

ERPL

DSL for escape rooms

TIAGO LUÍS DIAS DA SILVA

DEPARTAMENTO DE INFORMÁTICA, UNIVERSIDADE DO MINHO, BRAGA, PORTUGAL,

JOSÉ JOÃO ALMEIDA

ALGORITMI, LASI, DEPARTAMENTO DE INFORMÁTICA, UNIVERSIDADE DO MINHO, BRAGA, PORTUGAL

Motivation

Why Escape Rooms?

- Present in education, tourism and leisure
- Promote cognitive skills
- Promote teamwork
- Problem solving in a stimulating environment
- Great demand for unique experiences
- Lots of diversity in themes, formats and mechanics.

Why a DSL?

- Specificity in a domain
- Ease of understanding
- Versatile and easy to reuse (e.g. templates and scripting)
- Fast development

Goals

- Facilitating the creation of escape rooms
- Promoting reusability and modularity
- Ensuring scalability and extensibility

Technologies

Tools that can be used to create Escape Rooms

Interactive Presentation Tools. Genially, PowerPoint, Prezzi, etc.

Scratch. Visual programming platform designed to facilitate the creation of interactive projects, games and animations.

CospacesEdu. Tool for creating three-dimensional environments.

Room Escape Maker. Free online escape room editor game.

Escapp. Tool for configuring escape rooms aimed at education.

Engines for creating 2D games. GameSalad, BuildBox, Godot, etc.



Tools used in this project

Lark. Python parsing library



Pygame. Python game library.



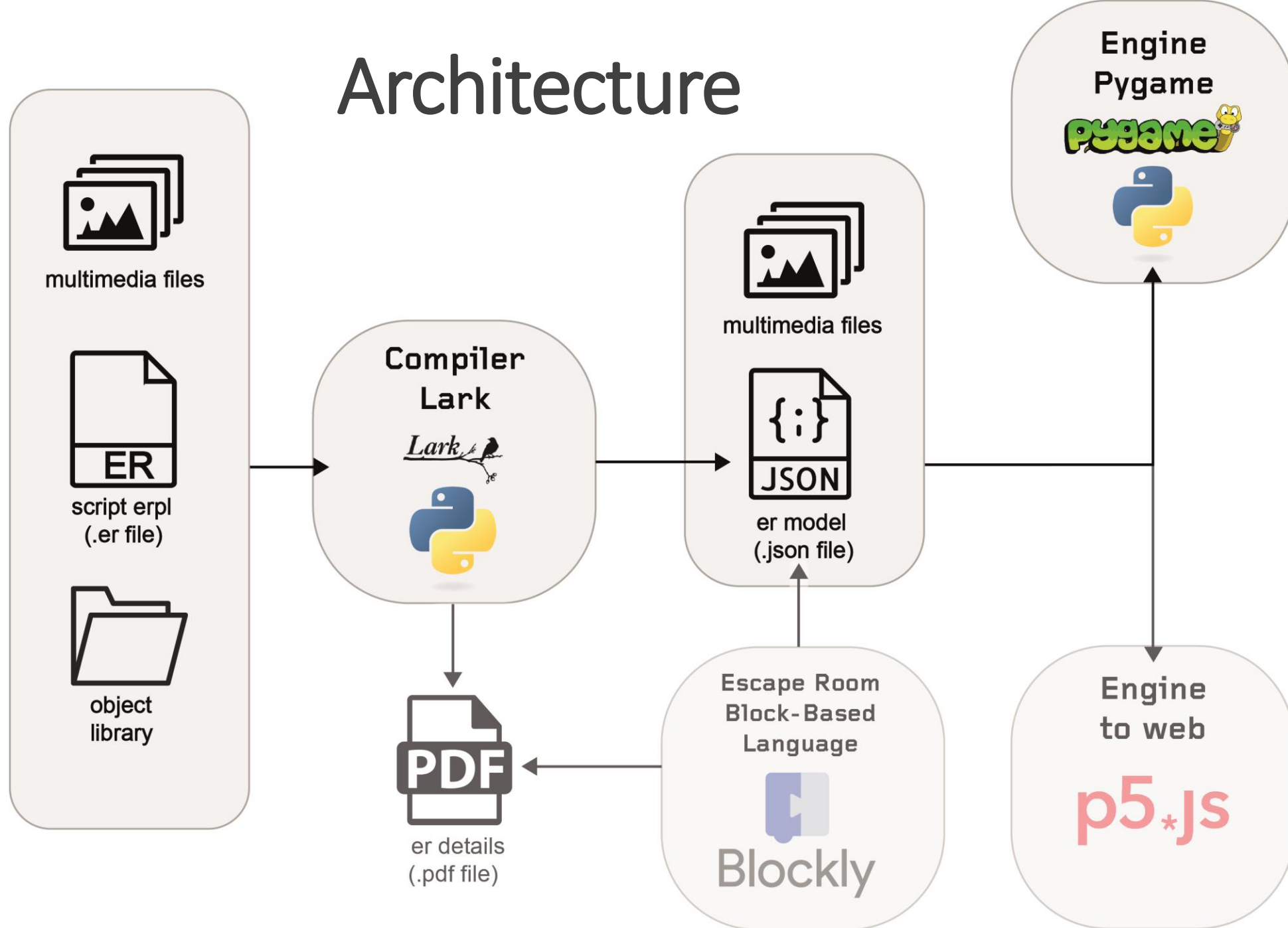
P5. JavaScript library aimed at creating graphics and interactivity in web contexts.



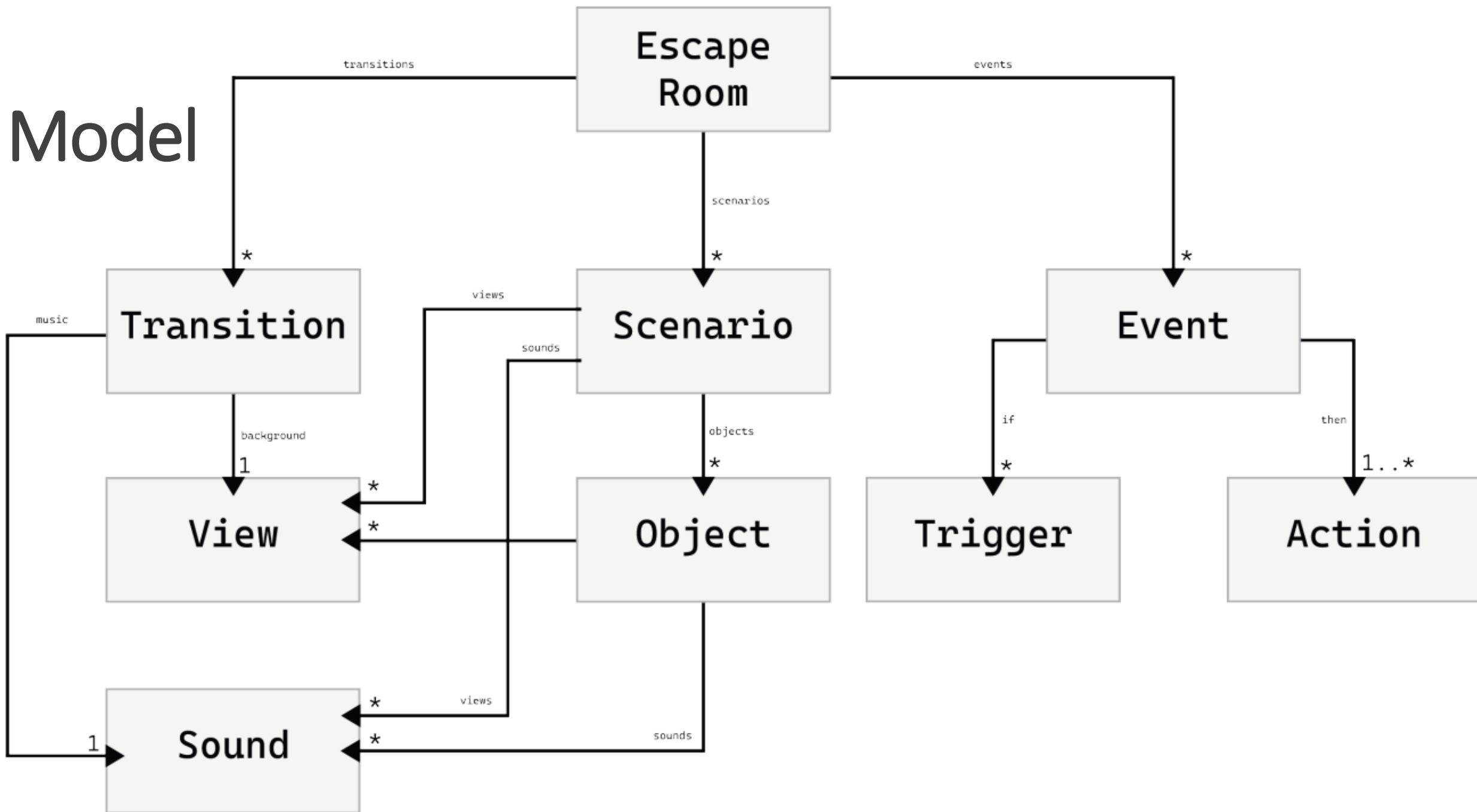
Blockly. Client-side library for the JavaScript programming language to create block-based visual programming languages and editors.



Architecture



Model



Escape Rooms Typologies

Themes (Medieval, Futuristic, Educational, Terror, Adventure, etc...):

- Allow imports of new assets (images, audio, etc)
- Starter kits specialized in different themes
- Templates

Format:

- Discard in-person escape rooms, due to their nature.
- It will not support 3D and VR virtual escape rooms, due to the complexity in development.
- It will focus on covering 2D virtual escape rooms.

Mechanics:

- Point-and-Click
- Drag-and-Drop
- Interactions with text
- Logic Flow
- Multiple Paths
- Online Cooperative Elements

Events - Triggers

Triggers

- click Object_id
- click not Object_id
- Object_id is View_id
- Event_id already happened
- Object_id is in use
- Number seconds have already passed

Events - Actions

Actions:

- Object_id change to View_id
- Object_id goes to inventory
- end of game
- show message Text in Position
- Object_id change size to Size
- Object_id move to Position
- change to Scenario_id
- Object_id is removed
- play Sound_id of Object_id | Scenario_id
- start challenge Challenge_id
- transition Transition_id

Events - Challenges

Challenges:

- Question
- Motion (one object to another)
- Multiple Choice
- Connection
- Sequence
- Puzzle
- Slide Puzzle

Escape Room

```
EscapeRoom(  
    title="Mission Solar System",  
    size=[1280,720],  
    scenarios=[station],  
    events=[try_open,event_open,show_error,exit],  
    transitions=[intro],  
    start=intro  
)
```

View/Sound

```
sound_opening = Sound(source="door-opening.mp3")
door_closed = View.Static(image="door_closed.png")
door_opened = View.Animated(
    images=["door_semi_closed.png", "door_open.png"],
    repetitions=1,
    time_sprite=3
)
```

Object

```
door = Object(  
    initial_view = door_closed,  
    views = [door_closed,door_opened],  
    position = (497,310),  
    size=[286,300],  
    sounds = [sound_opening]  
)
```

Scenario

```
station = Scenario(  
    initial_view = View.Static.background(image="room.png"),  
    views = [background],  
    objects = [door],  
    sounds = []  
)
```

Transition (Telling the story)

Transition(

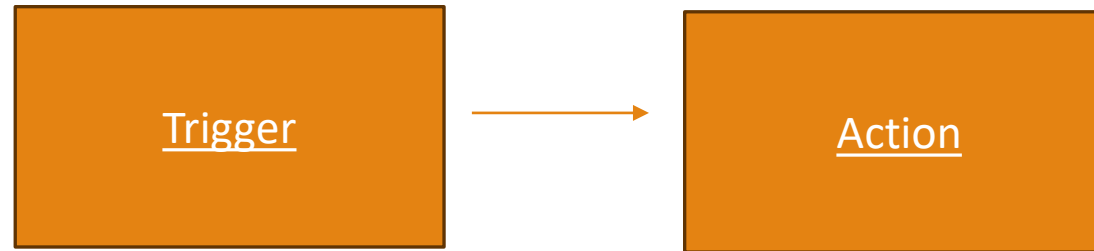
background = View.Static.station(image="station.png"),

music = Sound.music(source="intro.mp3"),

story = "You and your team are astronauts on a special mission to explore the solar system.\n However, during the journey, you get trapped in a space station and need to solve a \nseries of puzzles to escape before time runs out and the spaceship leaves without you."

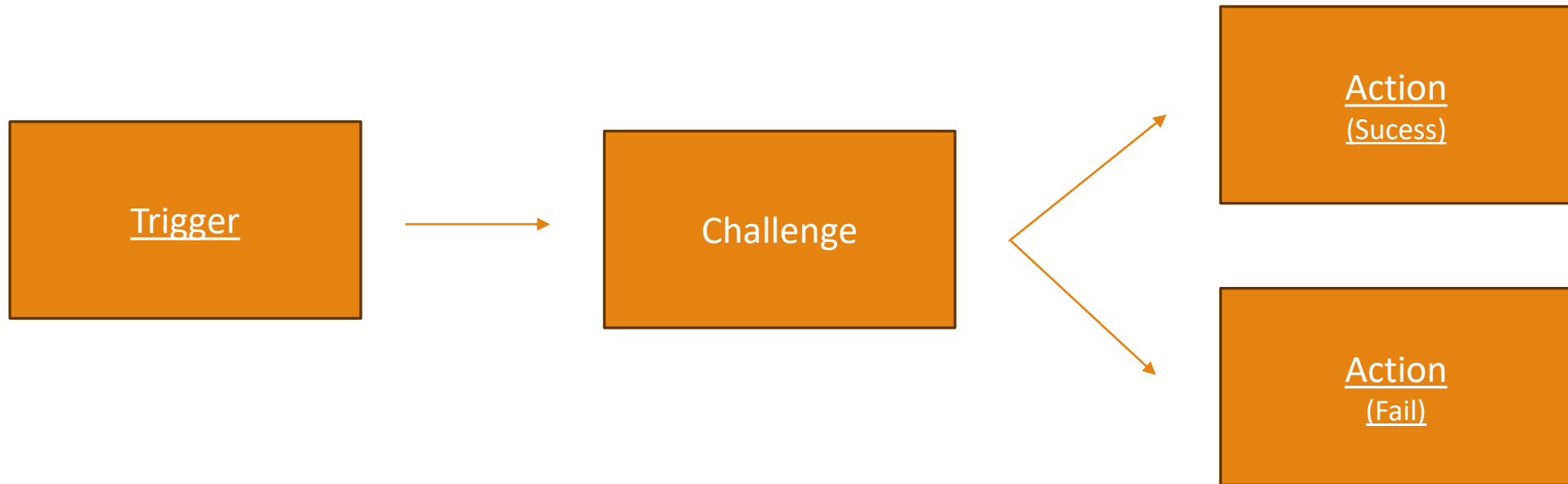
)

Events



exit = Event(if = click door and door is door_opened, then = end of game)

Events



Events

```
event_open = Event(then = door change to door_opened and play sound_opening of door)
```

```
show_error = Event(then = show message "Wrong!" in (497,310))
```

```
question_planets = Challenge.Sequence(
```

```
    question= "Choose from the planet clos...",
```

```
    sequence= ["Mercury","Earth","Jupiter","Neptune"],
```

```
    sucess= event_open,
```

```
    fail=show_error
```

```
)
```

```
try_open = Event(if = click door and door is door_closed, then = start challenge question_planets)
```

Example – Space Station

You and your team are astronauts on a special mission to explore the solar system. However, during the journey, you get trapped in a space station and need to solve a series of puzzles to escape before time runs out and the spaceship leaves without you.





Choose from the planet closest to the sun to the furthest

Earth

Mercury

Jupiter

Neptune

Choose from the planet closest to the sun to the furthest

Earth

Mercury

Jupiter

Neptune

Choose from the planet closest to the sun to the furthest

Earth

Mercury

Jupiter

Neptune

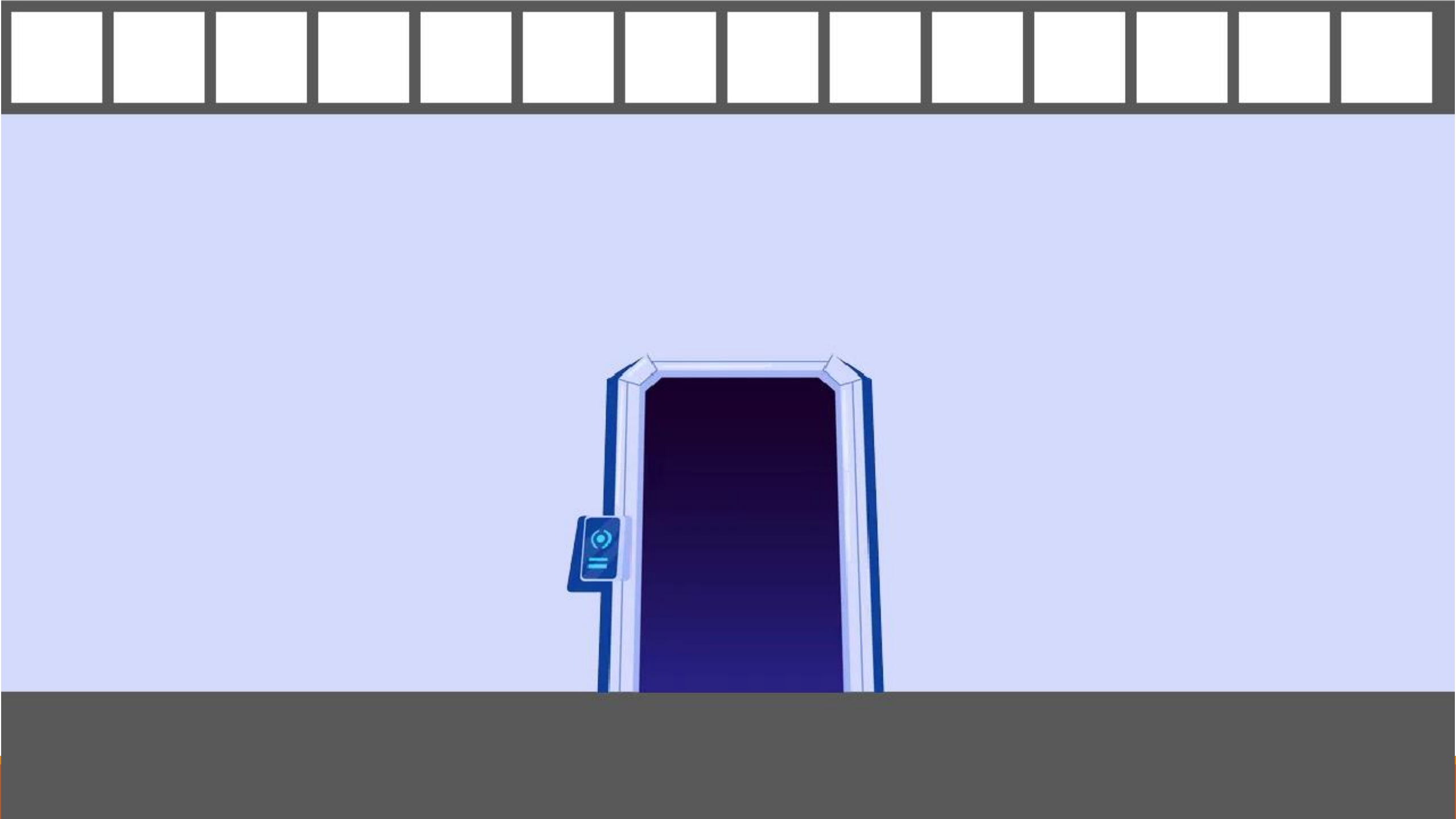
Choose from the planet closest to the sun to the furthest

Earth

Mercury

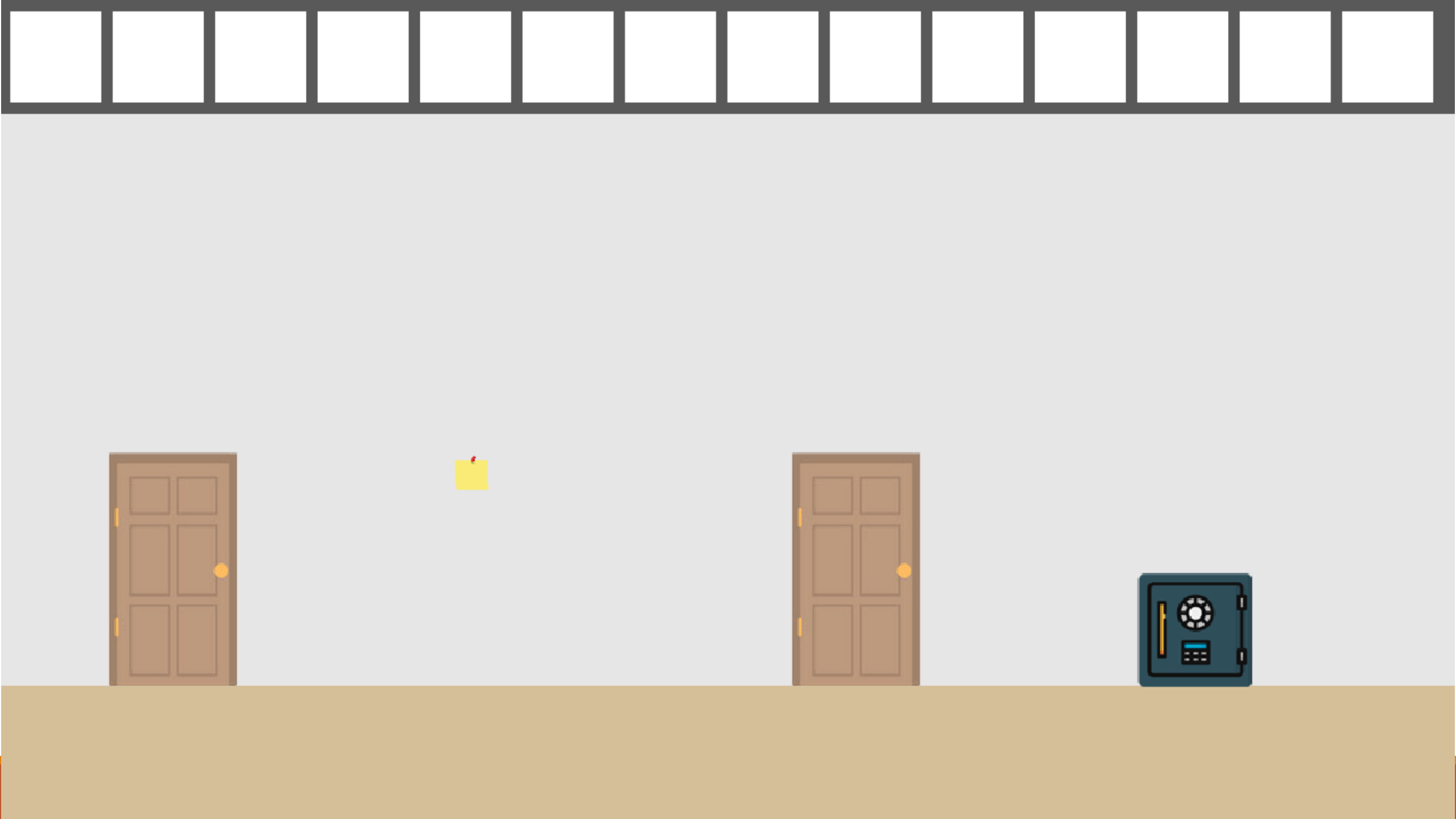
Jupiter

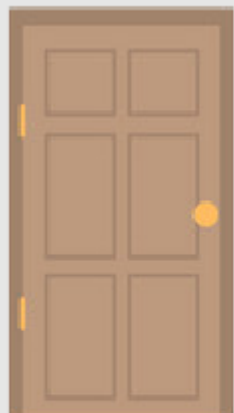
Neptune



Congratulations! You Escaped.

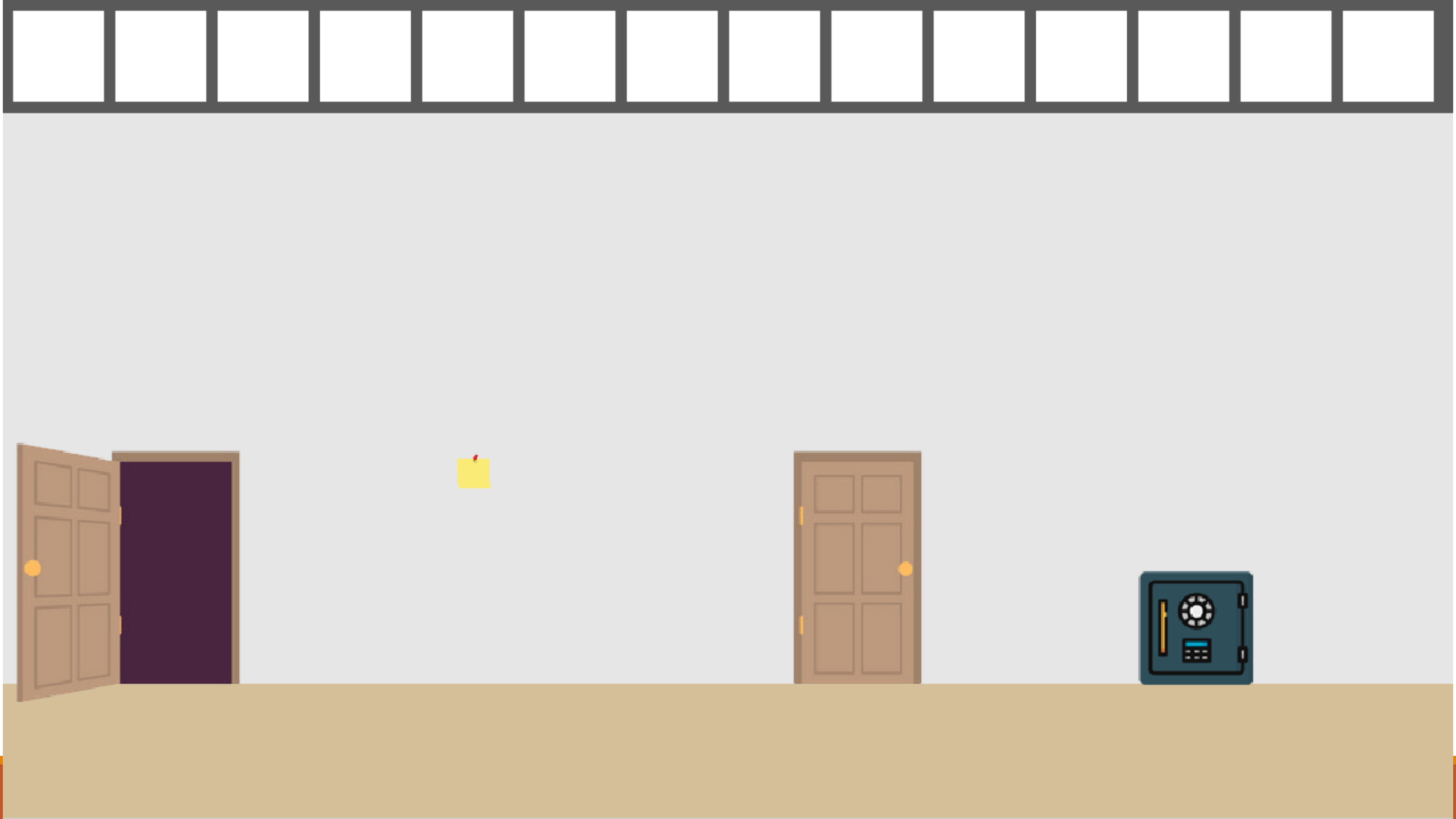
Example – Locked Room

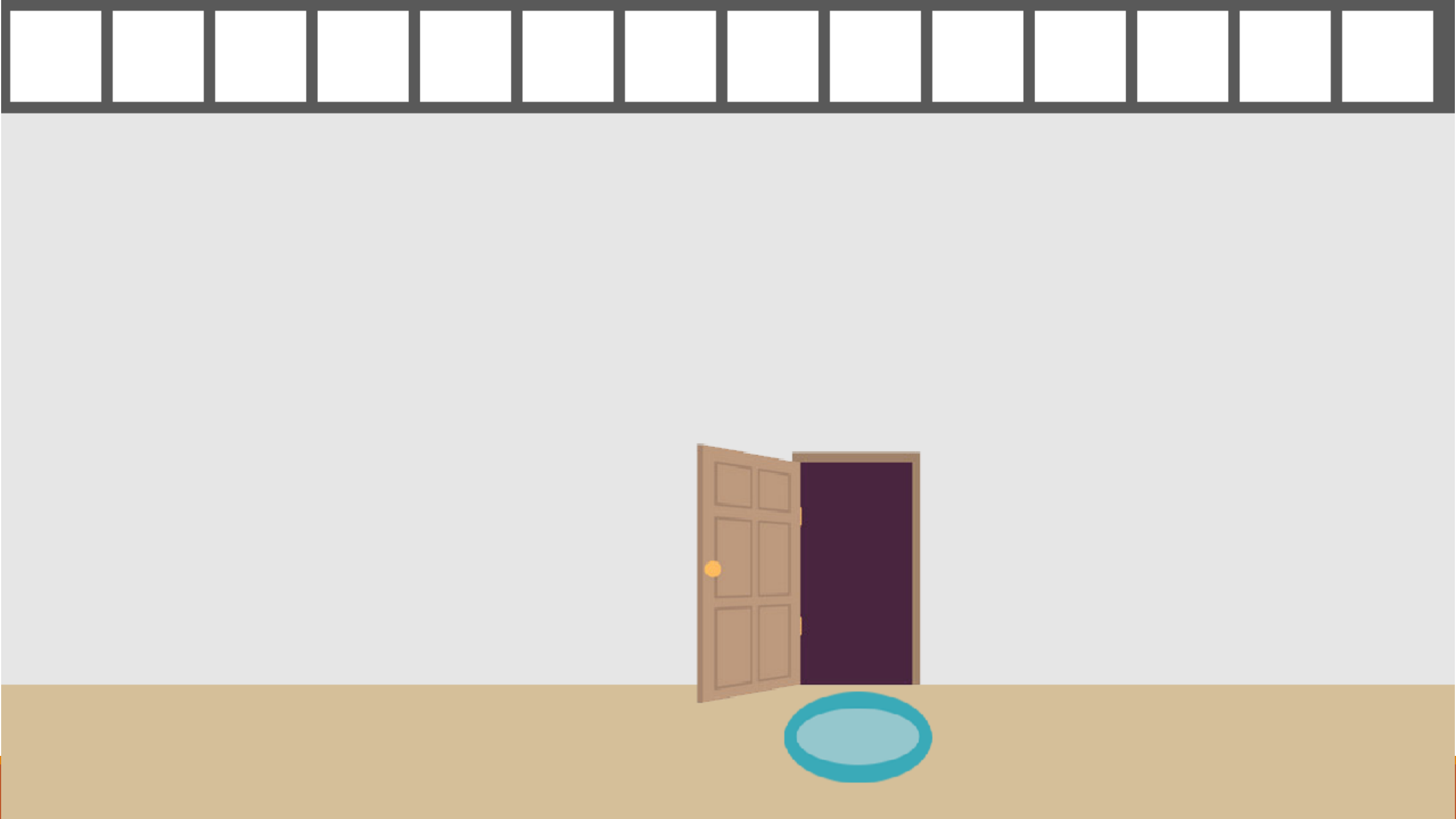


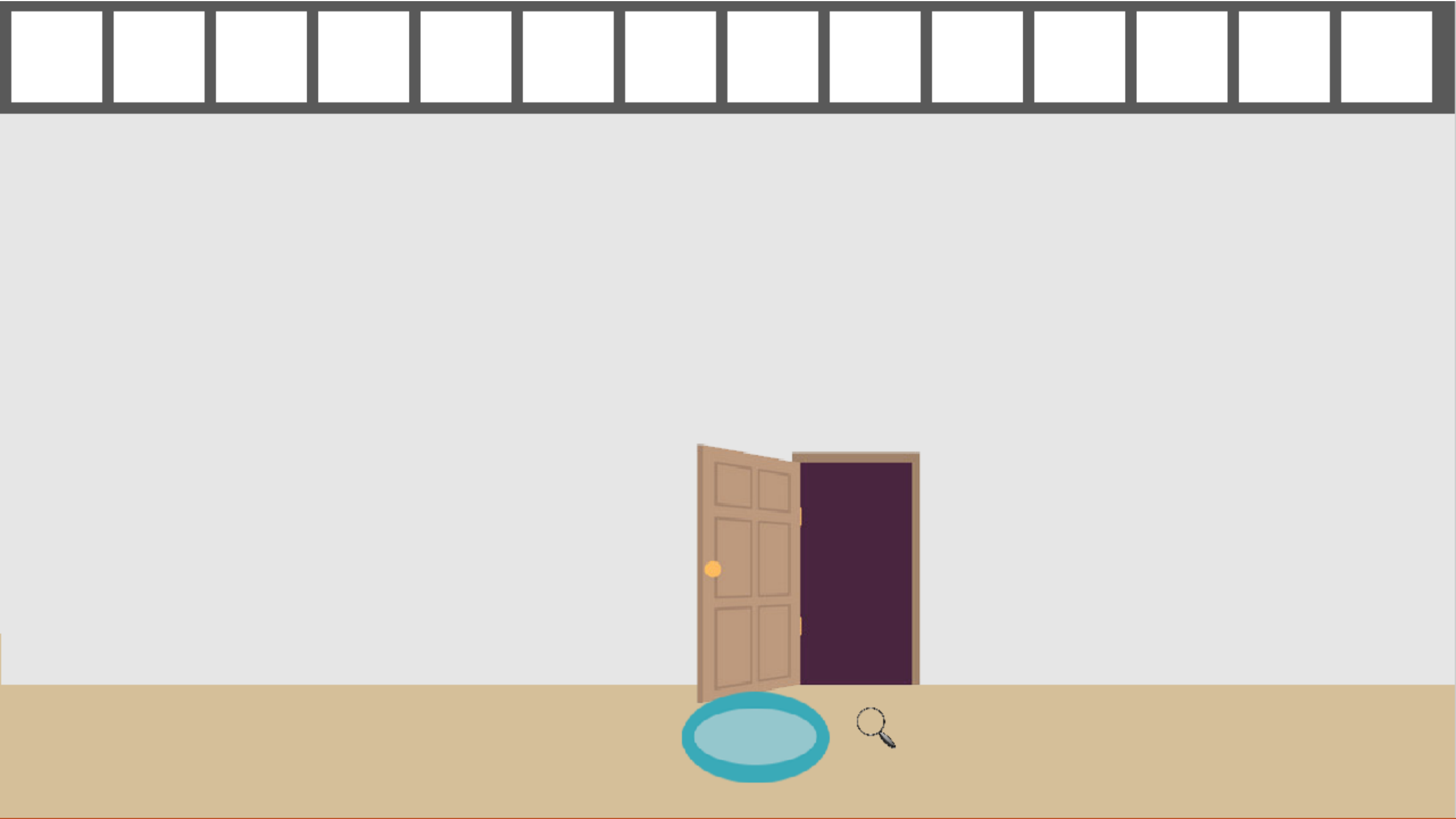


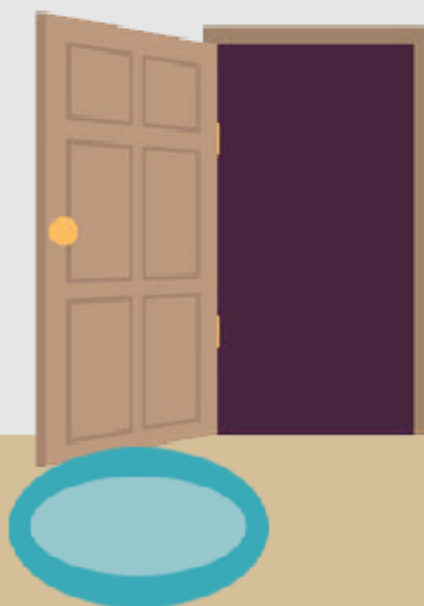
porta trancada! procura uma chave!

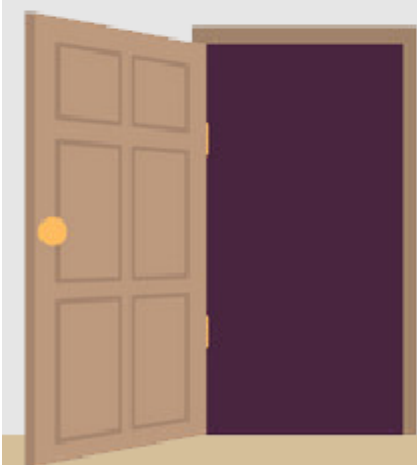














932





Qual é o código do cofre?



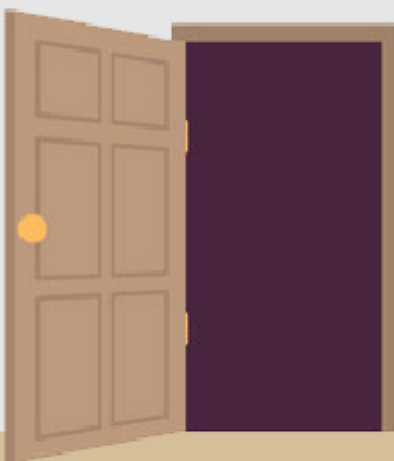


Qual é o código do cofre?

932









Congratulations! You Escaped.

Imports

```
import Object.door
```

```
door1 = Object.door
```

```
door2 = Object.door(position = (100,100))
```

#It is possible to change some parameters of an imported object.

Python Block

```
door_animated_example1 = Python.door
door_animated_example2 = Python.get_images("door")
door_opened = View.Animated(
    images=door_animated_example1, repetitions=1, time_sprite=3
)
```

```
__Python__ #This is python code
import glob
door = glob.glob("../assets/images/door/*.png")
def get_images(object):
    return glob.glob(f"../assets/images/{object}/*.png")
```

Escape Room Block-Based Language (Web ERPL) [Demo]

Classes

Utils

Triggers

Actions

Challenges

Scenario

My Scene

Initial view : initial_view

views :

objects :

View

My View

image :

position :

size :

View

My View

image :

position :

size :

Object

My Object

Initial view : initial_view

views :

Transition

My Transition

story :

view :

Event

My Event

if

do

Event

Escape Room

My Escape Room

scenarios :

Scenario

My Scene

Initial view : initial_view

views :

View

initial_view

image : image room

position : Position (0 , 0)

size : Size (0 , 0)

objects :

Object

My Object

Initial view : initial_view

views :

View

initial_view

image : image door

position : Position (600 , 342)

size : Size (225 , 300)

events :

transitions :

Convert

Exportar Blocos

Escolher ficheiro

Nenhum ficheiro selecionado

Project Links

Project Github: github.com/surumkata/ERPL

Pypi Project Page: pypi.org/project/erpl/

Web ERPL Demo: surumkata.github.io/weberpl/

Conclusion

Future Work:

- Enhance the engine to improve performance and add new features.
- Introduce a variety of challenges to enrich player experiences.
- Develop a web engine and block-based language to expand the reach of virtual escape rooms.
- Create usable templates and scripts to streamline development processes.
- Implement a graphical interface that integrates with the language to simplify escape room creation.

Questions and Sugestions