

Procedural Generation in Super-W-Hack!

Ahmed Khalifa Gabriella A. B. Barros

Super-W-Hack! is a synchronous (<http://amidos-games.com/different-time-systems/>) roguelike game where everything is automatically generated: levels, weapons, bosses, and sounds. It is a tribute to the roguelike genre, and is inspired by various games, such as NetHack, Super Crate Box, The Binding of Isaac, Spelunky, Sproggiwood, and more.

In Super-W-Hack!, the player explores five levels designed into a 2D map similar to those in The Binding of Isaac. To proceed, one needs to clear each room of enemies. After killing all the enemies, the player must accept a crate that will replace his/her current weapon with a new one.

In this game, weapons are represented as patterns. One will not see the actual weapon causing damage, but will see its effect on the map before choosing to trigger it. Our intentions behind weapon generation was to (hopefully) increase diversity, since one player may never get the same weapon twice, and encourage strategic planning.

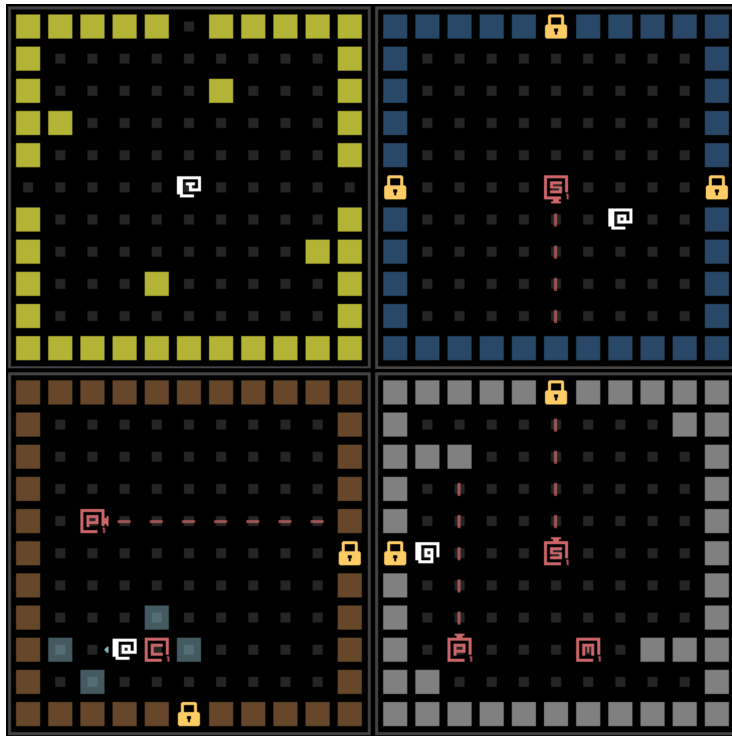
Weapon generation starts with a pattern that is filled arbitrarily in a random size grid. Patterns can be centered on the player or appear in front of him/her. If placed in front, they may also be infinite patterns, which repeat themselves until they reach a wall or enemy. Additionally, playtesting showed us that most players had a hard time predicting patterns behaviors when they were too “noisy”. Our solution was mirroring patterns in relation to the player.

Finally, the weapon’s name is generated using a combination of adjectives and nouns, and its sound effects are generated using sfxr. Both sounds and name are based on how good a given weapon is, which is calculated regarding how large is the area of attack and how protected is the player (can he attack from a distance?). A set of 100 weapons is generated at the beginning of the game, and sorted according to their power. Whenever the player gets a new weapon, one is selected and removed from the set, based on the level difficulty.

Levels in Super-W-Hack! are generated using multiple steps. First, the game chooses the dungeon name in the following format “*The #dungeonType of the #adjective #bossType*”. #dungeonType consists of 5 categories of dungeons which affects tile colors. #adjective is a list of funny adjectives added to the #bossType. #bossType is a list of different objects, animals and jobs.

After that, the game selects the map dimensions and generate a 2d maze of rooms using breadth first search, then assign a type for each room. Breadth first search is a search algorithm that, starting from a certain room, explores all non visited neighboring spaces and may transform it into a room, then repeats using the new rooms. Possible room types are the starting room, an enemy room, an empty room, or a boss room (only on the fifth level of the dungeon).

Finally, as soon as the player enters, the game starts generating the room. The first step is using cellular automata to generate the room structure. Cellular



automata is a technique inspired by Conway's Game of Life. Each map tile have a probability to be either solid or empty based on the surrounding neighbors. After that, if the room type is an empty room, then produce a crate ; if it is an enemy room, add enemies based on the level number and the generated gun power.

Super-W-Hack! has 4 enemy types: (C)haser chases the player, (P)atrol moves horizontally or vertically and shoot laser if the player in front of it, (S)pinner rotates in the middle of the room and shoot laser if the player in front of it, and (M)iner moves randomly leaving a mine trail behind it. Enemies can attack each other if an enemy is in between the player and another enemy.

Bosses contain two or three behavior strategies, which can be either movement strategies (moves randomly, teleports, chases the player), attack strategies (leaves a mine on the floor, charges towards the player, shoots a single spot or a laser in front of it) or a special strategy (spawns enemies, heals itself). The generator selects strategies and creates the boss based on the #bossType in the level name.

Super-W-Hack! was an ambitious project. We aimed at implementing many interesting features in a short time span. Although we didn't have enough time for enhancing the game to its full potential, it received positive feedback for its nostalgic art style, generated weapons, fast paced gameplay and short respawn time. On the other hand, some found the game confusing, as they expected the enemies to move after the player and not at the same time, and the current tutorial didn't clarify it enough. It was also a hard game, due to one-hit kills, enemies spawning too close to the player, among other reasons.