



CES-22

**Alunos:**

Bruno de Paula Ferreira

Gabriel Ribeiro Pardini

Brendon Diniz Borck

**Turma COMP24**

**Instituto Tecnológico de Aeronáutica - ITA**

# 1 Descrição do Projeto

Jogos de escape room tratam-se de salas reais, temáticas e imersivas, contendo vários enigmas e desafios palpáveis, onde o jogador é o personagem e precisa escapar em determinado período de tempo.

Mystery House trata-se, portanto, de um jogo de escape room virtual, no qual o jogador deve escapar de diferentes salas constituídas por diversos enigmas. O projeto consiste em criar um jogo 2D, com visão de cima, no qual o personagem acorda em um quarto sem memórias de como chegou nesse local. O objetivo do personagem é escapar das salas em que está preso por meio da resolução de enigmas e entender quem ele é e qual a sua relação com um certo crime. O jogo será composto por alguns níveis distintos, cujas dificuldades serão elevadas com o progresso do personagem.



Figura 1: Ambiente inicial do jogo

O jogo possui um enredo elaborado pelos elementos do grupo, e para a resolução dos enigmas é necessário desvendar toda a história que se passa, sem o completo entendimento do enredo não é possível se concluir Mystery House. Por muitas vezes, haverá textos enigmáticos mas que contribuem para entender como tudo está ambientado.

O jogo tem 5 salas distintas e cada sala é diferente da anterior. Um fato interessante de se mencionar é que algumas vezes é necessário utilizar informações de salas anteriores para resolver um enigma de uma sala, portanto recomenda-se anotar uma informação que "não faça sentido", pois no futuro ela pode ser utilizada.

Para a composição dos cenários, foi utilizada algumas imagens da internet sem direitos autorais, as imagens que contenham direito encontram-se referenciadas. Adicionalmente a isso foi realizado a criação de muitas outras figuras que foram incluídas no ambiente.

## 2 Arquitetura

Primeiramente, definimos uma classe **Game**, na qual irá gerir todo o funcionamento do jogo. Todas as outras classes terão algum tipo de relação com ela. Podemos considerar que seria uma classe mestre para todo o jogo. Nessa classe **Game**, há vários métodos que irão realizar a execução do jogo em si, como o método **rungame** na qual irá conter todo o loop principal do jogo. Além disso, há um método que verifica a tecla pressionada (watchevents) pelo jogador e realiza alguma ação, há também outros métodos acessórios que auxiliam todo o funcionamento do jogo.

Além disso, há uma classe **Player** que é a classe do personagem principal do jogo. No método **rungame** é criado um objeto dessa classe, logo as duas classes possuem uma associação do tipo possui, um Game possui um Player. Além disso, há uma classe **Room** que irá representar a sala em questão que o personagem está que também é instanciada um objeto dentro da classe **Game**.

Dentro da classe **Room** há uma série de objetos que a compõem, como uma porta, uma cama, um relógio, uma mesa, um quadro entre outros. Cada um tem uma classe específica e todos são instanciados dentro da classe **Room**. Além disso, foi criada uma classe mãe para todos os móveis que compõem a sala, essa classe se chama **InteractiveObjectcs**, essa classe define alguns padrões de interação entre personagem e objeto, portanto todos os móveis herdam de **InteractiveObjectcs**. Todas essas relações podem ser visualizadas no diagrama de classes simplificado da Figura 2

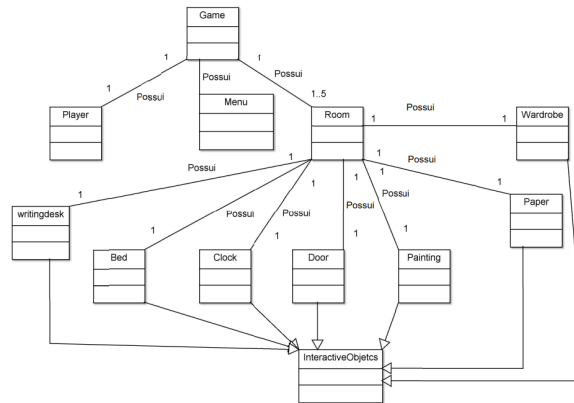


Figura 2: Diagrama de classes simplificado

Esse diagrama de classes é simplificado apenas com as relações principais entre os componentes. Um diagrama completo ficaria muito grande e poderia ficar confuso e perder a sua funcionalidade.

### 3 Explicação dos Arquivos Entregues

No GitHub, há uma série de arquivos que serão usados para o funcionamento do código:

- min.py: Arquivo principal na qual há o funcionamento do jogo.
- menu.py: Arquivo secundário que realiza a montagem do menu.
- objects.py: Há a classe InteractiveObjectcs.
- paintings.py: Há a classe para a criação dos quadros que tem na sala.
- papers.py: Há a classe para a criação dos pergaminhos que tem na sala.
- player.py: Há a classe do jogador.
- room.py: Há a classe das salas.
- utils.py: É um arquivo acessório com algumas funções utilitárias para todo o código.
- wardrobe.py: Há a classe do armário que tem na sala
- writingdesk.py: Há a classe da mesa de escrever que tem na sala.
- bed.py: Há a classe da cama que tem na sala.
- clock.py: Há a classe do relógio que tem na sala.
- constants.py: É um arquivo que guarda algumas constantes utilizadas no código.
- doors.py: Há a classe das portas que tem na sala.

Além disso, há uma pasta assets na qual há as imagens de todos os móveis que serão utilizadas na sala.

### Referências

Imagem Menu: Disponível em <<https://www.gigamic.com/jeu/mystery-house>>

Cenário 1: Disponível em <<https://www.deviantart.com/thetophatturtles/favourites?page=56>>