Contents

1 Renpy Tutorial

The goal is write a good story, not to finish all the exercises. If the intermediate exercises are too difficult, work on making your story longer and better instead!

1.1 Download

Download Renpy bundled with a copy of a the starter project here: https://github.com/electronstudio/renpy_starter/releases/download/1/renpy-coderdojo1.zip

On *Windows* just unzip the file to your Desktop and double click **renpy.exe** to start.

On *Mac* after you unzip you will need to type this in to Terminal:

```
xattr -r -d com.apple.quarantine Downloads/renpy-coderdojo1/renpy.app
xattr -r -d com.apple.quarantine Downloads/renpy-coderdojo1/editra/
    Editra-mac.app
```

1.2 Basics (Beginner)

In Renpy, click **script.rpy** to open the editor. You will be given a choice of editors. On Windows and Mac I recommmend you select **Editra**. On Linux you are probably best with **System editor**. There is also **Atom** which is nice but very big so it will probably be slow to download and run. You can change your editor later in Renpy preferences.

The script has already been started for you. Type the remaining lines. Be careful to type indents with the tab key. You can add extra blank lines to make the script easier to read.

```
label start:
    call basic
    return

define k = Character("Kieran")
define r = Character("Reena", who_color="#ff0000")

label basic:
    scene bg school
    "I am the narrator."
    show kieran normal
    "Kieran" "I am Kieran."
    show kieran at left
    k "Defined as 'k' for short."
    show reena normal
    r "I am Reena and my colour is red."
    show reena at right with move
    r "I can move my position."
    scene bg space with fade
    show kieran angry
    play sound spacetrash1
    "POW" with vpunch
    k "We are in space now."
    return
```

Each time you change the script you should press **ctrl+S** to save and then quit the game and click **Launch Project** again. (You can reload the game without quitting but things often do not work if you do that.)

1.2.1 Backgrounds

`scene` changes the background. You must follow it with the name of a file in your images directory.

1.2.2 Characters

`show` shows a character. You must follow it with the name of a file in your images directory. You can change the position by adding `at left`, `at right`, `at center`.

Character variables store the name and color of a character to save you from typing it every time the character speaks. You `define` them at the top of the script.

Characters disappear automatically at the end of a scene, but you can also `hide` them if you need to.

1.2.3 Transitions

Adding `with move` makes a character move. You can also `with zoomin` or `with moveinright` when the character first appears.

There are several transitions for scenes:

```
with fade
```

```
with vpunch
```

```
with dissolve
```

```
with pixellate
```

1.2.4 Tasks

1. Continue the conversation between Kieran and Reena.
2. Define another character and have him speak.
3. Change the expressions of the characters.
4. Change the scene background.
5. Write your own story.

1.3 Flow control (Beginner)

Each example is separate but you can put them all in the same `script.rpy` file, so long as you remember to change the `call` at the top to the label you are currently working on.

IMPORTANT: *On line 2, change `call basic` to `call flow`.*

Then add this code at the end of the program. (Line numbers shown here will **not** match the line numbers in your program.)

```
label flow:
    scene bg lake
    show kim smile with moveinright
    "Kim" "This is the lake"
    menu:
        "Where should we play our music?"
        "The Park":
            call park
        "The school":
            call boring
    scene bg lake
    "Back to the lake!"
    stop music
    return

label park:
    play music music4
    scene bg park
    "OK"
    return

label boring:
    play music music1
    scene bg school
    "OK"
    return
```

Renpy scripts are very similar to Python programs. You can even write Python commands in a Renpy script, but it is easier to write Renpy commands because they are shorter.

`label` is similar to Python `def` for creating a function. The first label must be called `start`.

After creating a label we use a colon and then everything is *indented* using the *tab key*, just like in a Python function.

To end the function/label we use `return` just like Python. Unlike Python, Renpy does not return automatically return if you leave it out. I recommend you always return at the end of a label to avoid confusion. When you return from the start label your game will end.

To allow the player to choose what happens next in the story, use `menu`.

To go to a label we use `call` command, which is like `()` in Python.

1.3.1 Tasks

1. Add another location to the menu. Create a label for this location.
2. Write a story that has different choices. Perhaps some choices lead to a good ending and other choices lead to a bad ending.

1.4 Input and conditionals (Intermediate)

On line 2, change `call flow` *to* `call input`.

Then add this new code at the end of the program. (Line numbers shown here will **not** match the line numbers in your program.)

```
1  define player = Character("[name]")
2  define jen = Character("Jen", who_color="#00ff00")
3
4  label input:
5      scene bg forest
6      show jen normal with zoomin
7      $ name = renpy.input("What is your name?")
8      player "My name is [name]."
9      jen "Hi [name]"
10     if name == "Richard":
11         jen "That is a good name."
12     elif name == "Nick":
13         jen "That is a silly name."
14     else:
15         jen "I don't know you."
16     return
```

Note the $ sign allows you to enter any python statement. Mostly we use when we want to set a variable because Renpy does not have a built in command for this.

1.4.1 Tasks

1. Add additional `elif` statements for your name and your friend's name.

1.5 Booleans (Intermediate)

On line 2, change `call flow` *to* `call input`.

Then add this new code at the end of the program. (Line numbers shown here will **not** match the line numbers in your program.)

```
1  define player = Character("[name]")
2  define jen = Character("Jen", who_color="#00ff00")
3
4  label input:
5      scene bg forest
6      show jen normal with zoomin
7      $ name = renpy.input("What is your name?")
8      player "My name is [name]."
9      jen "Hi [name]"
10     if name == "Richard":
11         jen "That is a good name."
12     elif name == "Nick":
13         jen "That is a silly name."
14     else:
15         jen "I don't know you."
16     return
```

1.6 Sound and music

You can play a sound file with `sound play`. You can play a loop of music with `music play`. Look in the **audio** directory to see what sounds are available. You can record or download your own sounds. They should be **.ogg** or **.mp3** format.

1. Add `music play music1` to somewhere in the script.
2. Add a sound effect of your choice to the script.

1.7 Integers (Advanced)

On line 2, change `call input` to `call booleans`.

Then add this new code at the end of the program. (Line numbers shown here will **not** match the line numbers in your program.)

```
1  define has_book = False
2
3  label booleans:
4      scene bg lake
5      show malucy serious
6      "Malucy" "Knowledge is power."
7      menu:
8          "Choose one item to take."
9          "Book":
10             show item book
11             $ has_book = True
12         "Sword":
13             show item sword
14     "You take the item"
15     scene bg lake night with fade
16     show gravekeeper
17     play sound spacetrash2
18     "A monster attacks" with hpunch
19     if has_book:
20         play sound laser1
21         hide gravekeeper with zoomout
22         "You use the book to cast a spell on the monster."
23     else:
24         scene bg black
25         play sound game_over
26         "Your sword is useless and the monster kills you." with vpunch
27     return
```

1.7.1 Tasks

1. Write a quiz where you get points for choosing the correct answer.
2. Add questions that use `renpy.input` to get the player to type an answer.

1.8 Integers (Advanced)

On line 2, change `call booleans` *to* `call input`.

Then add this new code at the end of the program. (Line numbers shown here will **not** match the line numbers in your program.)

```
define points = 0

label integers:
    scene bg stars
    show mustafa crying
    "Mustafa" "I am sad because I do not have any friends."
    menu:
        "I will be your friend.":
            $points = points + 1
        "That is because you are fat.":
            $points = points - 10
    scene black
    "You scored [points] kindness points."
    if points > 0:
        play sound congratulations
        "You win."
    else:
        play sound you_lose
        "You lose"
```

1.9 Your own artwork

You can draw your own backgrounds and characters using a graphics program, e.g. Krita. You can also download images from the web. Backgrounds should be `1280x720` size. Characters should be about `400x720`. You might have to load them into Krita and resize them. You must save them as `.png` or `.jpg` files in the `images` directory.

1. Draw or download your own background.
2. Draw or download your own character.