

Original Game Analysis

The original game I selected to reimagine was Dig Dug, an arcade game released by Namco in 1982. The core mechanics of Dig Dug are digging through dirt in a sort of tile-based movement system, attacking enemies with Dug's pump, dropping rocks on enemies, and collecting pickups. The skill of Dig Dug comes from strategically using the rocks in the level to defeat multiple enemies at once, because removing a certain number of enemies this way causes a pickup to spawn that gives the player more points. The player also has to avoid being caught by the enemies themselves, which can maneuver through terrain faster than the player and sometimes spit fire at them. There is also an element of timing Dug's pump attack as well, as he cannot move while it is being fired and is left vulnerable. As the game progresses, the difficulty is amplified as the speed and number of enemies increases.

Dig Dug faced all the quirks of being a classic arcade machine game. For example, like many arcade games, Dig Dug has a “kill screen” glitch due to integer overflow. While the game should theoretically be infinite, at level 256 the error manifests and an enemy spawns on Dig Dug and drains all remaining lives, ending the game.

In analyzing Dig Dug, I was surprised to find that the game had more depth to it than I remembered when playing it as a child. I especially find it interesting how much was able to be accomplished with the limitations at the time.

Reimagined Version

A. New Theme

“Dig Dug reimagined as a game about mowing your lawn and removing lawn pests.”

B. Kept Mechanics

- Tile/Grid-based movement system

- I kept the idea of grid-based movement where the player creates a trail as they play. In both Dig Dug and my version, the player moves slower when creating the trail but faster on the trail itself.
- Enemy attack variants
 - I also kept the two main types of enemies in Dig Dug: one enemy which tries to follow the player and deals contact damage, and one that has a projectile-based attack.
- Level progression
 - In both versions, the level changes when all enemies on screen have been defeated.

C. Changed or Added Mechanics

- Placing traps instead of falling rocks
 - This change works well specifically with the version of enemy AI/movement patterns that I implemented, as it is not very complicated but makes planning around enemy movements more rewarding literally in the form of points.
- Spawning pickups
 - In my version, the pickup is spawned when enough grass has been mowed. This helps my game in two ways: 1) it emphasizes the “lawnmowing” aspect of the game more, giving the player an incentive to engage with that part of the game, and 2) as they are necessary for placing traps, it means devising more strategy as to how the player creates their trail to maximize tiles mowed as well as designing paths for enemies to follow.
- Enemies don’t move through walls
 - This change also emphasizes thinking out the path the player creates while moving, as the enemies will be making use of their trail as well.

D. Production Notes

The largest hurdle in creating this game was figuring out the tile-based movement. It was especially difficult creating a system that accounted for the

difference in speed for the player on empty or full tiles. I think figuring that out was the first big breakthrough in the project, as it was the most essential part of the game to get right to emulate Dig Dug properly. The code used for that process is also used repeatedly for many of the other systems of the game.

I would also consider the design of how each level is created a breakthrough. Effectively, my team created a very lightweight level editor for our template, so creating new levels was as easy as drawing them in the Pico-8 map editor. This allowed us to focus on implementing other, harder mechanics.

One of the major insights I remember from playtesting is that the original mechanic present in Dig Dug of enemies leaving the screen on their own did not especially make sense considering the reimagined game's new themes. With the current enemy AI, it would have also made the game kind of difficult to score highly on, as the enemies tend to find their way upwards quite a lot. That is probably the number one thing I would improve with more time, trying to develop a more robust enemy pathfinding system as the current version is very barebones and leads to the enemies not being especially difficult to predict or avoid (for example, they can only turn when they reach the end of their current path). I think I would also like to add more enemy variants just to spice things up for later levels.