**Resource Management Systems**

Harrison Francis-Lyster - Sep 25th

| **#** | **Description** | **Author** | **Date** |
| --- | --- | --- | --- |
| 0.1 | Initial revision | HFL | 25-Sep-2023 |

| **Table of Contents** |
| --- |
| [Goals](#_77am1vxdp3za)  [Dependencies](#_ua71e8z15zik)  [Building Systems](#_kapwshxuajkw)  [Plan:](#_qvvex77x2rs)  [NPC AI Systems](#_9oniix4zpiod)  [Plan:](#_u4x7ip6rp5h3)  [Dynamic UI Systems & Data Management Systems](#_iess6rlf9o8n)  [Plan:](#_8xdxe2ykx5o2)  [Requirements](#_jm4ce76ganwt)  [The Resource Pool](#_f1lr45tgc1gm)  [Resource Objects](#_hepo69zgub31)  [Map Generation](#_odk6mpvtyssy)  [The Resource Pool](#_z9btdgd6358s)  [Resource Objects](#_m4p4ee6c0zdu)  [Map Generation](#_471w7t59tlh3)  [Random Generation](#_9a1a6llebpdw)  [Hand Placed](#_q1603cvzlcoh)  [See Also](#_omxloiqvood5) |

# **Goals**

Any good RTS needs a robust system for resources. The main goal is to make a system that can handle gaining resources from buildings and from the environment, and spend said resources building structures and recruiting units.

Key goals include:

1. Making a resource “pool” where all the resources that the player gains are stored.
2. Interfacing with units and buildings to gather resources.
3. Having an easy way to spend resources.
4. Map generation, where resources spawn in the map.

# **Dependencies**

## Building Systems

This feature heavily overlaps with building systems as we plan to have the game played on a grid and as such the resources will have to fit with that grid.

### Plan:

* Work closely with building systems to ensure a unified goal
* Create ways to easily interface with systems

## NPC AI Systems

Both enemy AI and player units have to be able to work with this system.

### Plan:

* Create a universal system that can be attached to either a player or enemy controller
* Create ways to easily interface with systems

## Dynamic UI Systems & Data Management Systems

These are probably the easiest to create functionality for as they only need access to the values and to set the values of the resource pool

### Plan:

* Create ways to easily access for both getting and setting the resource pool

# **Requirements**

*("What" -- clarify what must be delivered to successfully meet the goals, as well as any additional constraints and requirements as appropriate.)*

## The Resource Pool

The resource pool is fairly straightforward, all it really needs is storage for how many resources the player/AI has.

## Resource Objects

This needs to be flexible to work with any type of resource.

## Map Generation

This could come in two forms: random generation or hand placed resources.

## 

**Technical Approach**

*("How" - what is your proposed strategy for meeting the requirements? Provide enough detail to evaluate the approach. Need not be a full-on technical design, but should be detailed enough to either start development or guide a detailed technical design when needed.)*

## The Resource Pool

The resource pool is a fairly straight forward script that all it really needs is a bunch of variables and ways to influence and check those variables. It would be attached to the player/AI objects.

## Resource Objects

The script will be fairly straightforward, how many resources the object has and what can gather resources from it, i.e. specific units or buildings.

## Map Generation

### Random Generation

This system would use perlin noise to generate where the resources are placed. It would also determine where players should be able to start from.

### Hand Placed

For a hand placed system I would make tools that would make it easy for a level designer to place resources in the map, and where players can spawn.

# **See Also**

*(Further reading and reference material as appropriate. Link external web sites, internal docs, etc. with a brief description to explain why it's relevant.)*