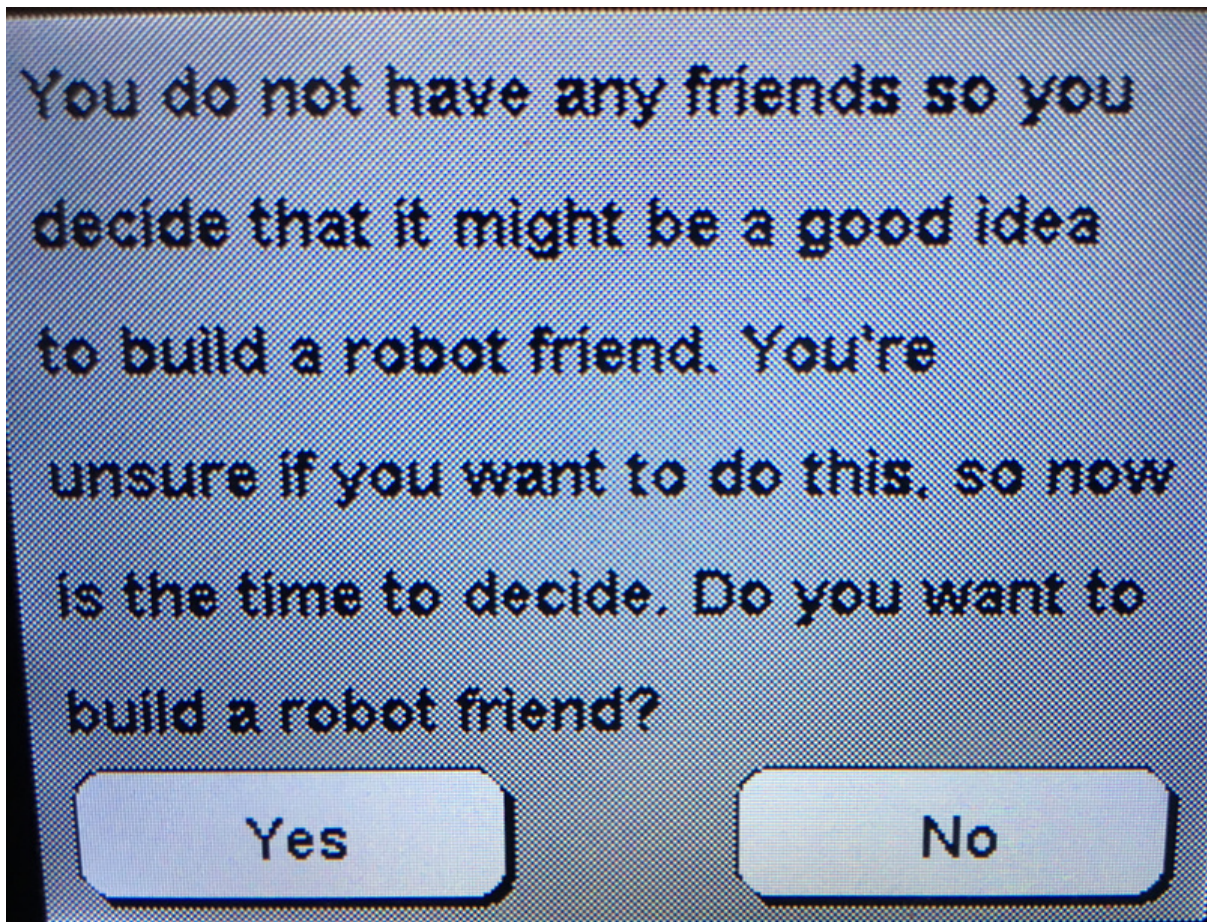




CircuitPython Your Own Adventure

Created by lady ada



<https://learn.adafruit.com/circuit-python-your-own-adventure>

Last updated on 2024-06-03 02:48:47 PM EDT

Table of Contents

Overview	3
<hr/>	
• Circuit Python Your Own Adventure	
• Hypercard	
• Products	
Preparing Your PyPortal	5
<hr/>	
• Libraries	
Designing an Adventure	8
<hr/>	
• Common Patterns	
• Mazes	
• An Example	
Writing an Adventure	15
<hr/>	
• Cards	
• Linking Cards	
• An Example	
• Driving it	
Tips and Tricks	20
<hr/>	
• Images	
• Sounds	
• Storage	
A Fuller Example	21
<hr/>	
• The Flow	
• The JSON	

Overview

```
.RUN ADV11

WELCOME TO ADVENTURE!!  WOULD YOU LIKE INSTRUCTIONS?

YES
SOMEWHERE NEARBY IS COLOSSAL CAVE, WHERE OTHERS HAVE FOUND
FORTUNES IN TREASURE AND GOLD, THOUGH IT IS RUMORED
THAT SOME WHO ENTER ARE NEVER SEEN AGAIN. MAGIC IS SAID
TO WORK IN THE CAVE.  I WILL BE YOUR EYES AND HANDS. DIRECT
ME WITH COMMANDS OF 1 OR 2 WORDS.
(ERRORS, SUGGESTIONS, COMPLAINTS TO CROWTHER)
(IF STUCK TYPE HELP FOR SOME HINTS)

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK
BUILDING . AROUND YOU IS A FOREST. A SMALL
STREAM FLOWS OUT OF THE BUILDING AND DOWN A GULLY.

GO IN
YOU ARE INSIDE A BUILDING, A WELL HOUSE FOR A LARGE SPRING.

THERE ARE SOME KEYS ON THE GROUND HERE.

THERE IS A SHINY BRASS LAMP NEARBY.

THERE IS FOOD HERE.

THERE IS A BOTTLE OF WATER HERE.
```

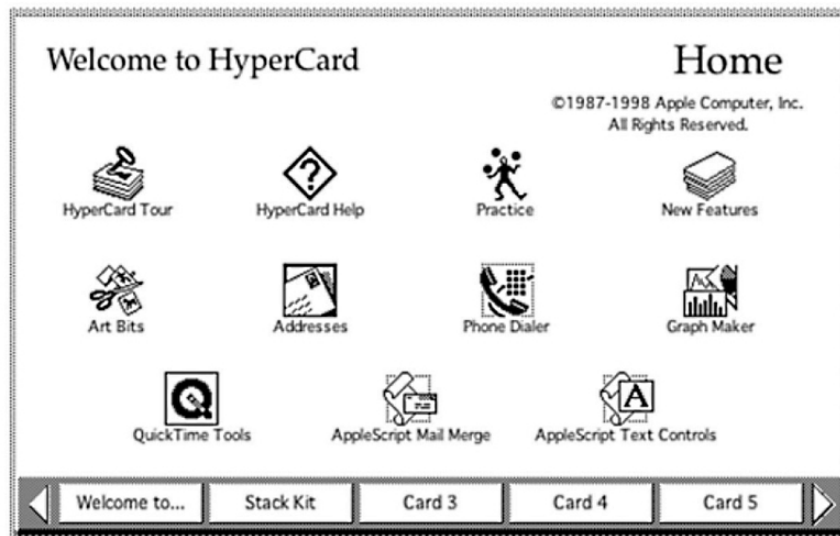
Circuit Python Your Own Adventure

Interactive fiction is a popular game style for computers, the game presents a situation and you, the player, must decide which path to take. Historically, games like ADVENT or Zork did this with text only. A modern implementation is the Bandersnatch episode of Netflix's Black Mirror series.

The basic idea is that at various points in the story, the reader is given a choice about how to proceed. Those choices take the reader down different paths, eventually leading to alternative endings.

This guide is an introduction to the PYOA application that lets you write and run your own adventures on the PyPortal complete with images, sounds, text, and choices.

Hypercard



The system described in this guide is strikingly similar in many ways to an Apple product from the late 80s: Hypercard.

The Adafruit blog as had posts about hypercard that are worth checking out:

- [Apple's HyperCard – history and a possible remake](https://adafru.it/F4g) (<https://adafru.it/F4g>)
- [HyperCard Adventures – a classic Mac and HyperCard emulated in-browser](https://adafru.it/E9x) (<https://adafru.it/E9x>)
- [Learn Mac HyperCard via this 1989 educational package](https://adafru.it/Dgg) (<https://adafru.it/Dgg>)
- [How HyperCard inspired a generation of future developers](https://adafru.it/F6M) (<https://adafru.it/F6M>)

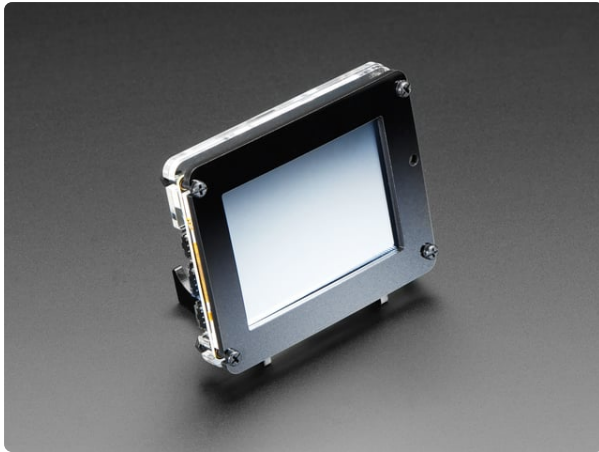
Products



Adafruit PyPortal - CircuitPython Powered Internet Display

PyPortal, our easy-to-use IoT device that allows you to create all the things for the “Internet of Things” in minutes. Make custom touch screen interface...

<https://www.adafruit.com/product/4116>



Adafruit PyPortal Desktop Stand Enclosure Kit

PyPortal is our easy-to-use IoT device that allows you to create all the things for the “Internet of Things” in minutes. Create little pocket...

<https://www.adafruit.com/product/4146>



USB A/Micro Cable - 2m

This is your standard USB A-Plug to Micro-USB cable. It's 2 meters long so you'll have plenty of cord to work with for those longer extensions.

<https://www.adafruit.com/product/2185>

Preparing Your PyPortal



CircuitPython is a programming language based on Python, one of the fastest growing programming languages in the world. It is specifically designed to simplify

experimenting and learning to code on low-cost microcontroller boards. Here is a guide which covers the basics:

- [Welcome to CircuitPython! \(https://adafru.it/cpy-welcome\)](https://adafru.it/cpy-welcome)

Be sure you have the latest CircuitPython for the PyPortal loaded onto your board, as [described here \(https://adafru.it/F6N\)](https://adafru.it/F6N). You will want at least version 4.1 (possibly a beta version of it) for maximum graphics performance.

CircuitPython is easiest to use within the Mu Editor. If you haven't previously used Mu, [this guide will get you started \(https://adafru.it/ANO\)](https://adafru.it/ANO).

Libraries

Plug your PyPortal board into your computer via a USB cable. Please be sure the cable is a good power+data cable so the computer can talk to the board.

A new disk should appear in your computer's file explorer/finder called **CIRCUITPY**. This is the place we'll copy the code and code library. If you can only get a drive named **PORTALBOOT**, load CircuitPython per [the guide mentioned above \(https://adafru.it/F6N\)](https://adafru.it/F6N).

Create a new directory on the **CIRCUITPY** drive named **lib**.

Download the latest CircuitPython driver package to your computer using the green button below. **Match the library you get to the version of CircuitPython you are using.** Save to your computer's hard drive where you can find it.

Go to GitHub to get the latest
CircuitPython library bundle

<https://adafru.it/zB->

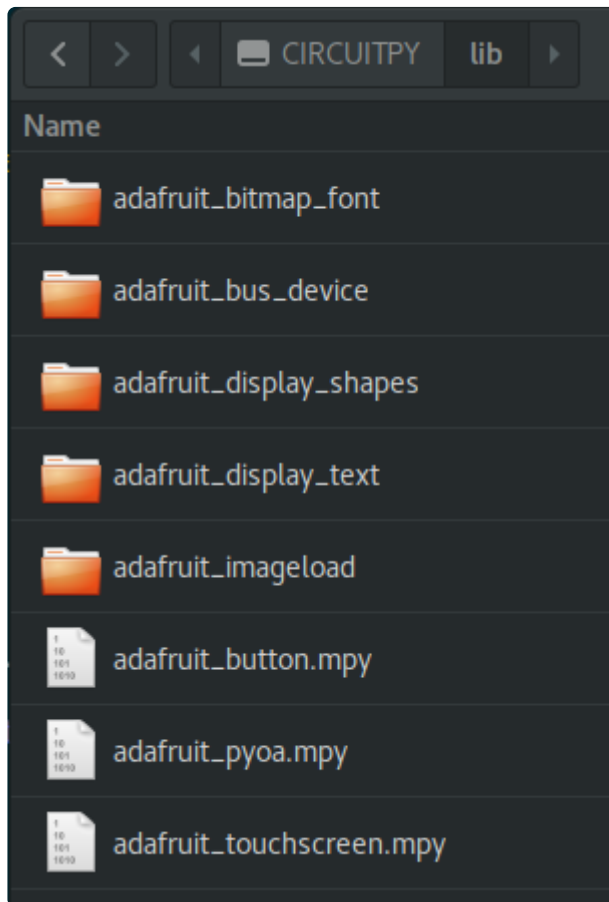
With your file explorer/finder, browse to the bundle and open it up. Copy the following folders and files from the library bundle to your **CIRCUITPY lib** directory you made earlier:

- **adafruit_bitmap_font**
- **adafruit_bus_device**
- **adafruit_display_shapes**
- **adafruit_display_text**

- adafruit_imageload
- adafruit_button.mpy
- adafruit_pyoa.mpy
- adafruit_cursorcontrol.mpy
- adafruit_touchscreen.mpy

All of the other necessary libraries are baked into CircuitPython!

Your **CIRCUITPY/lib** directory should look like the snapshot below.

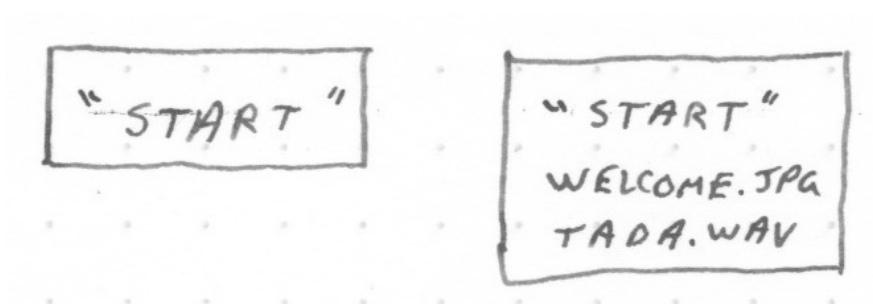


Designing an Adventure



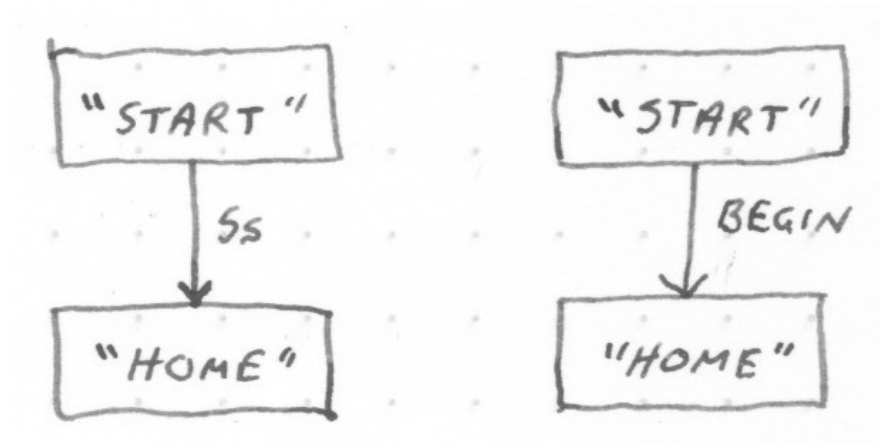
The structure of an adventure can be thought of as a state machine ([see this guide \(https://adafru.it/DtL\)](https://adafru.it/DtL)) or as a flowchart since state transitions are very constrained (time, single button press, or a binary choice). We'll use a limited form of the latter to diagram adventures.

Cards can be represented as a simple box. The box should contain at the very least the id of the card. This will serve to link cards in your diagram to those in the JSON file. You can add more information such as image, text, or sound, but the goal of the diagram is to work on the flow of the adventure, not detailed contents.

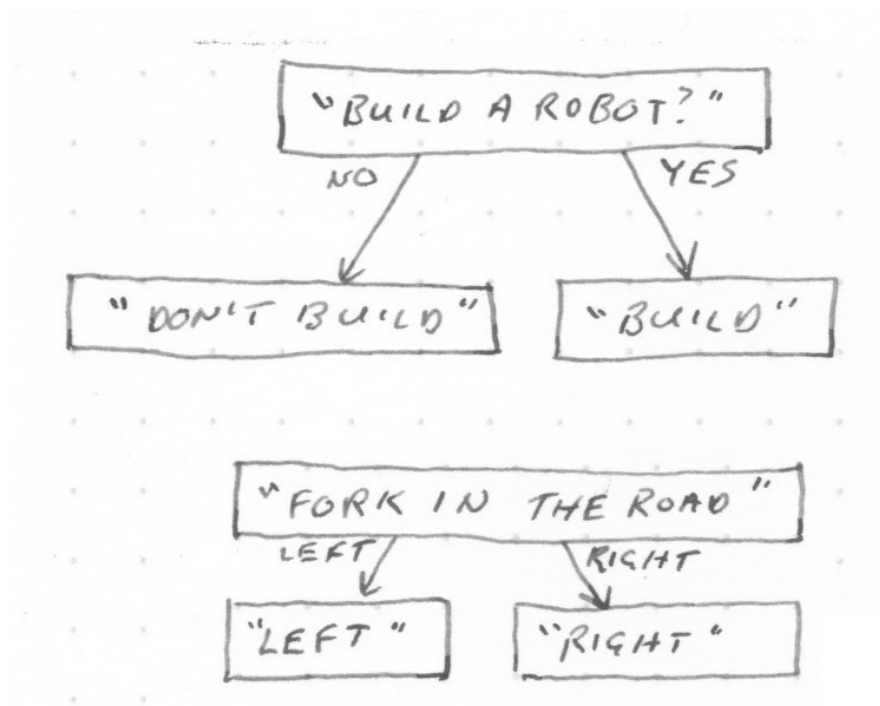


Transitions that are automatic or ones that occur when a single button is pressed can simple be a line connecting pages. Using an arrowhead to show the direction of the

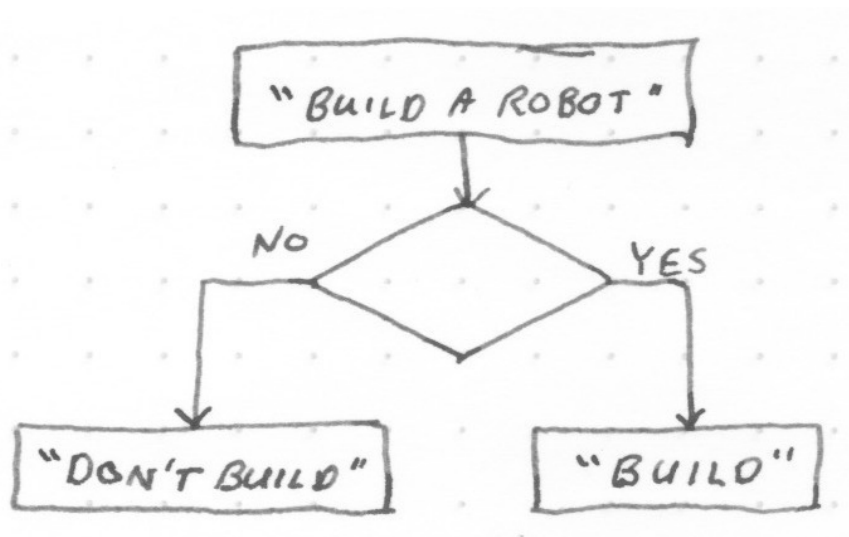
transition is generally a good idea and avoids any confusion. The line can be labelled with a time (e.g. "5s") or a button label (e.g. "Begin"), respectively.



When a card has two buttons, and each has a transition to another card. These should be labelled with the text of the corresponding button. While this will often be a yes/no choice, it can be any choice with two options.



An alternative is to use the diamond symbol from flowcharting. It makes the decision point clearer but does take more space. Both will be used in further examples. Note that the decision diamond is NOT a separate card, but is simply an aspect of the card above.



That's it: cards, simple transitions, and binary choices. These simple tools let us do plenty of interesting things.

Common Patterns

Building on the above atomic pieces, we can step back and look at structures of multiple cards that occur commonly.

Starting

Every adventure starts somewhere. Every story has a beginning. We denote that in our adventure diagram with a circle that is connected to the first card.



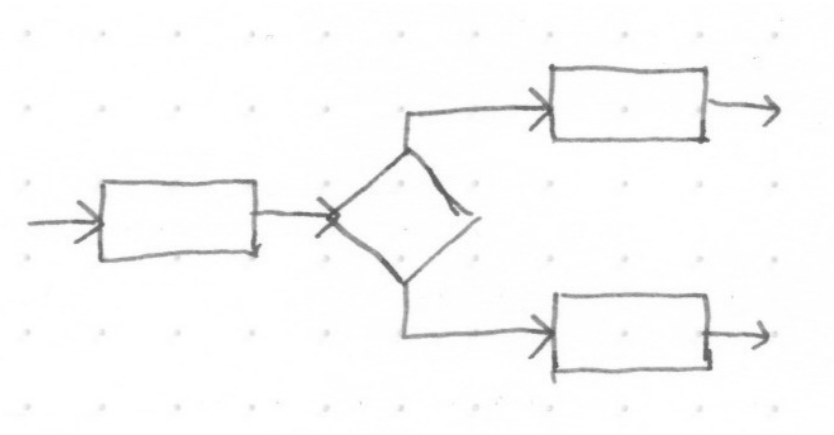
Sequence

A common thing to do is present several cards to the user in sequence, either automatically or by using something like a "Next" button.



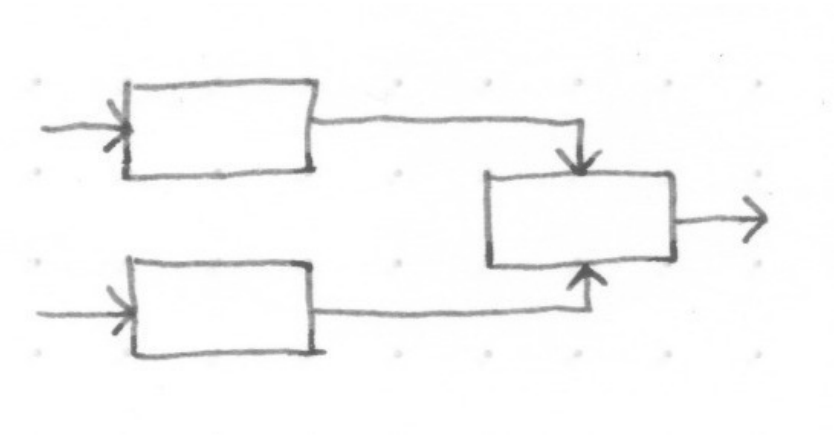
Choices

A big part of the idea of this framework is the ability to present the user with choices that let them have control over their path through the adventure. The framework supports binary choices. These can be Yes/No or any other two options. They provide points in the adventure where the path splits, taking the user down one of them based on their choice.



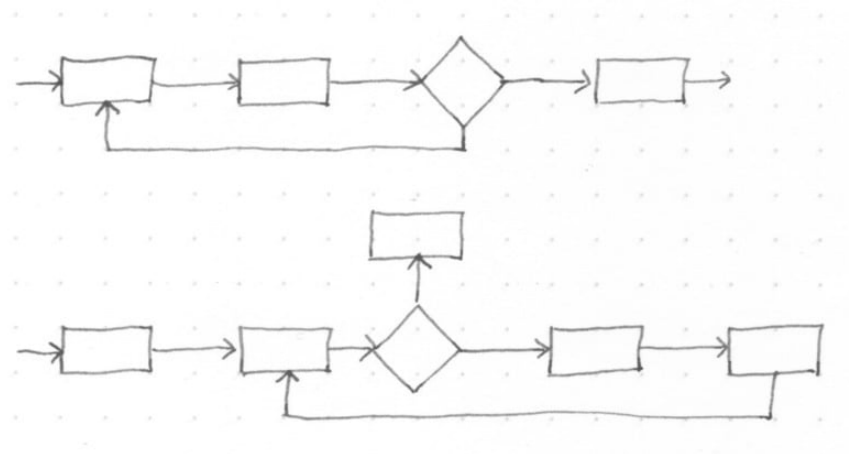
Merges

There is no limit on how many other cards can transition to a given card. You can make use of this when multiple paths in an adventure join back together.



Loops

Loops are when the path through an adventure returns to a card that was previously visited. It can be because of the user's choice or because the path they are on naturally returns them to a previous point; think in terms of a tunnel in a cavern that curves back and leads them to a previous part of the cavern.



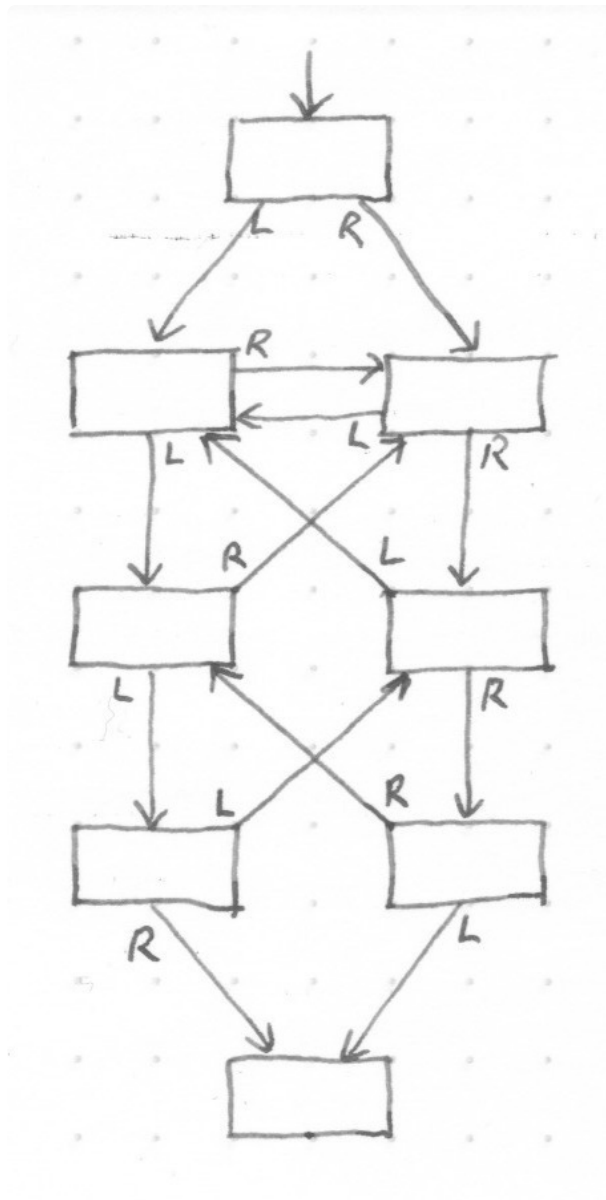
Starting Over

This is a common application of a loop. At some point or points there is a transition back to a card near the start of the adventure. This is often when the adventure reaches a conclusion either good (you saved the prince/princess) or bad (you were eaten by a grue).

Mazes

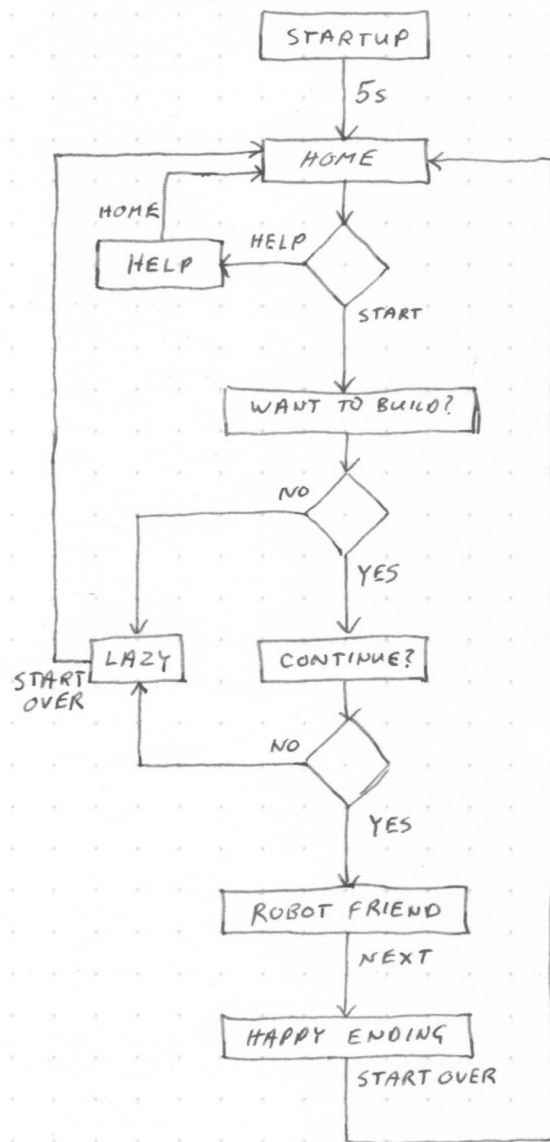
Adding a maze is a fun way to frustrate users. It's simple a collection of rooms with the same description and exits to the left and right. You interconnect them in such a way as to allow the user to go in circles if they make incorrect choices. Eventually, a series of correct choices leads them out of the maze.

For example:

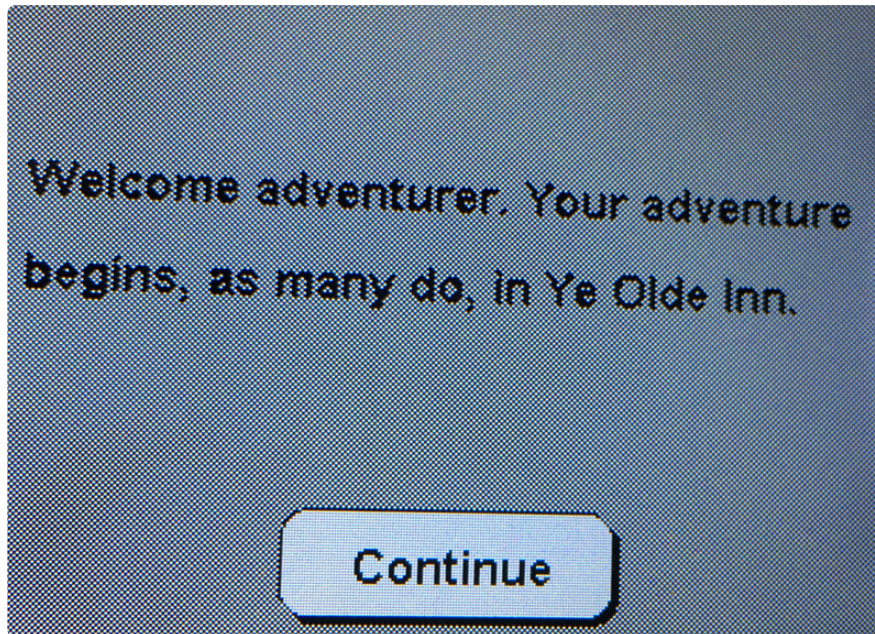


An Example

Here's the diagram for a simple, but complete example: the sample that is packaged with the library.



Writing an Adventure



An adventure is made up of a collection of interconnected cards. Each card represents a single screen on the PyPortal and single point in the adventure.

An adventure is defined by a JSON file that contains an array of objects, each one defining a card.

Cards

Each card defines a screen and point in the adventure. The core structure of a card is made up of several items:

card_id (required)

This is the unique (with the file) string id of the card. It is used to refer to the card from others.

background_image (optional, defaults to **None**)

This is the name of an image file to use as the background image of the card.

text (optional, defaults to **None**)

This is the text that is displayed on the card.

text_color (optional, defaults to black)

The color in which to display the text.

sound (optional, defaults to **None**)

This is the name of an audio file that is played when the card is displayed.

sound_repeat (optional, defaults to **False**)

A True value for this will have the sound (if specified) continue to loop while this card is displayed. Otherwise the sound will play once and stop.

```
{
  "card_id": "startup",
  "background_image": "startup.bmp",
  "sound": "startup.wav",
}
```

Linking Cards

A single card doesn't make for a very compelling adventure. To do that we need several, even many, cards. We also need a way to move from one card to another. Since this is a interactive adventure system, we need to give the user some agency in how they move between cards.

Auto-advance

This way of moving to another card doesn't give the user any agency, it just happens after a specified amount of time. The card advanced to is the next one defined in the adventure file.

In the example below, When the **startup** card is displayed and will stay for 5 seconds before automatically moving to the **home** card.

```
{
  "card_id": "startup",
  "background_image": "startup.bmp",
  "sound": "startup.wav",
  "auto_advance": "5"
},
{
  "card_id": "home",
  "background_image": "home.bmp",
  "sound": "home.wav",
}
```



```
"sound_repeat": "True",  
}
```

Single Button

You can provide a single button for the user to press to advance to another card.

Unlike `auto_advance`, this does not have to be the next card in the file; you use the `card_id` of the card to be moved to. The example below shows using the `button01` fields to specify the button text as well as the target card.

```
{  
  "card_id": "happy_ending",  
  "background_image": "happyending.bmp",  
  "sound": "happy_ending.wav",  
  "sound_repeat": "True",  
  "button01_text": "Home",  
  "button01_goto_card_id": "home"  
}
```

Two Buttons

The final option for moving is to provide the user with a choice. This choice has to have two, and only two, options. The simplest case of this is a yes/no question.

Similar to the single button case, the `button01` fields are used to specify the left button and a set of `button02` fields are used for the right button.

Below is a typical yes/no choice.

```
{  
  "card_id": "1",  
  "background_image": "page01.bmp",  
  "text": "You do not have any friends so you decide that it might be a good idea  
to build a robot friend. You're unsure if you want to do this, so now is the time  
to decide. Do you want to build a robot friend?",  
  "text_color": "0x000001",  
  "sound": "sound_01.wav",  
  "button01_text": "Yes",  
  "button01_goto_card_id": "2",  
  "button02_text": "No",  
  "button02_goto_card_id": "4"  
},
```

And here is an example of a two option choice that isn't a yes/no question. Notice that the only difference is the button text.

```
{  
  "card_id": "home",  
  "background_image": "home.bmp",  
  "sound": "home.wav",  
  "sound_repeat": "True",
```

```

"button01_text": "Help",
"button01_goto_card_id": "help",
"button02_text": "Start",
"button02_goto_card_id": "1"
},

```

An Example

Here is a simple, yet complete, example. This is the file corresponding to the diagram on the previous page.

Clicking on download **Project zip** below will give you a zip file that you will need to dig into to get the **cyoa** directory. Copy that to your **CIRCUITPY** drive.

```

[
  {
    "card_id": "startup",
    "background_image": "startup.bmp",
    "sound": "startup.wav",
    "auto_advance": "5"
  },
  {
    "card_id": "home",
    "background_image": "home.bmp",
    "sound": "home.wav",
    "sound_repeat": "True",
    "button01_text": "Help",
    "button01_goto_card_id": "help",
    "button02_text": "Start",
    "button02_goto_card_id": "want to build?"
  },
  {
    "card_id": "want to build?",
    "background_image": "page01.bmp",
    "text": "You do not have any friends so you decide that it might be a good idea to build a robot friend. You're unsure if you want to do this, so now is the time to decide. Do you want to build a robot friend?",
    "text_color": "0x000001",
    "text_background_color": "0xeeeeee",
    "sound": "sound_01.wav",
    "button01_text": "Yes",
    "button01_goto_card_id": "continue?",
    "button02_text": "No",
    "button02_goto_card_id": "lazy"
  },
  {
    "card_id": "continue?",
    "background_image": "page02.bmp",
    "text": "You spend all day, then all week, then all month building a robot, everyone stops talking to you, however a lot of progress has been made. Do you want to keep making the robots?",
    "text_color": "0xFFFFFF",
    "button01_text": "Yes",
    "button01_goto_card_id": "robot friend",
    "button02_text": "No",
    "button02_goto_card_id": "lazy"
  },
  {
    "card_id": "robot friend",
    "background_image": "page03.bmp",
    "text": "The robot is now your friend, everyone else wishes they had a robot,

```

```

this is the best thing ever. Good work!",
    "text_color": "0xFFFFFF",
    "sound": "Mystery.wav",
    "button01_text": "Next",
    "button01_goto_card_id": "happy ending"
},
{
    "card_id": "lazy",
    "background_image": "page04.bmp",
    "sound": "sound_04.wav",
    "text": "Welp, not only will you not have any friends, you are lazy. What's the
point of playing? Try again.",
    "text_color": "0xFFFFFF",
    "button01_text": "Start Over",
    "button01_goto_card_id": "home"
},
{
    "card_id": "help",
    "background_image": "help.bmp",
    "text": "All you need to do is click the buttons, that's it.\nThis is a new
line.",
    "text_color": "0xFFFFFF",
    "button01_text": "Home",
    "button01_goto_card_id": "home"
},
{
    "card_id": "happy ending",
    "background_image": "happyending.bmp",
    "sound": "happy_ending.wav",
    "sound_repeat": "True",
    "button01_text": "Home",
    "button01_goto_card_id": "home"
}
}
]

```

Driving it

adafruit_pyoa is a library and framework that let's you easily create adventures. You'll need to write a small **code.py** to set it up and make it work. Here's the example from the repo. Copy this file to **CIRCUITPY/code.py**.

```

# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import board
import storage
from adafruit_pyoa import PYOA_Graphics

try:
    try:
        import sdcardio

        sdcard = sdcardio.SDCard(board.SPI, board.SD_CS)
    except ImportError:
        import adafruit_sdcard
        import digitalio

        sdcard = adafruit_sdcard.SDCard(
            board.SPI(),
            digitalio.DigitalInOut(board.SD_CS),
        )
    vfs = storage.VfsFat(sdcard)
    storage.mount(vfs, "/sd")

```

```
    print("SD card found") # no biggie
except OSError:
    print("No SD card found") # no biggie

gfx = PY0A_Graphics()

gfx.load_game("/cyoa")
current_card = 0 # start with first card

while True:
    print("Current card:", current_card)
    current_card = gfx.display_card(current_card)
```

This shows how to hook up to the SD card interface so you can put the adventure JSON file along with images and sounds on an SD card. The example puts it on the flash file system for simplicity.

There are three key things that this code does to make the adventure run:

1. creates an instance to the `PY0A_Graphics` class that does the work,
2. loads the game by telling the framework where the adventure data is located (note that the JSON file must currently be called `cyoa.json`) and sets the index of the start card
3. goes into an infinite loop that calls `display_card(current_card)` in the framework which returns a new value for `current_card`.

Tips and Tricks

Images

Background card images should be 320x240, 16 or 24 bit BMPs. Most image editing/creation application will provide a way to export files in those formats. Remember that using smaller depth (i.e. 16 bit instead of 24 bit) will result in smaller image files. This isn't as much a concern with the PyPortal, especially if you have your files on an SD card.

Sounds

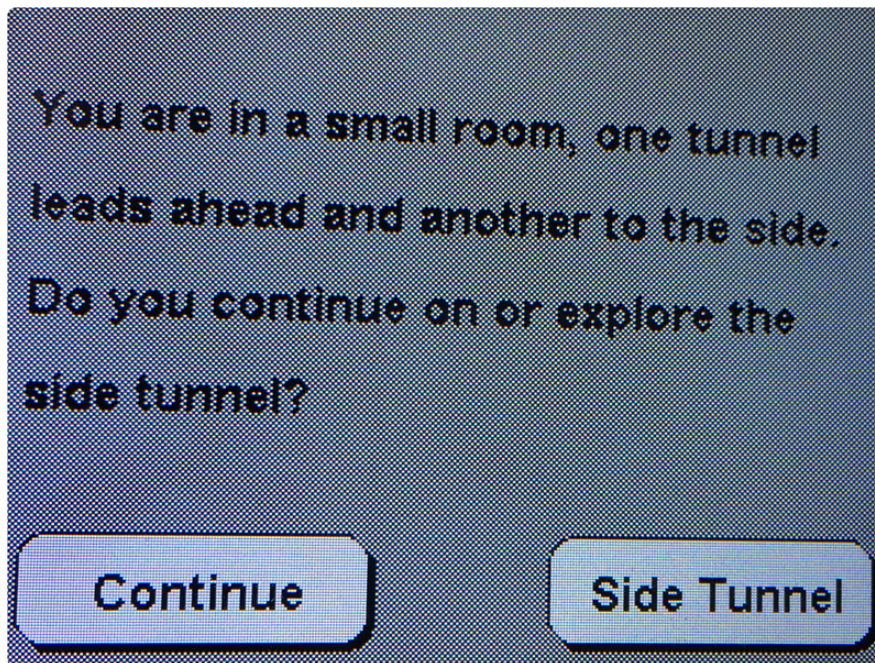
[Freesound](https://adafru.it/DOA) (<https://adafru.it/DOA>) is a source of free to use audio for various purposes. These are generally uploaded by community members and have varying quality.

Regardless where you get your sounds, they will likely have to be converted to work/fit on the PyPortal. See the [guide on sound file conversion](https://adafru.it/BvU) (<https://adafru.it/BvU>) for details on how to convert audio files to this format.

Storage

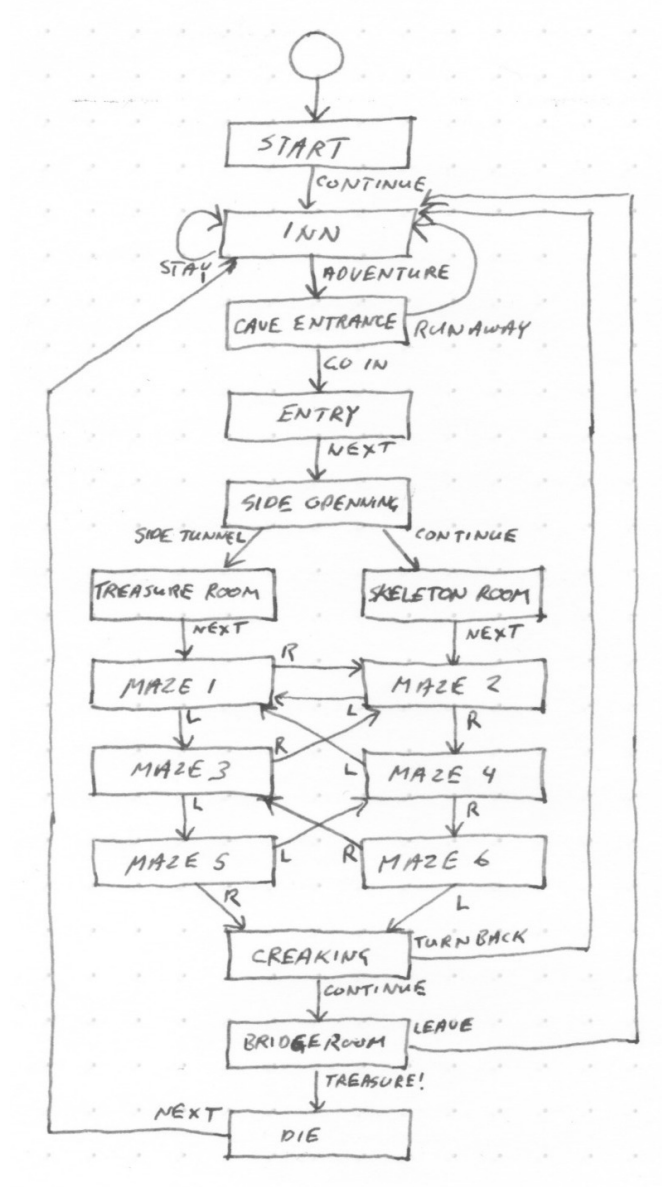
The example shown uses the PyPortal's flash file system to store the adventure files. To more easily share your adventures or to handle very large ones, you can use a micro SD card. The `code.py` shown has the code already to use the card, and all you need to do is specify `/sd` (or a subdirectory) in the call to `load_game`.

A Fuller Example



Here's a richer example: a cave exploration. Note that you'll need to load `"/cave"` instead of `"/cyoa"`.

The Flow



The JSON

Clicking on download **Project zip** below will give you a zip file that you will need to dig into to get the **cave** directory. Copy that to your **CIRCUITPY** drive.

```
[
  {
    "card_id": "start",
    "background_image": "page01.bmp",
    "text": "Welcome adventurer. Your adventure begins, as many do, in Ye Olde
Inn.",
    "text_color": "0x000001",
    "button01_text": "Continue",
    "button01_goto_card_id": "inn"
  },
  {
```

```

        "card_id": "inn",
        "background_image": "page01.bmp",
        "sound": "pub.wav",
        "sound_repeat": "True",
        "text": "This is a peaceful, happy inn with plentiful drink, tasty food,
and friendly staff.",
        "text_color": "0x000001",
        "button01_text": "Stay",
        "button01_goto_card_id": "inn",
        "button02_text": "Go!",
        "button02_goto_card_id": "cave entrance"
    },
    {
        "card_id": "cave entrance",
        "background_image": "page01.bmp",
        "text": "There is a dark cave in the hillside before you.",
        "text_color": "0x000001",
        "sound": "cave.wav",
        "sound_repeat": "True",
        "button01_text": "Enter",
        "button01_goto_card_id": "entry",
        "button02_text": "Run",
        "button02_goto_card_id": "inn"
    },
    {
        "card_id": "entry",
        "background_image": "page01.bmp",
        "sound": "cave.wav",
        "sound_repeat": "True",
        "text": "You are in a dark, narrow tunnel.",
        "text_color": "0x000001",
        "button01_text": "Next",
        "button01_goto_card_id": "side opening"
    },
    {
        "card_id": "side opening",
        "background_image": "page01.bmp",
        "sound": "cave.wav",
        "sound_repeat": "True",
        "text": "You are in a small room, one tunnel leads ahead and another to the
side. Do you continue on or explore the side tunnel?",
        "text_color": "0x000001",
        "button01_text": "Continue",
        "button01_goto_card_id": "skeleton room",
        "button02_text": "Side T.",
        "button02_goto_card_id": "treasure room"
    },
    {
        "card_id": "treasure room",
        "background_image": "page01.bmp",
        "sound": "cave.wav",
        "sound_repeat": "True",
        "text": "There is a pile of treasure here. Congratulations!",
        "text_color": "0x000001",
        "button01_text": "Next",
        "button01_goto_card_id": "maze 1"
    },
    {
        "card_id": "skeleton room",
        "background_image": "page01.bmp",
        "sound": "cave.wav",
        "sound_repeat": "True",
        "text": "There is a skeleton on the floor. From the items around it, it
seems to be that of an unfortunate adventurer.",
        "text_color": "0x000001",
        "button01_text": "Next",
        "button01_goto_card_id": "maze 2"
    },
    {

```

```

    "card_id": "maze 1",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "maze 3",
    "button02_text": "Right",
    "button02_goto_card_id": "maze 2",
  },
  {
    "card_id": "maze 2",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "maze 1",
    "button02_text": "Right",
    "button02_goto_card_id": "maze 4"
  },
  {
    "card_id": "maze 3",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "maze 5",
    "button02_text": "Right",
    "button02_goto_card_id": "maze 2"
  },
  {
    "card_id": "maze 4",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "maze 1",
    "button02_text": "Right",
    "button02_goto_card_id": "maze 6"
  },
  {
    "card_id": "maze 5",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "maze 4",
    "button02_text": "Right",
    "button02_goto_card_id": "creaking"
  },
  {
    "card_id": "maze 6",
    "background_image": "page01.bmp",
    "sound": "cave.wav",
    "sound_repeat": "True",
    "text": "There are passages to the left and right.",
    "text_color": "0x000001",
    "button01_text": "Left",
    "button01_goto_card_id": "creaking",
    "button02_text": "Right",

```



```

        "button02_goto_card_id": "maze 3"
    },
    {
        "card_id": "creaking",
        "background_image": "page01.bmp",
        "sound": "creak.wav",
        "sound_repeat": "True",
        "text": "You hear an ominous creaking from around the corner",
        "text_color": "0x000001",
        "button01_text": "Cont.",
        "button01_goto_card_id": "bridge room 1",
        "button02_text": "Go back",
        "button02_goto_card_id": "inn"
    },
    {
        "card_id": "bridge room 1",
        "background_image": "page01.bmp",
        "sound": "creak.wav",
        "sound_repeat": "True",
        "text": "There is a creaking, rickity wooded bridge leading across a gaping
chasm.",
        "text_color": "0x000001",
        "button01_text": "Cont.",
        "button01_goto_card_id": "bridge room 2"
    },
    {
        "card_id": "bridge room 2",
        "background_image": "page01.bmp",
        "sound": "creak.wav",
        "sound_repeat": "True",
        "text": "At the other end is a large treasure chest. There is also a short
tunnel with daylight at the end.",
        "text_color": "0x000001",
        "button01_text": "Treasure!",
        "button01_goto_card_id": "die",
        "button02_text": "Leave",
        "button02_goto_card_id": "inn"
    },
    {
        "card_id": "die",
        "background_image": "page01.bmp",
        "sound": "scream.wav",
        "text": "The bridge gives way and you fall to a painful death.",
        "text_color": "0x000001",
        "button01_text": "Next",
        "button01_goto_card_id": "inn"
    }
}
]

```