Avatar(AvatarState)

An avatar is a state that corresponds to a character, containing their name, level, and more.

- AvatarState

State

- Account Address: Addresses.Avatar
- State Address: The avatar's address is derived from the agent's address.

```
public Address GetAvatarAddress(Address agentAddress, int index)
{
    return agentAddress.Derive($"avatar-state-{index}");
    // or
    return Addresses.GetAvatarAddress(agentAddress, index);
}
```

Get State:

```
public AvatarState? GetAvatarState(IWorld world, Address address)
{
    IAccount account = world.GetAccount(Addresses.Avatar);
    if (account is null)
    {
        return null;
    }

    IValue state = account.GetState(address);
    return state switch
    {
        Bencodex.Types.List l => new AvatarState(l),
        Bencodex.Types.Dictionary d => new AvatarState(d),
        _ => null,
    };
}
```

Inventory

Inventories exist for each avatar and contains a variety of items.

- Inventory

State

- Account Address: Addresses.Inventory
- State Address: Use the address of avatar as it is.

Get State:

```
public Inventory? GetInventory(IWorld world, Address address)
{
    IAccount account = world.GetAccount(Addresses.Inventory);
    if (account is null)
    {
        return null;
    }

    IValue state = account.GetState(address);
    return state switch
    {
        Bencodex.Types.List l => new Inventory(l),
        _ => null,
    };
}
```

QuestList

QuestList exists for each avatar and contains the avatar's quest information.

- QuestList

State

- Account Address: Addresses.QuestList
- State Address: Use the address of your avatar as it is.

Get State:

```
public QuestList? GetQuestList(IWorld world, Address address)
{
    IAccount account = world.GetAccount(Addresses.QuestList);
    if (account is null)
    {
        return null;
    }

    IValue state = account.GetState(address);
    return state switch
    {
        Bencodex.Types.List l => new QuestList(l),
        Bencodex.Types.Dictionary d => new QuestList(d),
        _ => null,
    };
}
```

WorldInformation

WorldInformation exists for each avatar and contains information about the avatar's adventures.

- WorldInformation

State

- Account Address: Addresses.WorldInformation
- State Address: Use the address of your avatar as it is.

Get State:

```
public WorldInformation? GetWorldInformation(IWorld world, Address address)
{
    IAccount account = world.GetAccount(Addresses.WorldInformation);
    if (account is null)
    {
        return null;
    }

    IValue state = account.GetState(address);
    return state switch
    {
        Bencodex.Types.Dictionary d => new WorldInformation(d),
        _ => null,
    };
}
```