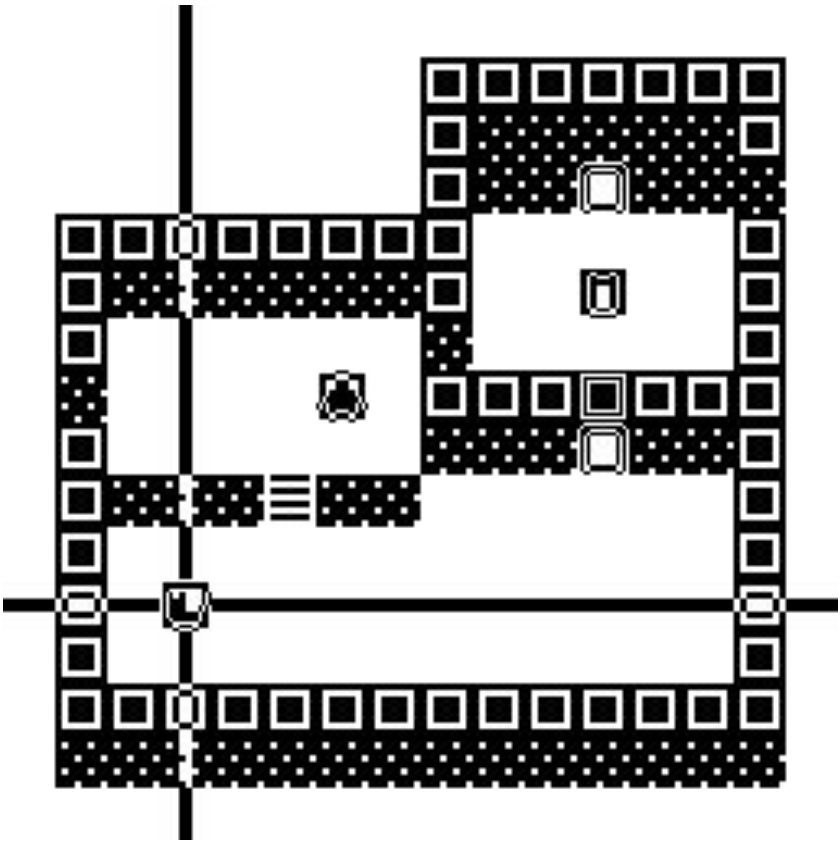


# bipsi game maker user guide

2021/08/24

# intro

bipsi is a tool for making small lo-fi games to play in the browser

this is a simple guide to help you understand and use bipsi to make your own games that you can share with others

bipsi is inspired by and an imitation of adam ledoux's bitsy game maker

bipsi is available at **kool.tools/bipsi**

if you encounter any bugs or need help using bipsi, you can contact me on twitter at @ragzouken, by email at ragzouken@gmail.com

you can also join the bipsi discord at with the following invite: https://discord.gg/mnARVsgSkc

# a bipsi game

bipsi is intended to make the same form of games as bitsy:

you control a character and navigate through interconnected rooms to encounter characters and objects that talk to you

through lo-fi imagery and poetic text, they depict a space, story, or feeling

there is rarely any element of puzzle solving, system simulation, or choice

they usually demand no more than ten or so minutes of play time

# aesthetics

bipsi uses the same lo-fi aesthetics as bitsy:

visually, everything is built from **tiles**: 8x8 pixel images that are individually used to represent characters, or arranged in a 16x16 grid to represent **rooms**

each **room** uses a **palette** of three colors (background, foreground, and highlight) to recolor the **tiles**

some **tiles** are animated, cycling slowly between two frames

dialogue is shown two lines at a time, at double the pixel density of the game (for readability)

# starting prompts

this guide contains descriptions of all the areas of the tool and how to use them but if you want to jump straight in:

try changing the **palette** colors so it looks like a nighttime scene

try changing the layout of starting **room**--remember to update the **walls**

try adding some windows or furniture to the **room**. you'll need to add new **tiles** first

try adding a new **room**. you'll need to use the **exit** template **event** to allow the player to get there

try adding a **character** who has something interesting to **say**

# choosing colors

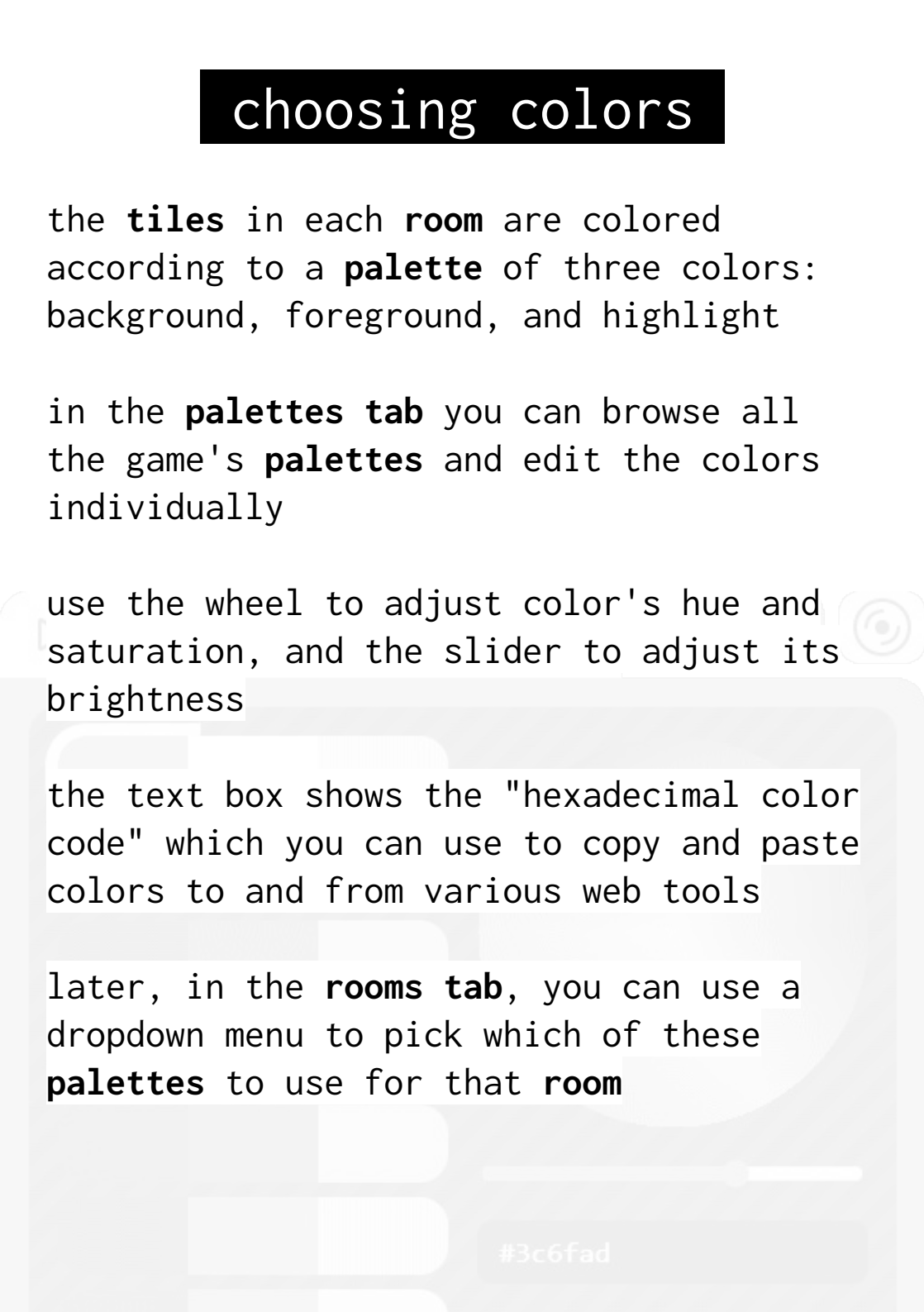
the **tiles** in each **room** are colored according to a **palette** of three colors: background, foreground, and highlight

in the **palettes tab** you can browse all the game's **palettes** and edit the colors individually

use the wheel to adjust color's hue and saturation, and the slider to adjust its brightness

the text box shows the "hexadecimal color code" which you can use to copy and paste colors to and from various web tools

later, in the **rooms tab**, you can use a dropdown menu to pick which of these **palettes** to use for that **room**



# drawing tiles

**tiles** are the visual building blocks in bipsi: they are used individually to represent characters and objects, and arranged together to compose **rooms**

the **draw tiles tab** is split into a tile editing display and a tile browser

in the browser you can select from all the **tiles** in the game and add, delete, and duplicate, and reorder them

you can make changes to the selected tile in the editing display: select a frame and start clicking to edit the appearance of the tile

for non-animated tiles, both frames are the same. the animation toggle sets whether the tile is animated or single frame

# drawing rooms

**rooms** are spaces: the characters live here, and the player explores

the **draw rooms tab** is split into a room editing display and a room and tile browser

the room browser allows you select a **room** for editing, and to copy, paste, or clear it. the tile browser is used to select **tiles** for the editing display

the editing display shows the selected **room** and allows you to paint the selected **tile** into it to change how it looks

there are also tools to paint highlight tiles, select a tiles from the room, shift all tiles at once, and paint **walls**

the **palette** drop down selects which set of colors this **room** should use

# drawing walls

**walls** are portions of a **room** that cannot be walked over by the player

walls are set in the **draw rooms tab** using the **draw walls** tool

you can paint **walls** individually into the room as you would with **tiles**

to save time, you can also **hold shift** and click to fill **walls** over every **tile** of the type you clicked on

**events** tagged as solid will also block the player's movement, but are not considered **walls**

**walls** are used by the room browser to decide how to color the preview images: walls are shown in the foreground color and non-walls in the background color

# events

in bipsi, all of the interactive elements are specified as **events**

**events** allow you to make **exits** between **rooms**, characters that deliver **dialogue**, and more

**events** sit in a **room** waiting to be **touched** by the player

when **touched**, they run through a set of behaviours such as delivering **dialogue**, moving the player to another **room**, etc

each **event** has a table of data **fields** that are used to specify which behaviours it should perform

the event **templates** are preset collections of **fields** you can use for convenience, but all **events** work the same and you can use any **field** on any **event**

# exits

**exits** are spots that move the player between rooms when touched

you can add **exits** to your game with the **exit** event template or by adding an **exit** location field to any event

when the player bumps into or stands on events with an **exit** field, they are moved to a new location

if the **event** also has **dialogue**, the player will not be moved until they have dismissed it

# dialogue

**dialogue** is text that shows on screen for the player to read

you can add **dialogue** to your game with the **character** or **message** event templates, or by adding a **say** field to any **event**

when the player bumps into or stands on events with a **say** field, the **dialogue** is shown on screen for the player to read and page through

you can add special animation effects to the text using the following markup:

```
~~gently waves up and down~~
##shake energetically##
==cycles through the rainbow==
__always visible__
```

unlike in bitsy, scripting does not happen inside dialogue

# playing and sharing

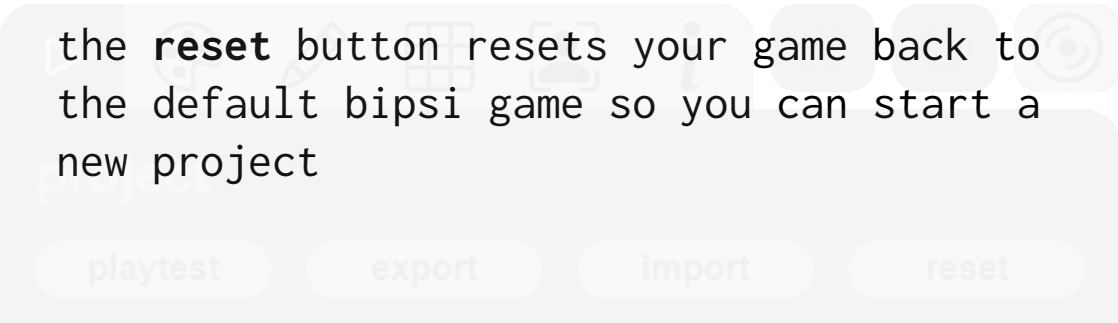
the **playtest tab** is where you come to deal with your game as a whole

the **playtest** button starts a playtest of the game from the beginning, where you can play through your game and check it works as you expect it to

the **export** button prompts you to download a playable copy of your game (as a single html web page). you can send this directly to others, or upload it to your website or itch.io profile

the **import** button prompts you to provide an existing exported bipsi game to load for editing (replaces the current game)

the **reset** button resets your game back to the default bipsi game so you can start a new project



# publishing online

exporting gives you a single html file which is a standalone web page for playing your game

i recommend neocities as a free website host, where you can directly upload your bipsi game export: https://neocities.org

you can also publish your game by uploading it to itch: https://itch.io

for itch, use these settings:

set "**kind of project**" to "**HTML**"

upload your bipsi game export and select "**this file will be played in the browser**"

under "**embed options**" select "**click to launch in fullscreen**" and select "**mobile friendly**"

# tips and tricks

**exits** don't have to be walkable! a player can **touch** an **event** on a **wall** by bumping into it

try not to give each **event** more than a couple of pages of **dialogue**. if you have more to say, put it on another **event**!

it's hard to communicate a whole concept in a single **tile**. use the **palette**, nearby **tiles**, and **dialogue** to add context!

**tiles**, **rooms**, colors, and even **dialogue** all work together to make the game--don't get hung up focusing on just one

remember to **save** often! use the disc icon next to the undo/redo buttons to save your project into your browser. you can also use **export** and **import** to take save management into your own hands

# standard event templates

**character** - similar to bitsy "sprite": a tile that blocks movement and shows dialogue when touched

**exit** - similar to bitsy "exit": invisible and moves the player after touching

**message** - similar to bitsy "item": a tile that shows dialogue when touched and then disappears

**ending** - similar to bitsy "ending": shows ending dialogue when touched

**player avatar** - similar to bitsy "avatar": marks where the player should begin, visible as a tile, contains game title

**custom code** - no defined behavior. entry point for adding custom javascript to your game

# standard event flow

**player attempts to move:**

1. if the destination is unblocked, the player event is moved
2. if there is an event at the destination, touch it
3. otherwise if there is an event at the player position, touch it

**player touches event:**

1. update style based on "page-color"
2. switch audio based on "music"
3. wait for any "title" dialogue
4. wait for any "say" dialogue
5. move the player if there's an "exit" location
6. remove event if tagged "one-time"
7. wait for any "ending" dialogue
8. update player graphic from "set-avatar"

# standard event fields

**is-player (tag)** - this event is the player avatar (unique)

**graphic (tile)** - visible as a tile

**solid (tag)** - block player movement

**one-time (tag)** - remove after touched

**say (dialogue)** - show this dialogue when touched. see **say-mode** for advanced usage

**exit (location)** - move player here after touched

**title (dialogue)** - show this dialogue as a title when touched (put this on the player avatar for a game title)

**ending (dialogue)** - show this dialogue as ending when done touching

# standard event fields

**say-mode (text)** - specifies how to handle multiple **say** fields. each time the **event** is **touched** one **dialogue** is shown:

- **sequence** (default) - step through the **dialogues** in order, repeating the last one when the list runs out
- **cycle** - step through in order, starting from the beginning when the list runs out
- **shuffle** - randomize the order, then step through them, repeating the process when the list runs out

**say-shared-id (text)** - all events with same shared id will share a sequence/cycle/shuffle of dialogue

**page-color (text)** - set the page's background color when touched (put this on the player avatar for initial color)

# standard event fields

**set-avatar (tile)** - change player's graphic to another tile

**transparent (tag)** - draw the event on top of tiles instead of solid background

**is-library (tag)** - this event is the **library** (other events can refer to named **file** fields on this event)

**music (file/text)** - switch looping music track (file or named **library** field)

**stop-music (tag)** - stop looping music

**background, foreground, overlay (file/text)** - show this image (file or **library** reference) on screen on a particular layer

**clear-background, clear-foreground, clear-overlay (tag)** - stop showing image

# glossary

**tile** - the basic unit of graphic used to represent characters and compose **rooms**

**frame** - the individual images making up a **tile**: one for static or two for animated

**rooms** - the spaces that the player moves in. composed of **tiles** and contains **events**

**wall** - specially marked areas of a **room** that the **player** cannot walk through

**palettes** - sets of three colors that are applied to the **tiles** in a **room**

**events** - bundles of behaviour that sit in rooms waiting for the player to activate them. for **dialogue**, **exits**, etc

**player avatar** - the **event** in the game controlled by the player. is **touched** once at the beginning of the game

# advanced: custom scripts

you can redefine an **event's** touch behavior by giving it a **field** called **"touch"** with type **"javascript"**

when **touched**, the **event** will ignore the standard event flow and instead run the **field's** value as javascript code

to run custom code at the beginning of the game, remember that the player avatar is touched after the title is show, so if the player event has javascript in the **touch** field, that will be run

this feature isn't very useful without knowing how bipsi allows you to manipulate it with code--for detailed information on that check out the **scripting guide**

https://kool.tools/bipsi/scripting-guide.pdf

# notes

# about

bipsi is a web tool for making simple browser games in the bitsy style

**kool.tools/bipsi**

this user guide explains how to use bipsi at a beginner level, and some of the finer details that are useful for advanced usage

i'm mark wonnacott a.k.a candle, and i created bipsi

**kool.tools**

thank you to adam ledoux for creating bitsy, em reed for feedback on bipsi, gamemakingtools for feedback on this user guide, and everyone in the candle cove discord for feedback and encouragement