

## The IGame interface

The arcade core interacts with an IGame interface. This interface is composed of the following member functions.

```
class arc::IGame {
public:
    struct Specs {
        int x;
        int y;
        uint pixelStep;
        uint fps;
    };

    virtual ~IGame() = default;
    virtual ItemList &getItems() noexcept = 0;
    virtual const Specs &getSpecs() const noexcept = 0;
    virtual void processInteraction(Interaction &) noexcept = 0;
    virtual void envUpdate() noexcept = 0;
};
```

*Note : used data types are documented in the "arcade\_data\_types" manual.*

## Methods description

`ItemList &getItems() noexcept;`

Returns all the items that need to be displayed so each of them can be passed by the arcade core to the display library.

---

`const Specs &getSpecs() const noexcept;`

Returns the specifications of the game including the board size, the cell size and the number of frames/cycles per second it is supposed to operate at.

---

`void processInteraction(Interaction &) noexcept;`

It is used to process the interactions revived from the display library. For instance, if the interactions is a movement or a shoot instruction, the game should react accordingly and update its items.

---

`void envUpdate() noexcept;`

The core call this method after each cycle to let the game process anything that is not triggered directly by a player interaction. For instance, it is the moment for the game to update the positions of the NPC or to spawn new droppable objects.