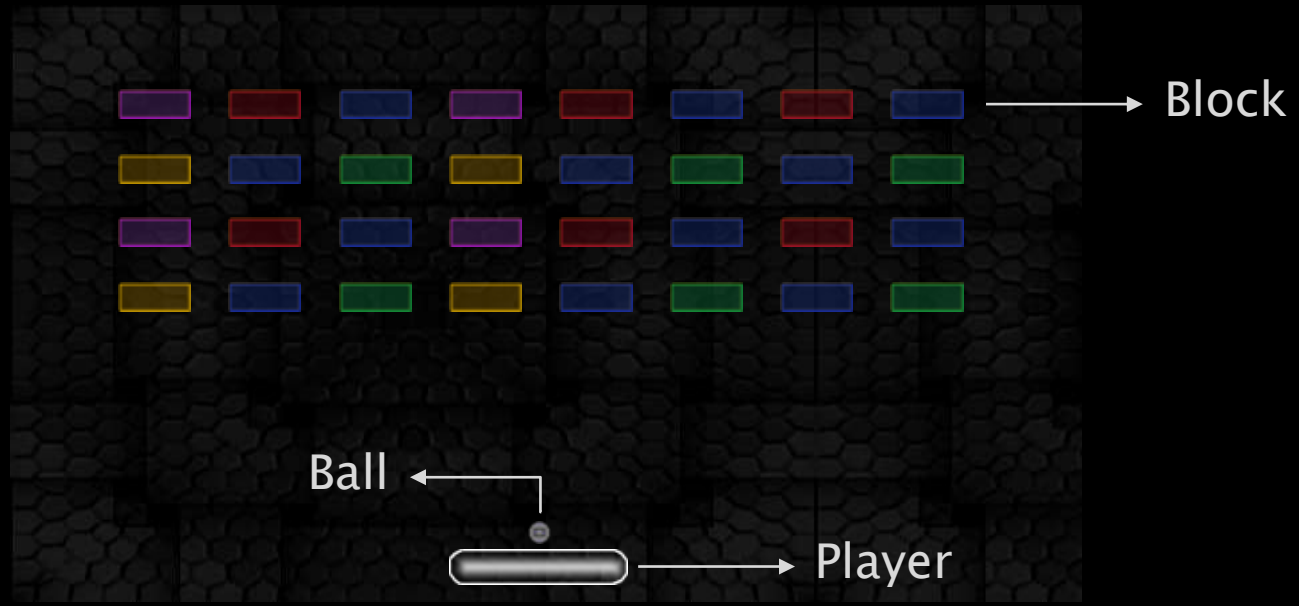
Gerenciador de Cena

Programação de Jogos

Judson Santos Santiago

Introdução

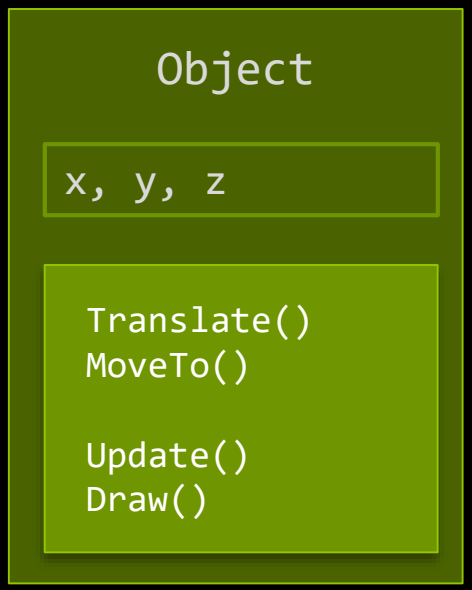
- Um jogo é composto por um grande número de objetos que precisam ser atualizados e desenhados a cada quadro



Objetos do Jogo

- Criar uma classe base para representar objetos de um jogo simplifica a sua construção:
  - Todos os objetos são obrigados a implementar métodos para atualização e desenho do objeto
  - Possibilita tratar os objetos de forma genérica:
    - Listas de objetos podem ser criadas e tratadas independentemente das particularidades do objeto

    Ex.: Player, Ball e Block



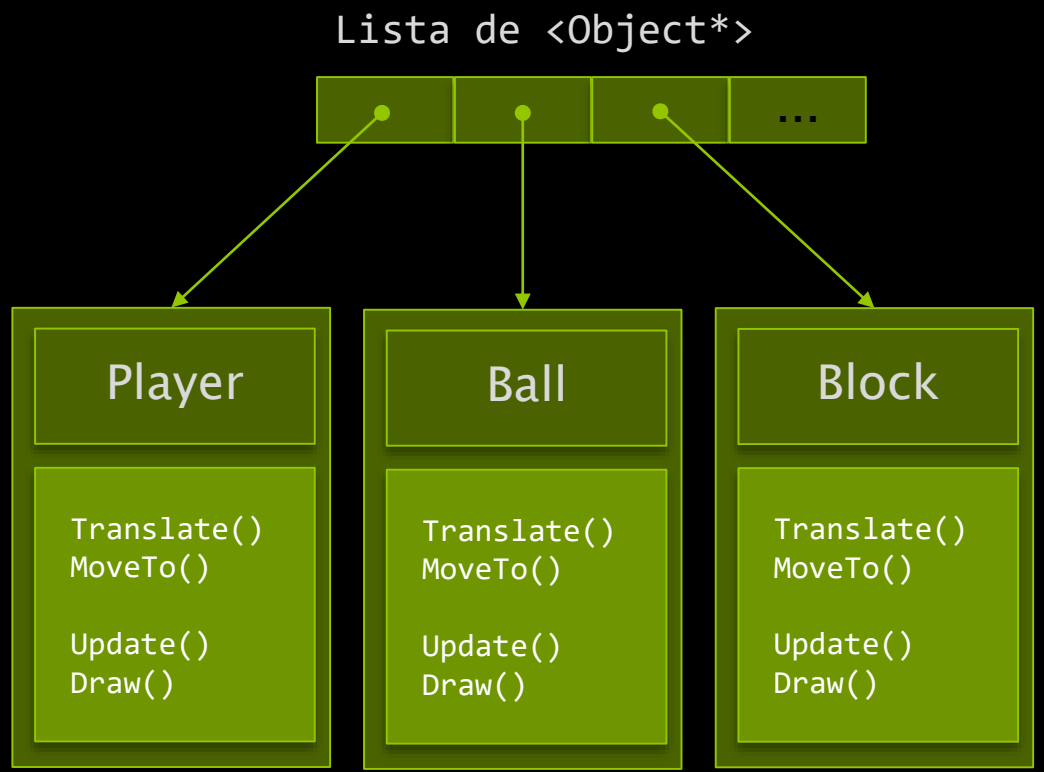
Coleção de Objetos

- Se todos os objetos derivam da classe Object, podemos criar coleções de objetos:

```
// lista de objetos
list<Object*> objects;

// adiciona objetos na lista
objects.push_back(new Block());
objects.push_back(new Player());
objects.push_back(new Ball());

for (auto i : objects)
   i->Update();
```



Gerenciador de Cena

- Os jogos são compostos por coleções de objetos
  - Para simplificar as operações comumente realizadas sobre estas coleções, podemos criar um gerenciador de cena:
    - Encapsula a estrutura de dados utilizada para organizar a coleção de objetos
    - Um jogo pode ser composto por várias cenas
    - Uma cena é composta por vários objetos

```
class Scene
{
private:
    list <Object*> objects;

public:
    Scene();
    ~Scene();
    void Add(Object * obj);
    void Update();
    void Draw();
};
```

Resumo

- Um gerenciador de cena pode acelerar o processo de adicionar, atualizar e desenhar objetos do jogo
  - Um gerenciador de cena permite:
    - Adicionar e remover objetos da cena
    - Atualizar e desenhar a cena
    - Calcular colisão entre objetos
    - Ordenar objetos por algum critério:
      - Profundidade
      - Textura
      - Tipo

```
class Scene
{
private:
    list <Object*> objects;

public:
    Scene();
    ~Scene();

    void Add(Object * obj);
    void Update();
    void Draw();
};
```