\_\_*Harmony Restored*\_\_ GDD

## 

## 1. Game Overview

* **Title:** \_\_\_
* **Genre:** 2D Isometric, Clicker, World-building
* **Platform:** PC, WebGL, Itch.io, [Optionally steam]
* **Development:** 10 days
* **Target:** Pirate Software - Game Jam 15 (2024)

## 2. Game Concept

**\_\_\_** is a clicker automation game in 2D isometric which turns into an alchemy-based worldbuilding game.

The game revolve around 2 worlds, one in the light, another in the shadows, which must combine their resource and craft to produce a final set of artefacts that will help them reunite.

## 3. Gameplay Mechanics

### Core Gameplay Loop

The player collects resources (manually, then automatically), builds and places craftsman buildings, discovers and produces new materials by material fusion in the craftsman buildings, combining the strength of the two worlds, until he is able to gather enough of the final resource to end and conclude the game.

### **Player controls**

* The player can click (LMB) and maintain on resource to collect them
* The player can drag and drop (LMB) resources from his inventory in various slots in the scene
* The player can switch between the two worlds at any time by pressing SPACE
* The player can build various buildings (mostly craftsman) on a grid in both worlds. Each building takes several cells of the grid (ex: 2x3 or 4x2…)
* The player can select (LMB) a building to display the capabilities it offers
* The player can create creatures in a dedicated portal building
* The player can test resources combinations in an alchemy-dedicated window (or in craftsman buildings)
* The player can drag and drop resources and fuse them in craftsman buildings to obtain a new one
* The player can drag and drop creatures in the craftsman buildings to perform the resources transformations

### Game Objectives

The player must discover and produce enough of the two final resources to end the game. The requirement in term of final resource depends on the number and type (Light/shadow) of creatures he created to help him in his task.

[Optional-1] The game can be timed, either with real-time timer or with game-time timer. These are different: real-time timer continue to run during any action of the player, which emphasizes on the speed of the UI action and planning, while game-time timer only considers the simulation time of the game to reach the end of the game, without considering the pauses, the navigation in the menus, and so on. This second version should be more appropriate for the game.

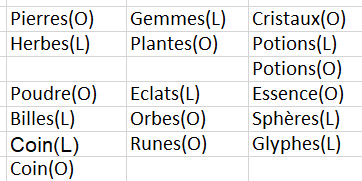
[Optional] The end game could be enriched with the necessity to build a big marvel structure dedicated to the player, but this ending should result in the player giving up to let the creatures leave in peace instead of working all day for nothing.

## 4. Game World

The game should consist of a single scene with an overworld with grass mostly, visible in isometric view. A grid of cells is visible on the ground as well. The switch between worlds should reveal more shadowy and darker ground. There should be dark stone resources in the shadow world and grass light resources in the light world. The buildings can be put on the grid, on either side of the world. In the same world, the buildings block each other, but not through worlds. Or on the opposite, the buildings could block each other across worlds (To be confirmed with playtest).

### Resources

The table of resources and crafting recipes is defined in an excel file.

The first version of resources types looks like that:  


The combinations of resources is defined as:

Gemmes = Pierres + Billes

Plantes = Herbes + Symboles

Eclats = Pierres + Poudre

Orbes = Billes + Poudre

Runes = Symboles + Pierres

Cristaux = Gemmes + Eclats

Potions = Plantes + Orbes

Essence = Eclats + Runes

Sphères = Orbes + Gemmes

Glyphes = Runes + Eclats

### Buildings

The list of buildings is the following:

* PORTALS (creature creation to be paid with a few base resources)
* ALCHEMY building (find recipes by trying combinations of resources)
* JOAILLIER (Jeweler) Pierres, Gemmes, Cristaux
* HERBORISTE (Herbalist) Herbes, Plantes, Potions(N
* MAGE (Mage) Poudre, Eclats, Essence
* VERRIER (Glassmaker) Billes, Orbes, Sphères
* SCRIBE (scribe) Symboles/Coins, Runes, Glyphes
* CARRIERES ou MINES pour recoller des pierres automatiquement
* SERRE (Greenhouse) pour faire pousser des herbes de facon automatique
* SCIERIE pour le bois? Si nécessaire.

## 6. Art and Graphics

The visual style of the game should be 2D isometric, in a medieval style (with medieval craftsman buildings). There should fairly simple and generic UI display for the crafting of resources, to allow for UI reuse between buildings, just changing the title of the building and a few visual clues for instance.The resources should be always accessible in each world, maybe with a dedicated window always visible on the right of the game, to allow for an easy drag and drop mechanic. Either:

* All the resources from both worlds are available directly to drag and drop
* The player only sees the resources of current world and has to switch between worlds to see the corresponding available resources.

### Sprites and animation

Buildings: There should be two overall backgrounds, one for each world with a grid on top. There should be dedicated sprites for each building, in both worlds (some buildings exist only in 1 world or the other).

**Visual Style:** [Description of the game's art style and visual direction]

* **Sprites and Animations:** [Details on character sprites, animations, and effects]
* **UI Design:** [Overview of the user interface and HUD elements]

## 7. Audio

Sound effects should support any aspect of the gameplay, from UI clicks to creatures actions, etc…

2 medieval music scores should run in parallel, and be dynamically changed depending on the current world display. One should be more quiet and simple happy than the other one, sad and slightly creepy.

## 8. Technical Specifications

**Unity** will be used for development, as well as Aseprite and Krita for sprites.

## 9. Monetization Strategy

**None.** Free game, except if we end up completing it in an interesting way and pushing it on Steam at the end. In this case the game should be cheap and accessible, not more than 5 euros, with no micro-transaction or any in-game purchase of any type and no ads.

## 10. Schedule

The game should be finished by the 26th of July (in a week).   
The deadline of the jam is the 31th july.

### Preliminary schedule

17-19 : GDD

19-21 : Draft assets and core gameplay loop

21-23 : UI, additional features and playtest

23-25 : Bugfixes, feature completion and final assets

25-26 : Playtest, Bug fixes and Publish first version

26-31 : Playtest, Bug fixes.

## 11. Team and Roles

**Draelent:** Code, art and music  
**Speufy:** Code, playtest and bug fixes  
**Gummy:** Code, playtest and bug fixes

**Others:** Playtests, feedback