# Deep Mine Design Document

2/3/2020

# Overview

*Build an empire and explore a vast fantasy world…deep underground*.

Phantom, the adder, has enslaved the world with his dark magical forces of undead minions, and poison jungles. You must build a base and travel deep underneath the crust of the world to destroy Phantom, and create world peace.

# Story

You live on a planet called Soltar, Soltar is at the edge of the universe, and it is an infinite planet which has a life force controlled by the monsters that inhabit it. Soltar’s crust is infinitely deep, and as you trek through the game you find that the crust gets deeper, and more bizarre as you work your way through the world.

You are an archaeologist in search for biological mechs.

The theme is a modified high fantasy setting. Humans, Orcs, Elves, Dwarfs. You are given the option of having any of these guys in your base.

This game is like Dwarf Fortress. You are given farmers, peons, and warriors, and you must build your fortress, and fight against enemies, in order to reach “Deathstone fortress” and defeat Phantom, the boss of the game.

Gameplay 1000ft.

The game is basically one of “Down-building.” You build into different layers of the world. Each layer contains different strengths, abilities, and enemies. The peeps that you raise in those layers have stats in them, based on the terrain blocks that you find in those layers. (ex. You are in the Fright layer of the world, and have a Ghast peep, the Ghast would draw power from the fright layer because that’s its habitat).

**Character Building** – making your peeps more powerful.

**Exploration** – Exploring the deep layers of the world and destroy Phantom.

**Defense** – Defend your base against enemy monsters that want your food.

**World-Building** – Building your base higher and making it more secure, in the wake of the enemies whom want to destroy it.

# Peeps

The peeps are the inhabitants of your world, and they are the driving force behind your army. You receive 2 peeps when you start the game. When you run out of peeps you lose the game (die) so keeping your peeps alive is of vital importance. Also, Peeps pay you taxes, and with taxes you’re able to purchase to build new things in the game world. There are multiple kinds of peeps and multiple races. Overall, the base peep is the beginning peep that can evolve into a subset of different peeps. The evolutions of peeps range from 1-4. The 4th evolution of a peep is a temporary evolution and may kill the peep if it achieves this too long.

Peeps evolve into just about anything, and you can make the peep monsters become other things as well. For example:

Spirit Form – a form of spirit that you can evolve people into.

Micron -> Zephyr -> Chronis ->– Three stage evolution of the Zephyr.

Clay Dragonis ->Gelded Dragonis -> Diamond Dragonis -> Three stages of the dragonis evolution.

Transforms happen during battle, when the peeps are fighting against each-other. They happen when a peep has reached a certain level of experience points. Experience points are gained by defeating other peeps. The XP gained by a peep is based on the Class of the peep, and the Level that that peep is at. The (Class \* Level). This is the class multiplier. For a weak peep the class may be, like 0.3.

Peeps are destined to become skilled at what they do, likewise you upgrade their skills by putting them to work on tasks that you find necessary for the fortitude of your base. Making the peeps hone these skills makes them more powerful.

You can put peeps to work attacking each other to gain more experience, however you will only gain the most experience attacking enemy peeps, and the experience gained by attacking your own guys through training converges to zero. This is a limitation to incite players to leave their base.

*Peep Skills*

All peeps are basically the same, in the sense that they all can level up, they all can evolve, and they all can equip things. They have different **images**, and different **stats**, and different **skills**. The skills are

**Battle** – Warriors are the main kinds of peeps. They fight other peeps, and they can be equipped with weapons and armor.

**Farming** – Farmer peeps plant crops on grass and soil types of blocks.

**Breeding** – Breeder peeps are used to make new peeps. The purpose of breeder peeps is to eat and reproduce.

*Peep Types*

**Peep** types are largely dependent on the various ecosystems that peeps tend to do well.

**Spirit** – Have high intelligence and Special attack power.

**Fighting** – Have high strength.

**Water** – Can move quickly in water, and draw more power from water terrain.

**Sand** – Move quickly in sand, and draw more powerful from sand terrain.

*Peep Races*

**Human** – Humans are what they are. They fight and die. Humans transform into Zephyrs and then into Angels.

**Orb, Ghast** – Ghast peeps are transparent and immune to physical attack. Orbs transform into Ghasts.

**Dragonis** – Dragon-humans. They transform 2 times.

**Machon** – Machons are peeps that are bred for battle. They raise in stats very quickly. They can evolve into Great Machons, which have more power.

**Scrab** – Scorpion humans types that stab to attack. They transform into Scuts (cut).

Equipment capabilities change when peeps evolve. Sometimes they get new equipment slots, but sometimes the equipment slots disappear.

# Game World

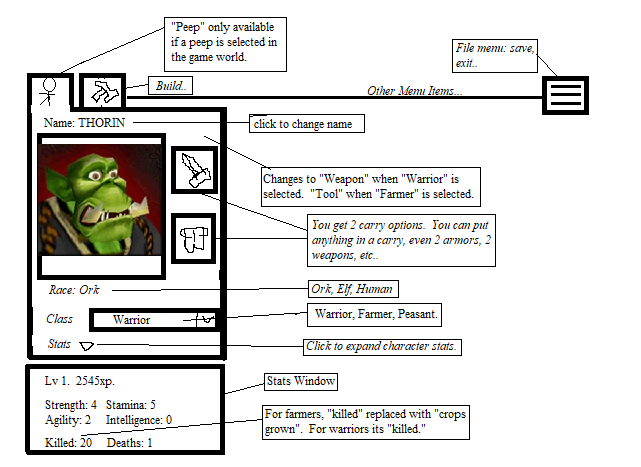
The game worlds are organized with different ground layers. Each layer of the world is a different game level, or world. The levels are themed like you would see in Mario, or Metroid (Metroid is our focus). The themes can range from just about anything we can imagine.

The view of the game world underground is simply one layered view, or two, etc. The layers above and below can be toggled in the menu at the top of the screen.

## World Building

The world editor will be like “legos” where, you’re able to place little blocks in the world that have preconfigured shapes. The world is a collection of density points, which are dependent on cube-like textures. The cubes are marching voxels, but they have slopes that allow us to create pathways for the peeps.

1. Open up the edit menu to enter edit mode.
2. Click a block to place it. Block will show transparent in the game world.
3. To orient the block, Click and hold the RIGHT mouse button then drag the mouse across screen to rotate the block. OR click and drag the rotation slider.
4. To place the block, click the left mouse button.



There are only two fundamental bricks. Slopes, and Blocks. Additionally, we may give the user the ability to add density points at will, however, the only *necessary* bricks in the game are slopes, and blocks. Slopes allow you to carve pathways up and down the world for your peeps to move along. Blocks are just structural material. In addition to these two fundamental components, there are also decoration blocks. Those are blocks that have block collision topology, but they have some decoration. They can be anything (sconces for instance). They are also items.

Bricks are collections of these points (like Minecraft blocks, but with slopes). You get the money from the people in your world paying you taxes. So really, you’re like a government official. While you build things in your world to make it bigger, you’re attacked by a series of enemies whom you must fight off. Enemies can be anything. Probably, zombies, though. You fight enemies by training troops and building various military tech. Military tech is the primary “new content” portion of this game, in addition to new weapons, ores, and treasures. Military tech evolves over time.

## Objects

You place objects for decoration, or for functionality. The most basic object is the torch. You can also place tables, and chairs, and other objects as well.

## Gameplay

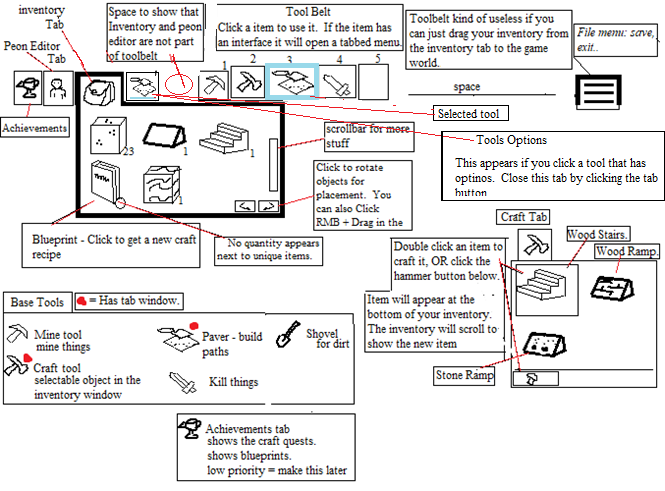
Your job is to modify the game world, dig, mine, build, and raise the peeps. The peeps serve to protect your base and keep the base flourishing. The base itself protects the treasure that you acquire. Your base is infinite. As your base grows, you’re able to travel to other parts of the world in search for different minerals. You’re able to adventure to different layers, and in these layers you get different materials that let you build your base stronger.

The layers of the world serve as gameplay gates. The overworld in the game does have some biomes, however, the overall game itself is built on different biome layers, each layer serves as a gate. In order to get through to the next layer you need the ability to destroy blocks in that layer. To destroy blocks in that layer you need to equip a pickaxe and such that are strong enough to mine through those blocks. At the start, we’ll use just basic granite, and keep going down, until you get to diamond, and death-stone, and other made-up rock types that are stronger than diamond.

The peeps of your world have different classes. Laborers provide the necessities to grow your world. They provide food and such. Military guys are equipped with bows, and swords, and they’re able to kill off intruders. Then, there are some people that don’t do anything. If they don’t do anything you’re able to turn them into a laborer, or a worker.

# Interface

The interface is like a combination of an RTS and Minecraft where you have your own toolbelt, but you’re also in god mode.



# Implementation Roadmap

1. **Phase 1 (World Edit) Due 2/19**
   1. Use Case - World Edit
      1. Implement **Inventory Tab button**
      2. **Implement 1 Inventory Slot** (for equipping pixaxe)
      3. Implement Inventory Tab
         1. Do not implement scrollbar. This won’t be needed until later. Simply place a single pickaxe inventory item in the inventory tab.
      4. Add a pickaxe object as a 2D image. And transform the mouse cursor to the pickaxe image.
      5. Pickaxe
         1. PDL
            1. Open Inventory
            2. Click Pickaxe and drag it to your inventory slot.
            3. With pickaxe selected in inventory slot, click a block in the game world.
            4. Holding down, the block gets destroyed.
            5. The block is then lay on the ground.
            6. Click the block to put it in your inventory.
            7. Block will appear next to the pick axe.
   2. Use Case - Scenery
      1. Update Renderer to handle detail (arbitrary) block geometry.
      2. Create Tree Objects in the game that may have arbitrary polygons (not blocks). This can be a simple **Block object that is skinner than the world blocks**.
      3. Generate these trees in the game world
2. **Phase 2 (Peep Refinement) Due 3/4**
   1. Peeps
      1. Create a sphere that wanders along the ground. Make it a “Peep” class.
      2. Implement “wander” mechanic (AI).
   2. Peep Selection – Clicking on a peep (sphere) highlights it.
   3. Get peeps to battle each other.
      1. Play some kind of “attack” animation for the sphere. For example, make the sphere rotate to attack.
      2. Add “battle” to the Peep’s AI stack.
      3. Increase Peep XP when they win the battle.
         1. XP = Peep Level \* Type Multiplier (type) \* Evolution Multiplier (per type).
      4. Evolve the peep if its XP goes over some amount
         1. Int evolve = 5000.
         2. Xp += xp; if xp > evolve then evolve() => { play an animation, set class level += 1}
   4. Peep Editor
      1. PDL
         1. Click a peep to select him in the game world. The peep will be highlighted.
         2. Show the peep editor button tab.
         3. IF the tab IS open THEN keep it open, otherwise, simply show the tab button.
         4. Draw basic stats in the tab.
   5. Generate enemy peeps randomly in the game world.
   6. **Content**
      1. Cannon – Shoots balls at enemies and destroys when hits ground. Must be operated by a peep.
3. **Phase 3 Paths (paver)**
   1. Create the path toolbar
      1. PDL
         1. Select the paver
         2. Clicking the paver opens up the “tools options” tab to the left of the screen. You can close the tools options tab by clicking the button.
   2. Build a path by placing a decal over the top of the block.
   3. Get peeps to follow the path if they are on it. Otherwise they wander around the ground.
4. **Phase 4 (World Refinement)**
   1. Add at least 5 details to the game world.
      1. Trees (should be done)
      2. Grass blocks.
      3. Dirt Blocks.
   2. First Game World
      1. Generate a second game layer below the first one.
5. **Phase 5 (Peep Refinement 2)**
   1. Render real peep as a Scrab (no more ball)
   2. Peep Editor - Peep Image – Draw the profile picture, or mugshot, of the peep in the peep editor.
   3. More peep design for the stats.

# Engine Features Wishlist

1. C# Scripting with Mono (probably unnc
2. Add a UI with multiple viewports to support the **shader graph.**
3. Project file format getting rid of */*data directory.
4. Vulkan Support for faster rendering.
5. Add a graphical PBR interface for modern shading.
6. **PBR**
7. **Cmake**
   1. Test CMAKE
   2. Change the engine to use Cmake as is used by most modern C++ projects.
8. Separate SDL window code so that we can have multiple windows.
9. **C# Script**
   1. Embed C# with Mono. Mono readily supports embedding.
      1. <https://www.mono-project.com/docs/advanced/embedding/scripting/>
      2. <https://www.mono-project.com/docs/advanced/embedding/>
   2. Remove “App” layer in place of C# callback function.
   3. Remove “Bottle” project in favor of C# language callback.
   4. C# callbacks to the C++ API
   5. C# Built-In classes
      1. Node
      2. MeshNode
      3. **vec3, ivec3 etc..**
   6. Compile by calling mono –
      1. Save the output CIL to /data/cache/scripts.dll
10. **User Interface**
    1. Description
       1. A UI is required to save projects and for editing functionality.
       2. The current UI isn’t well implemented and is abhorrently slow.
       3. The largest feature change in the UI will be having **multiple views**.
    2. UI Design
       1. The UI Design will be modeled directly off of Blender’s UI
          * Menu – File, Tools, Help
          * Toolbar - Left of screen.
          * Draggable View Splitter
          * Scene Outliner
          * Properties Window
    3. Views
       1. Because we are modelling this off of blender we will create views that show data aspects. We are only going to implement views that correspond to our PBR system.
          * Scene View
          * Shader Editor View
       2. Reorganize drawing code to work with multiple views.
11. **Vulkan Support**
    1. [https://vulkan-tutorial.com](https://vulkan-tutorial.com/)
    2. Reorganize GL code into generic units (especially “ShaderBase”)
    3. Research embedding SPIRV compiler (shaderc)
    4. Abstract the Vulkan API through SDL
12. **Component Infrastructure**
    1. We must change the game system to use components.
    2. Change Shake to use class aggregation (components) like Unity. Adding components when needed.
13. **Physically Based Rendering (PBR)**
    1. Albedo + Microsurface + Metal
14. **Shader Graph Editor**
    1. Shader Node
    2. Shader Node Link line.
15. **Physics Engine**
    1. Rip out the physics engine in favor of Bullet’s physics engine.
    2. Test BtVector3 vs Vec3x<> performance, and optionally, replace the math library.