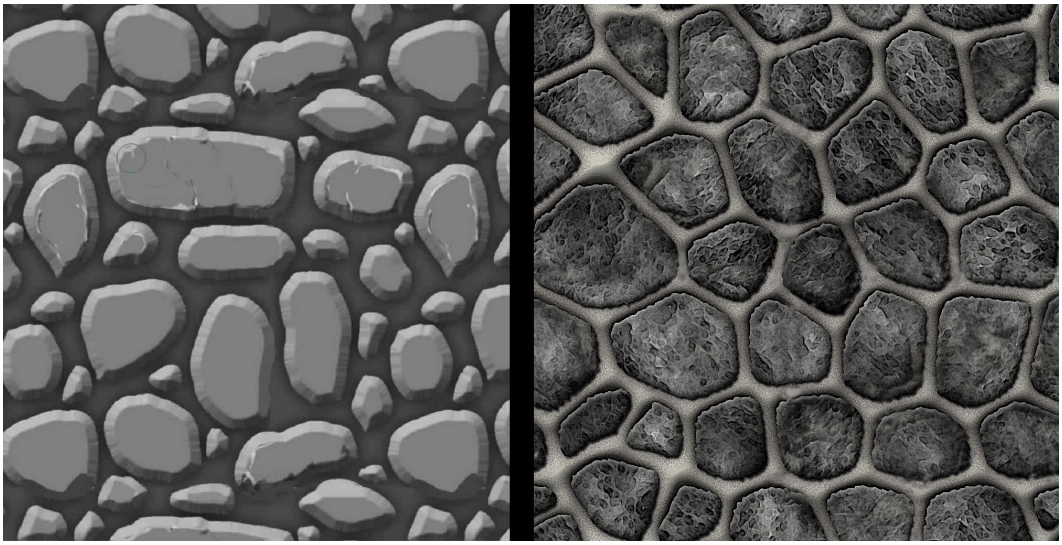
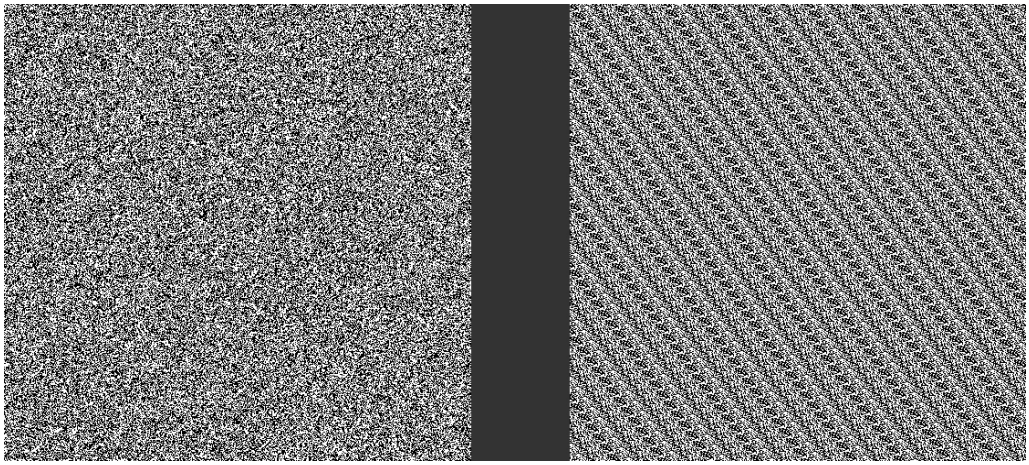
Chapter 1: Pseudo Random Numbers



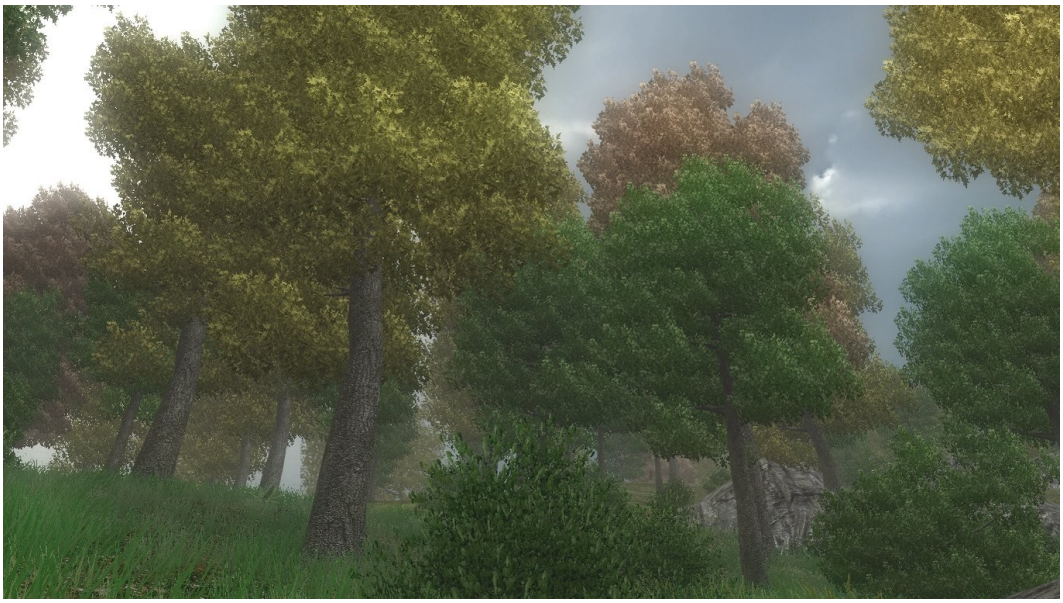
On the left is a hand-drawn texture, and on the right is a Procedurally Generated texture

A player-created building in the popular game *Minecraft*

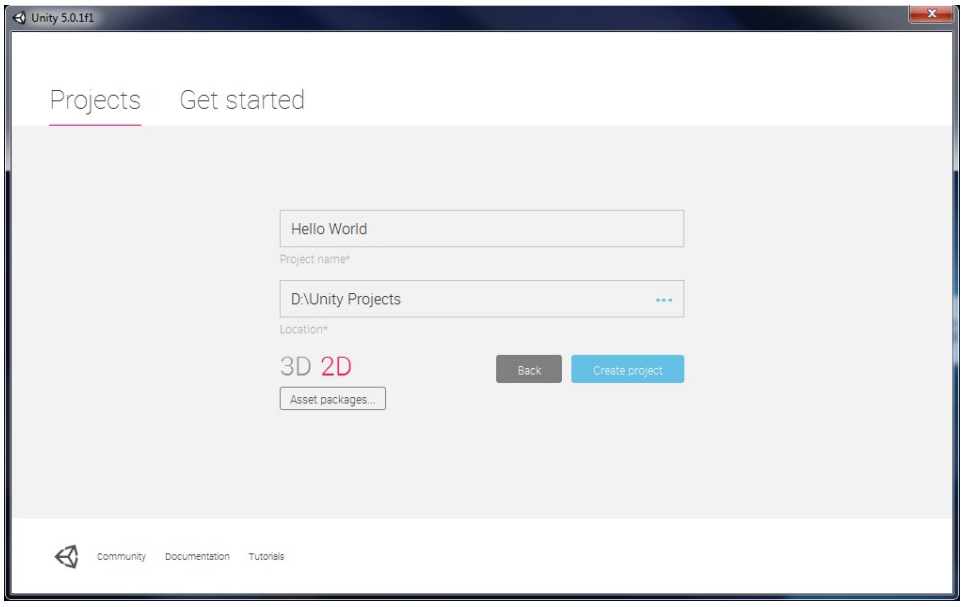
Some of Gearbox Software's *Borderlands* procedurally generated weapons each generated from asset modules



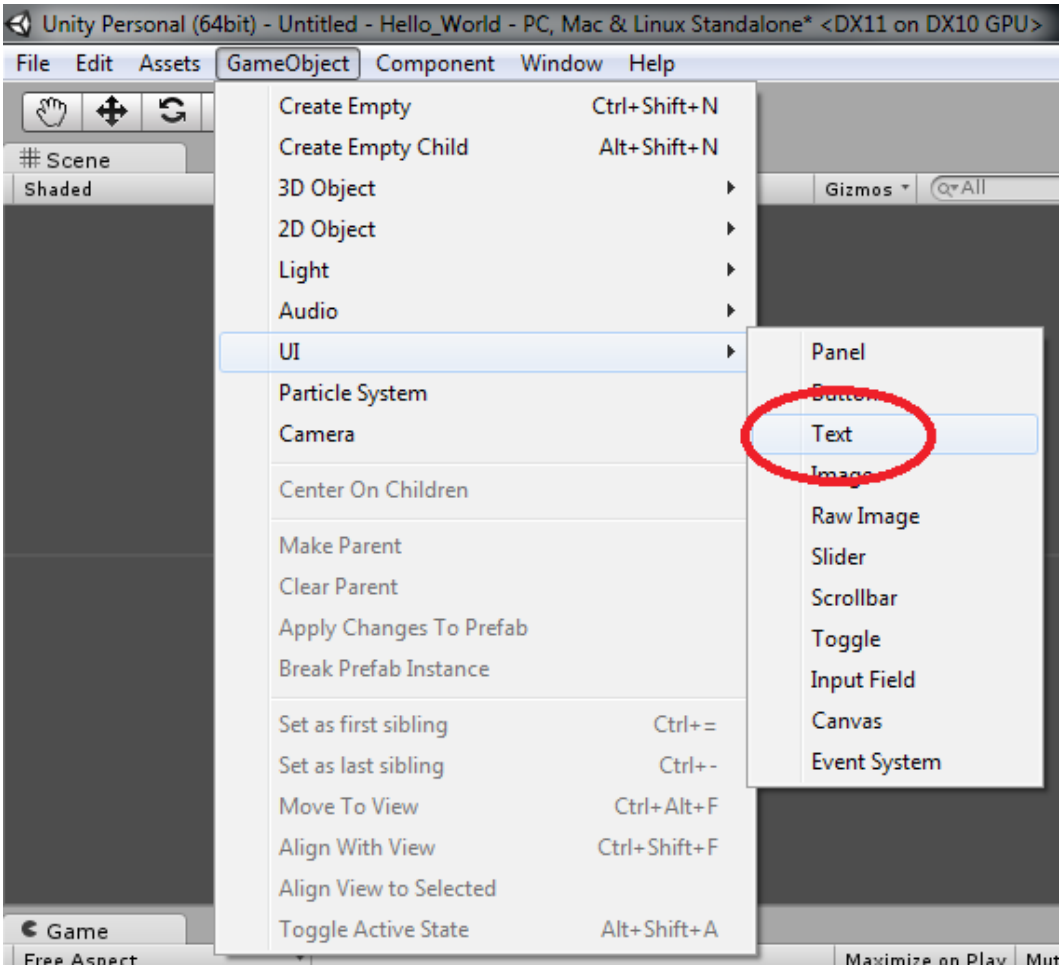
Random number noise signals – on the left is a random number pattern that does not repeat and on the right is a PRN pattern that repeats

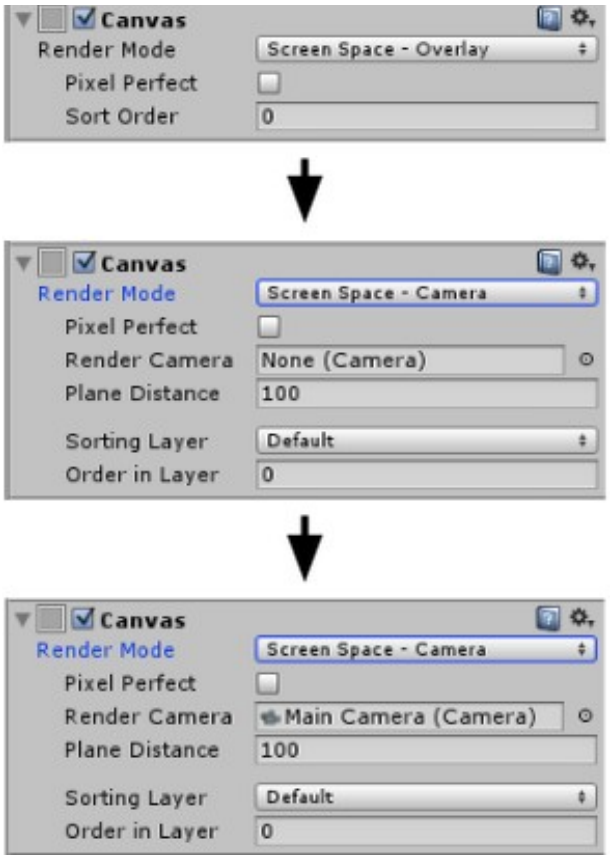


A forest scene created with Unity Terrain Engine which uses PRNs

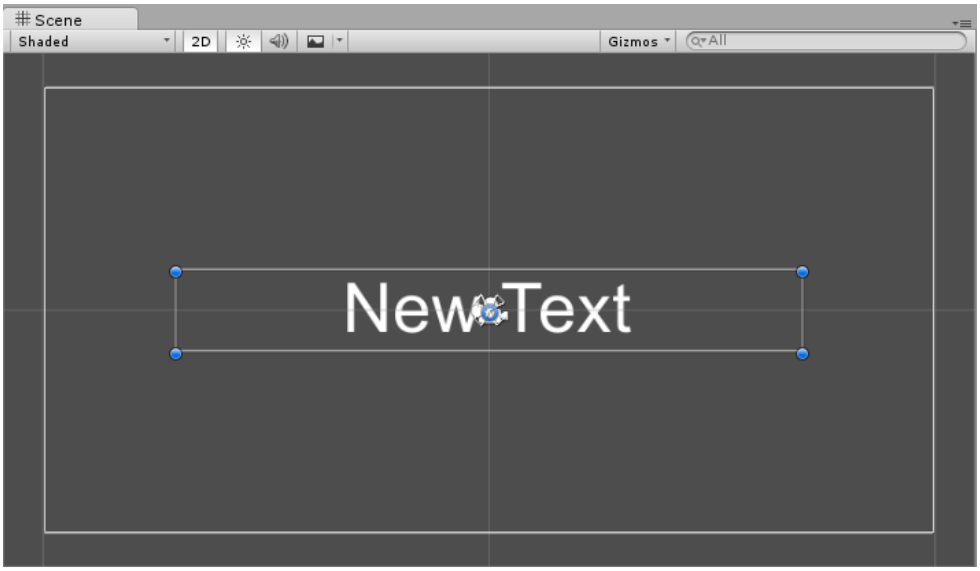


Unity launch screen

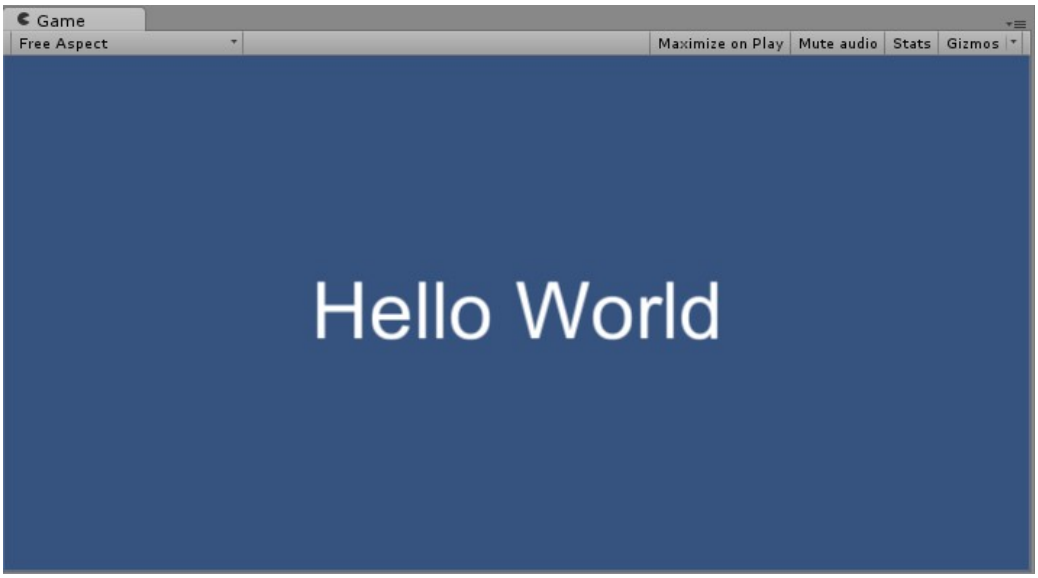

Unity Personal (64bit) - Untitled - Hello_World - PC, Mac & Linux Standalone* <DX11 on DX10 GPU>
File
Edit
Assets
GameObject
Component
Window
Help
Scene
Shaded
Gizmos
All
Create Empty Ctrl+Shift+N
Create Empty Child Alt+Shift+N
3D Object
2D Object
Light
Audio
UI
Particle System
Camera
Center On Children
Make Parent
Clear Parent
Apply Changes To Prefab
Break Prefab Instance
Set as first sibling Ctrl+=
Set as last sibling Ctrl+-
Move To View Ctrl+Alt+F
Align With View Ctrl+Shift+F
Align View to Selected
Toggle Active State Alt+Shift+A
Panel
Text
Raw Image
Slider
Scrollbar
Toggle
Input Field
Canvas
Event System
Game
Free Aspect
Maximize on Play



Workflow to get the Canvas in Main Camera view



Finished canvas and text formatting



Hello World program's result

Hello World with PRNs Program Result

Chapter 2: Roguelike Games

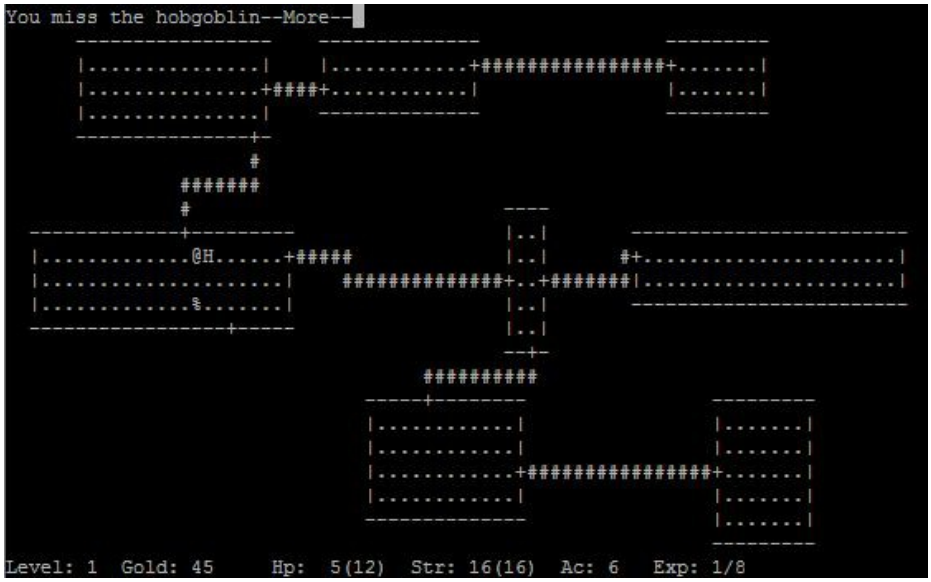
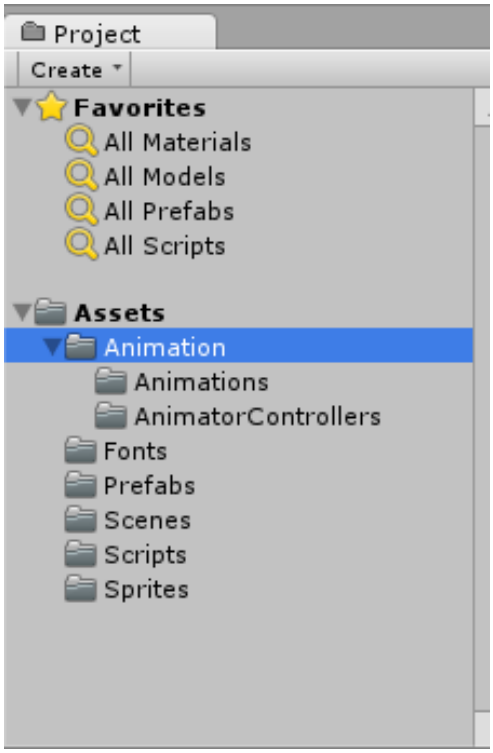


Image of Rogue (1980) by Michael Toy and Glenn Wichman

Tile sprite sheet from the popular game, *Pokemon*, developed by Game Freak

Unity Technologies' 2D Roguelike—courtesy of Unity Technologies

Results of importing the package and clicking on play

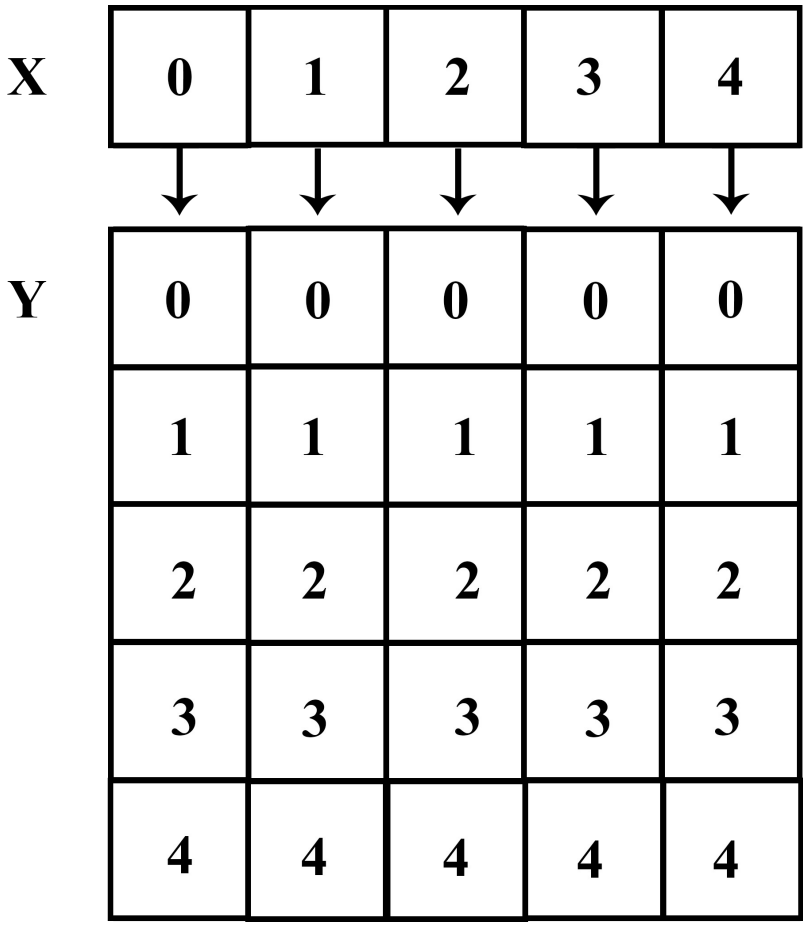


Folder structure for the Roguelike project

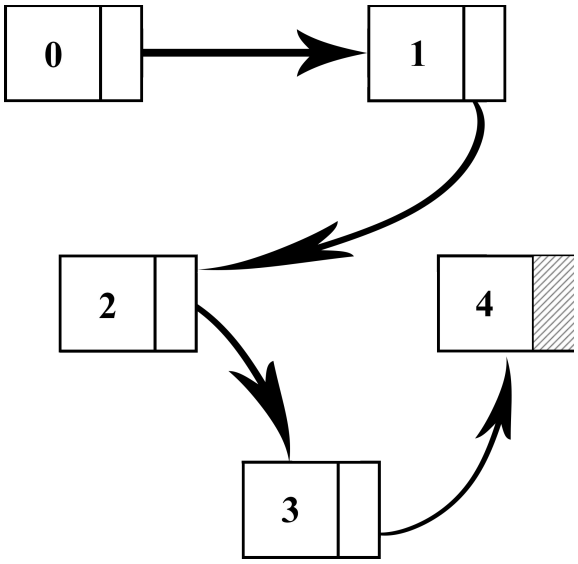
Chapter 3: Generating an Endless World

This is what our Roguelike endless PCG Game Board will look like

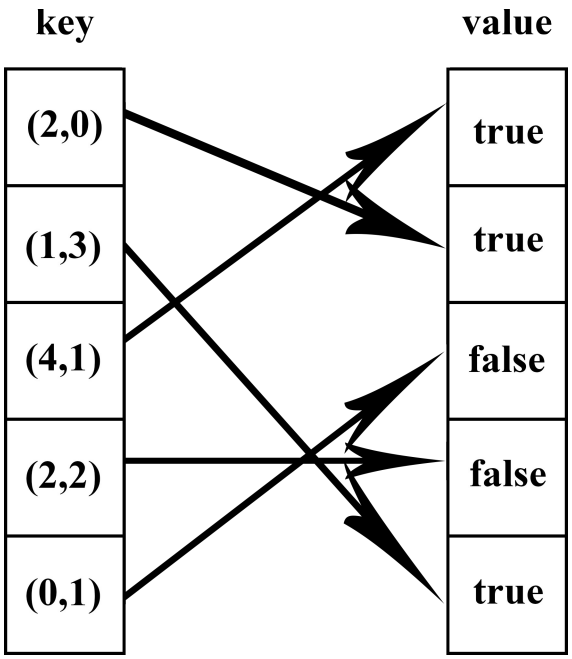
Imagine the Game Board within a grid



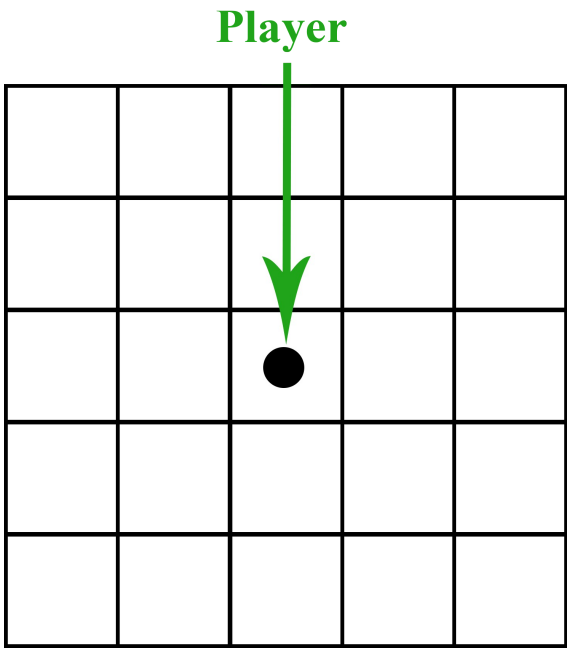
Visualization of a 2D array



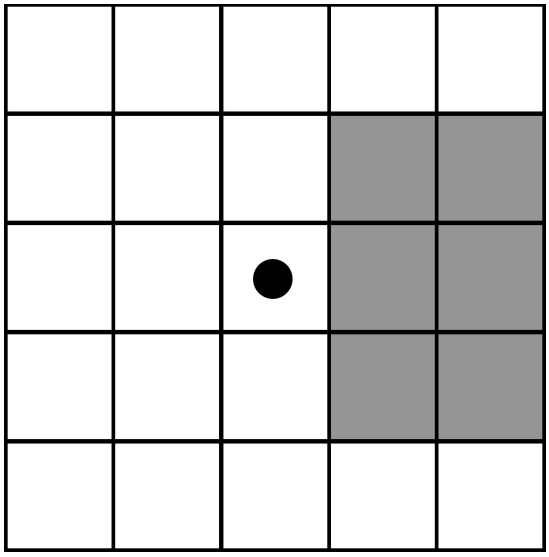
Visualization of a linked list



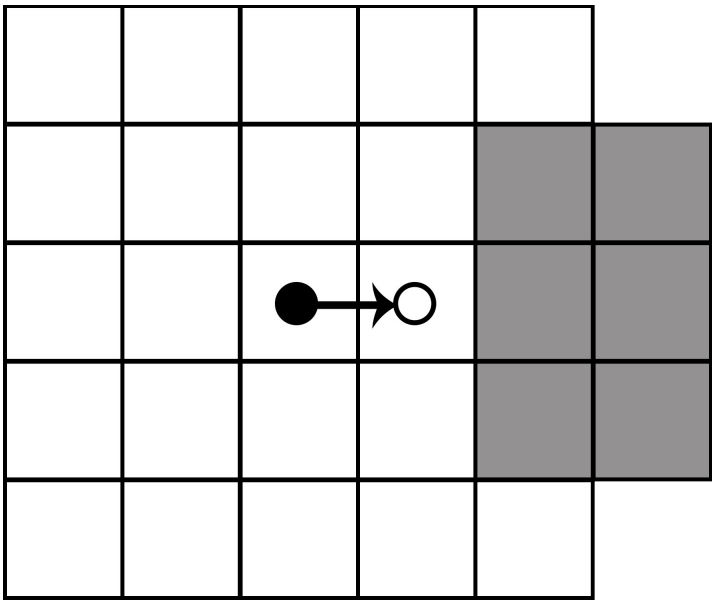
Visualization of a dictionary/map



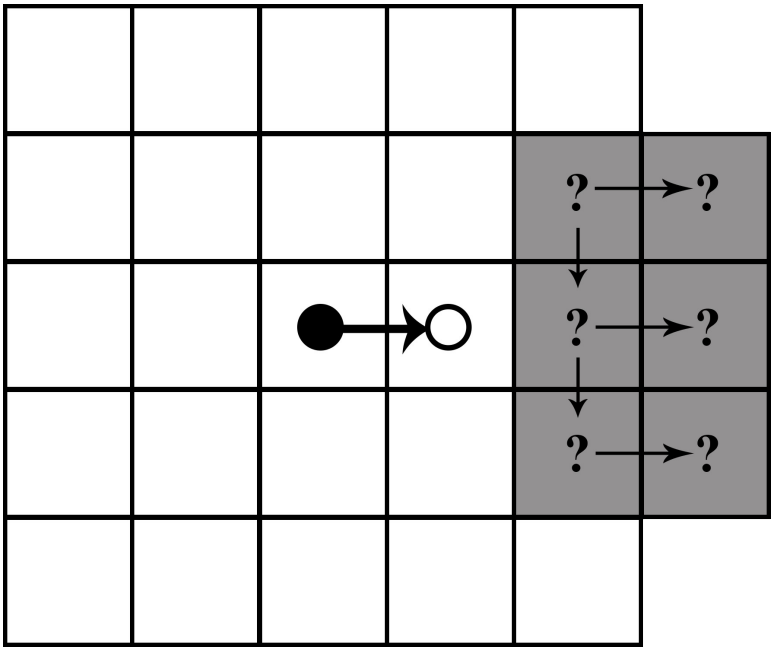
The initial Game Board grid with the player



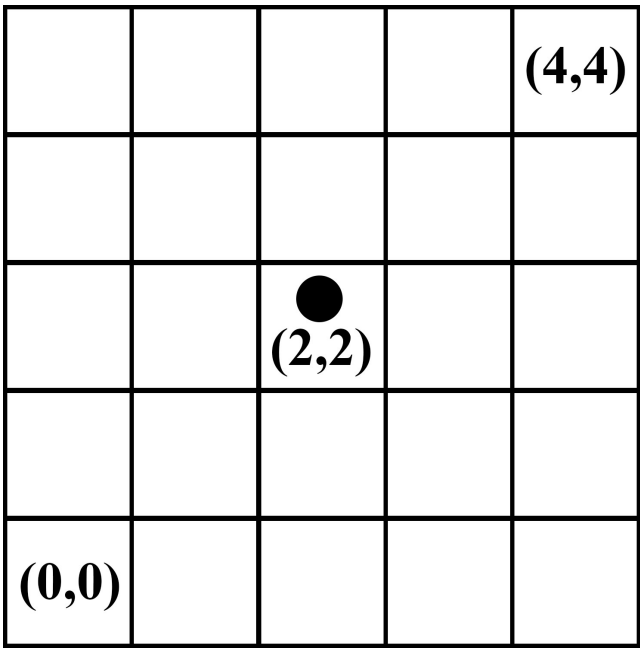
The line of sight grid squares are shaded



Revealing more tiles as the player moves right

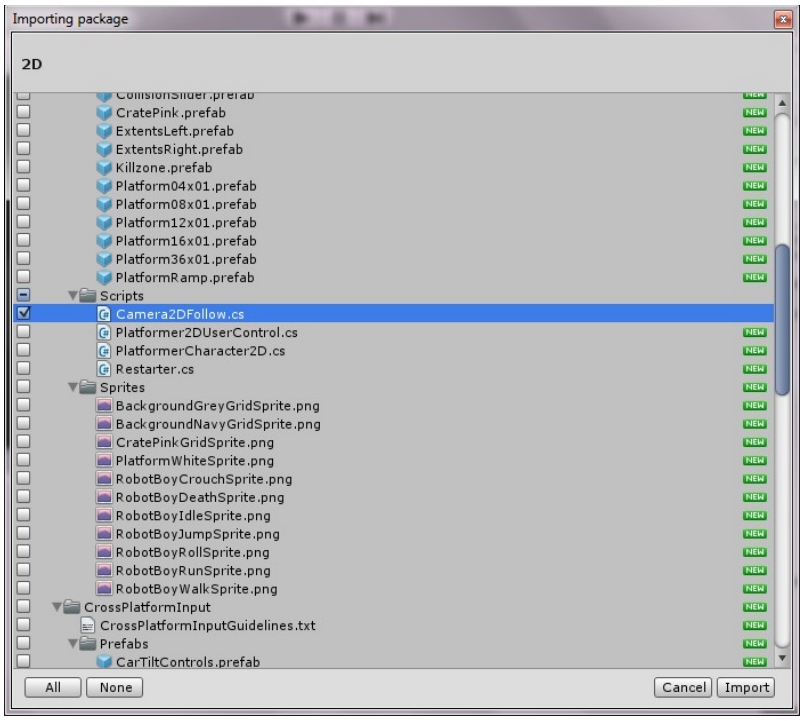


The tiles are iterated over to check whether they have already been discovered

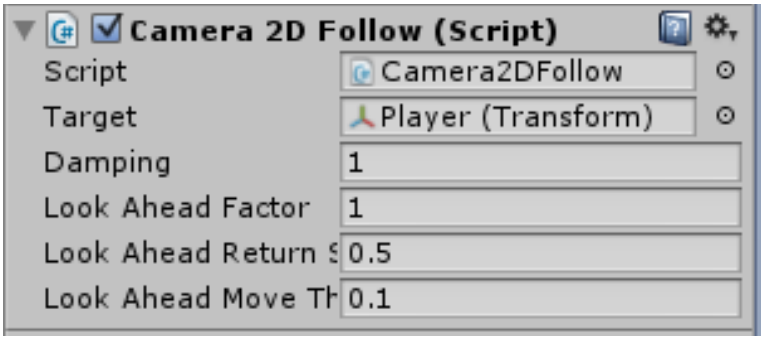


The grid will correspond with the x-y plane

All our sprites are 32 x 32, which is 1 unit of measure

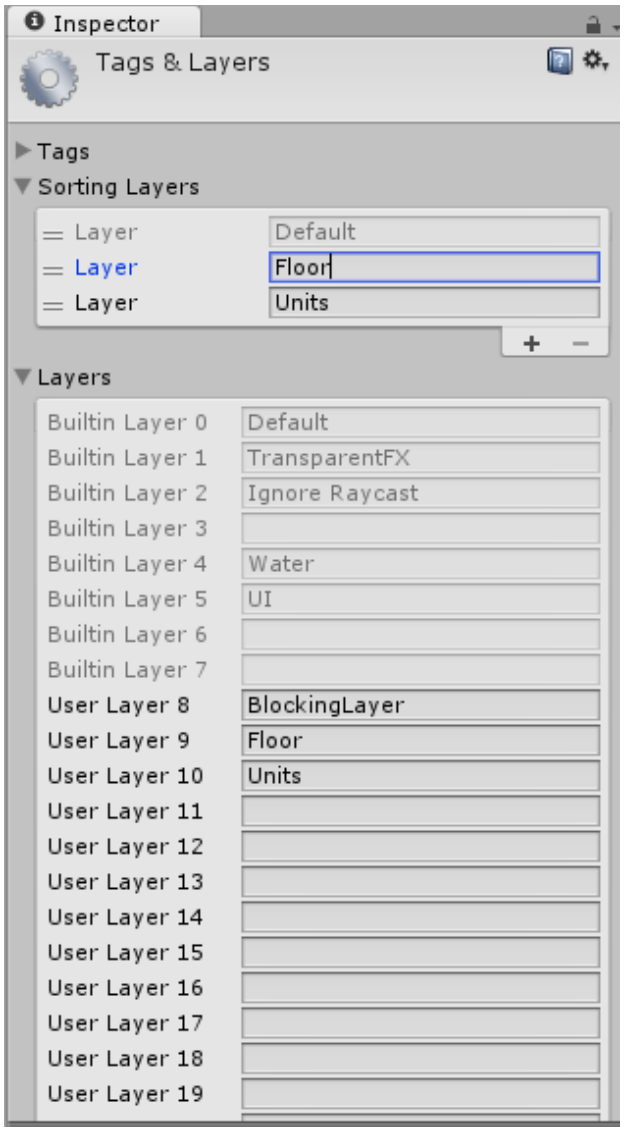


Import settings



The Camera 2D Follow settings screen

No Game Board



The Layer settings

Initial Game Board

PCG Game Board

PCG Game Board plus wall tiles

Chapter 4: Generating Random Dungeons

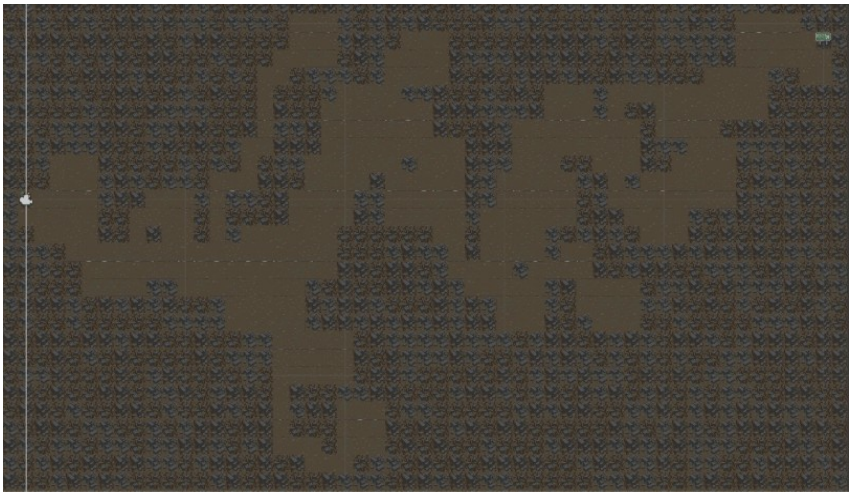
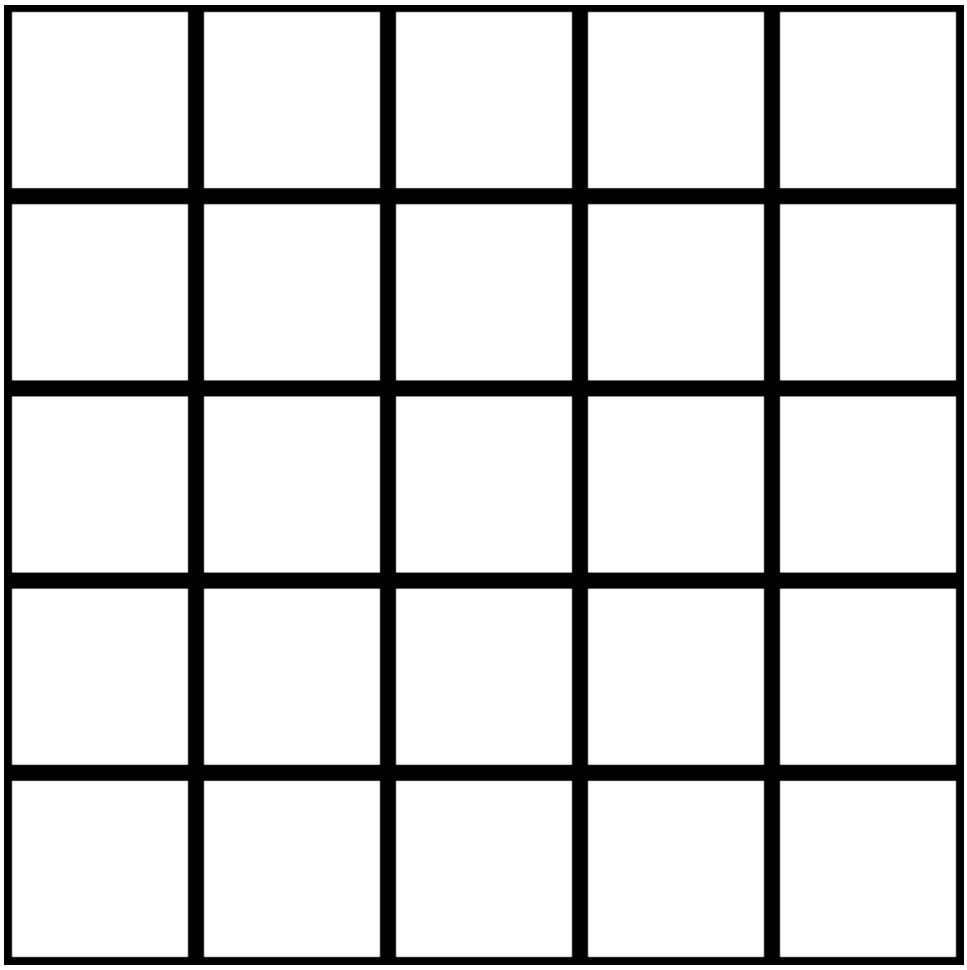
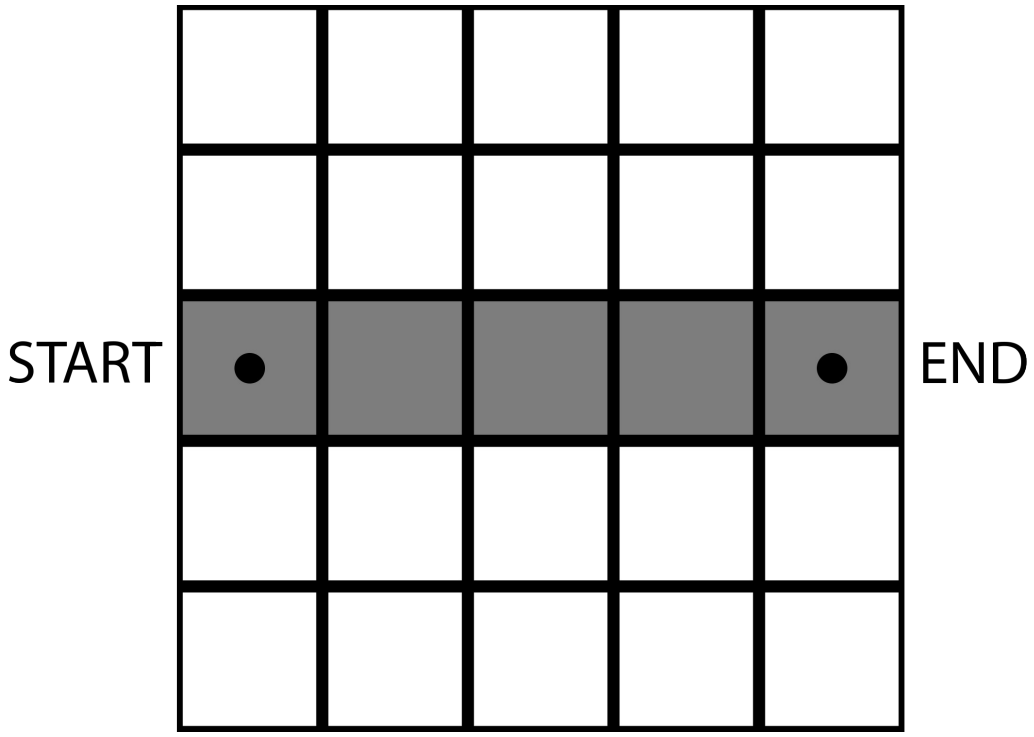


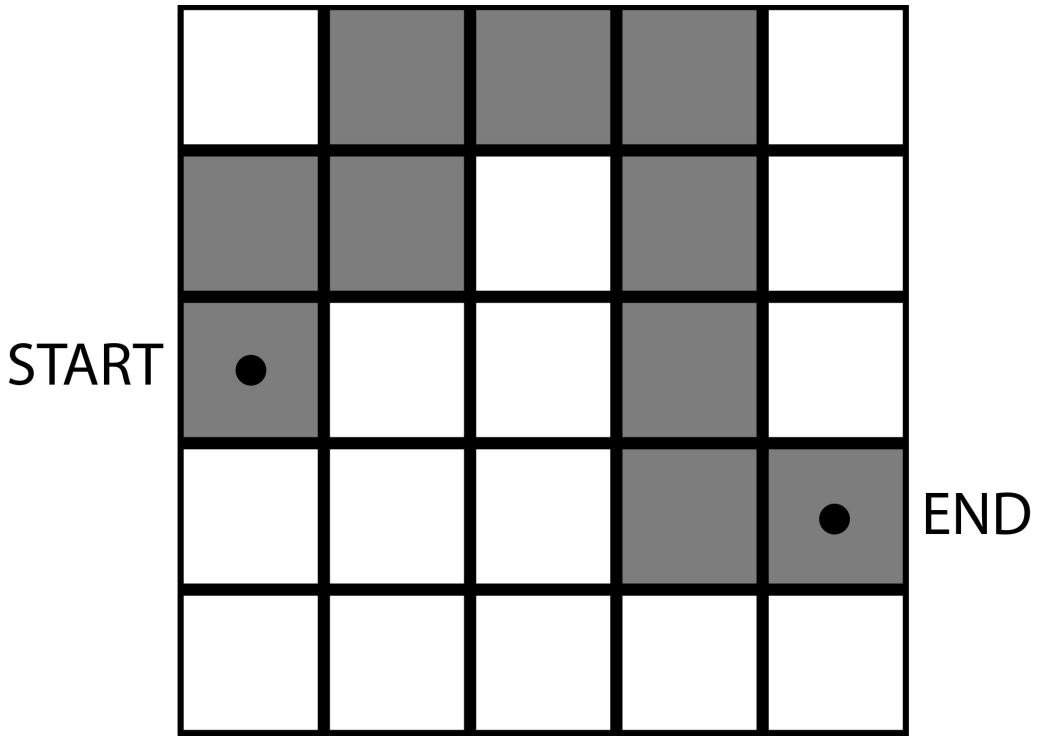
Image of final result of dungeon generator



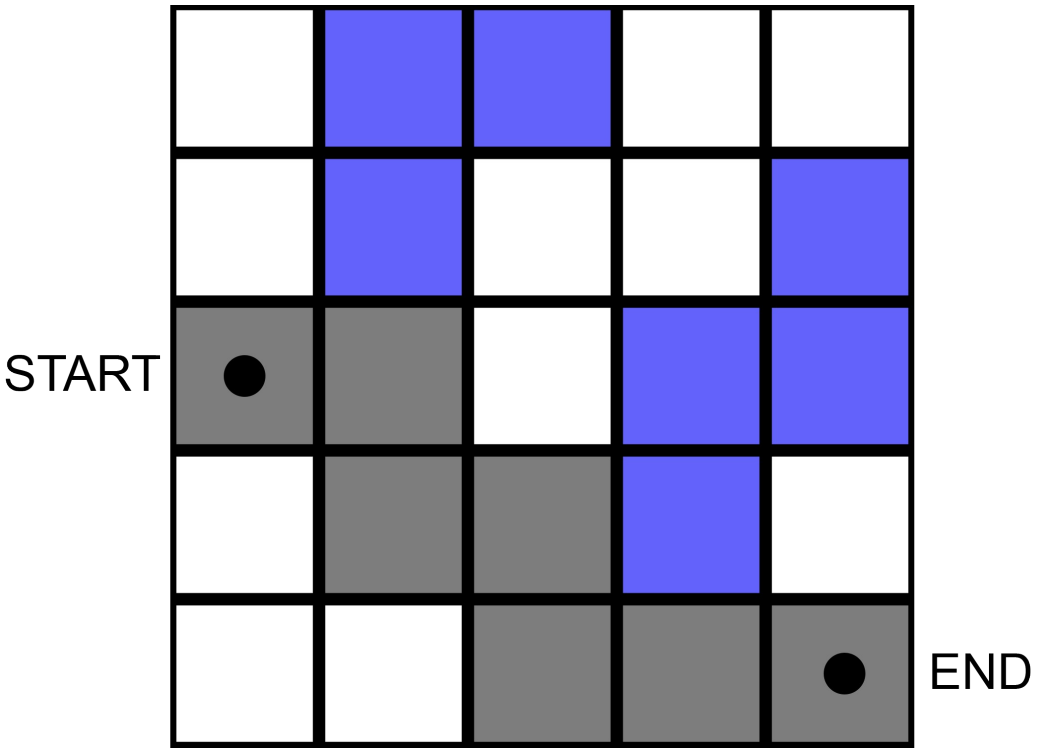
A diagram of the simple grid



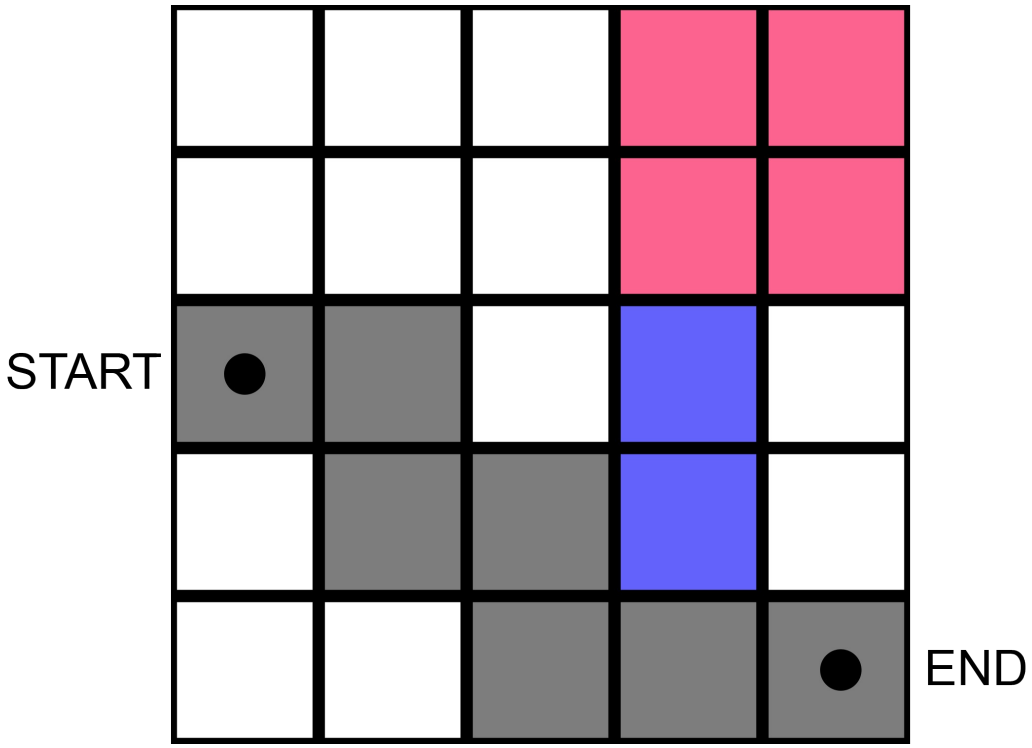
A diagram of the simplest essential path



A diagram of the winding essential path



A diagram of random paths in blue



A diagram of the chamber in red

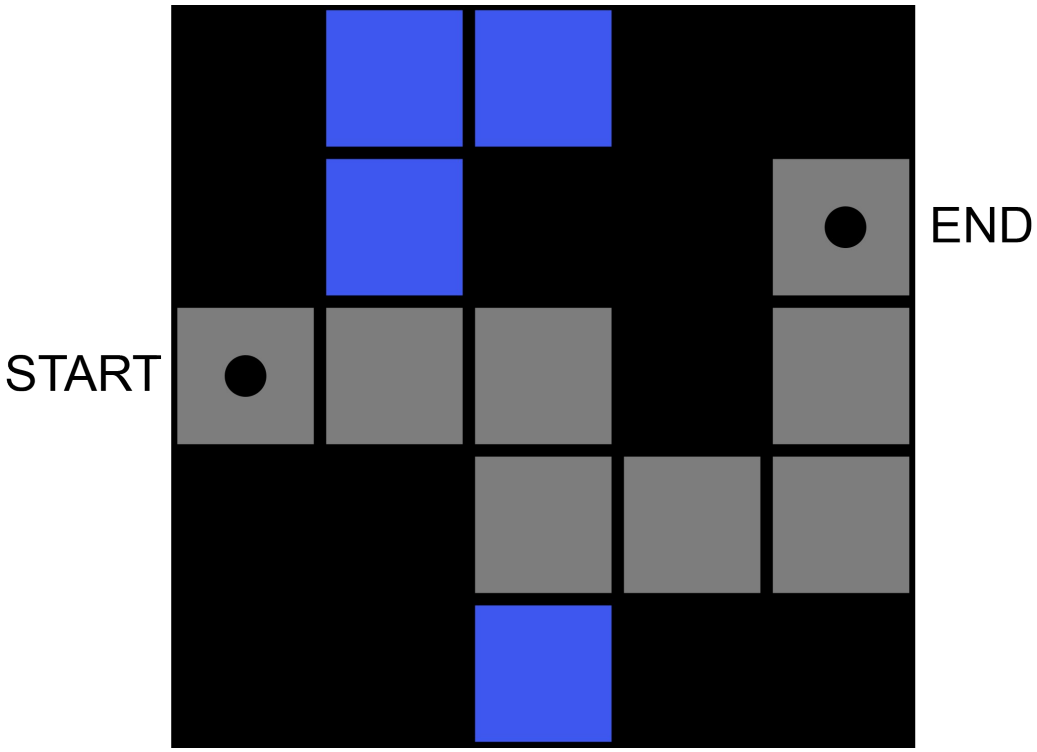
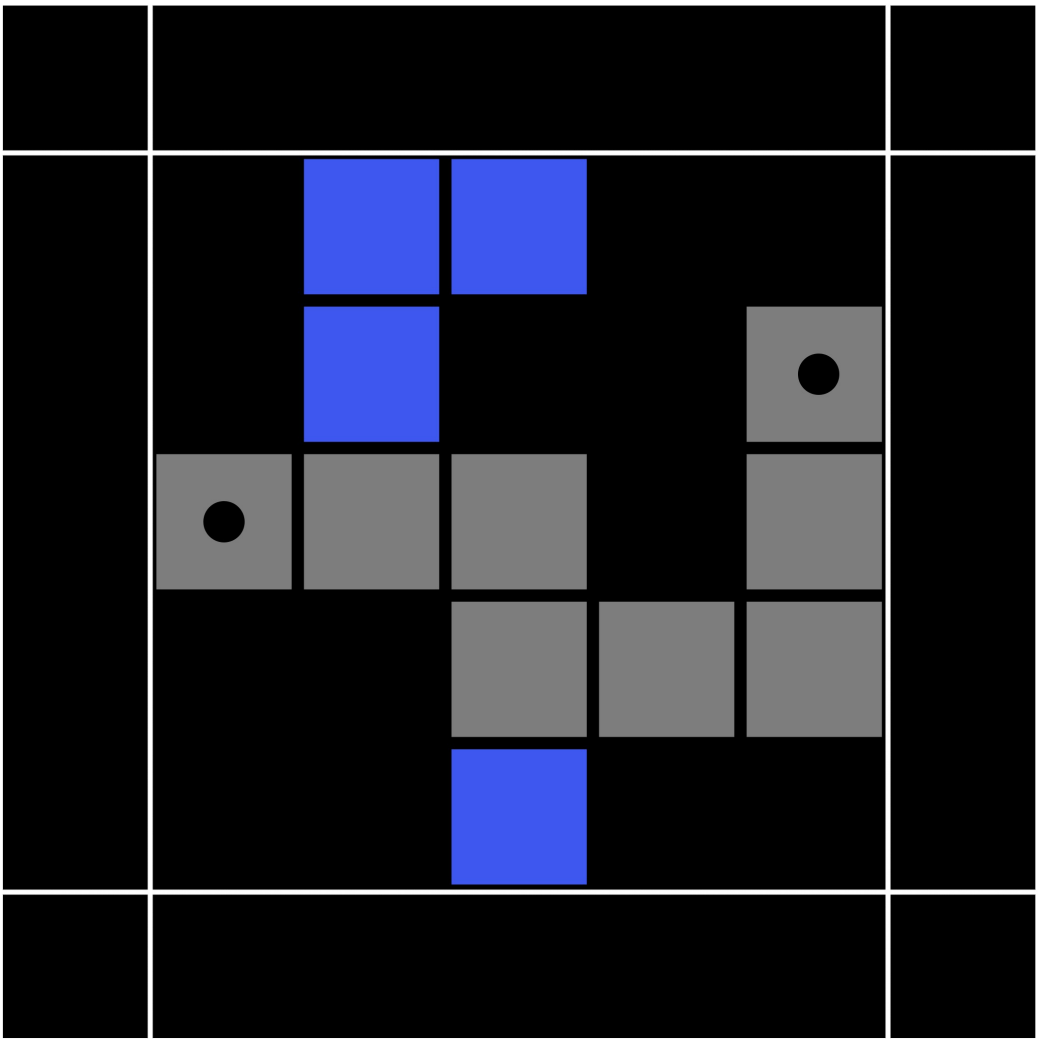
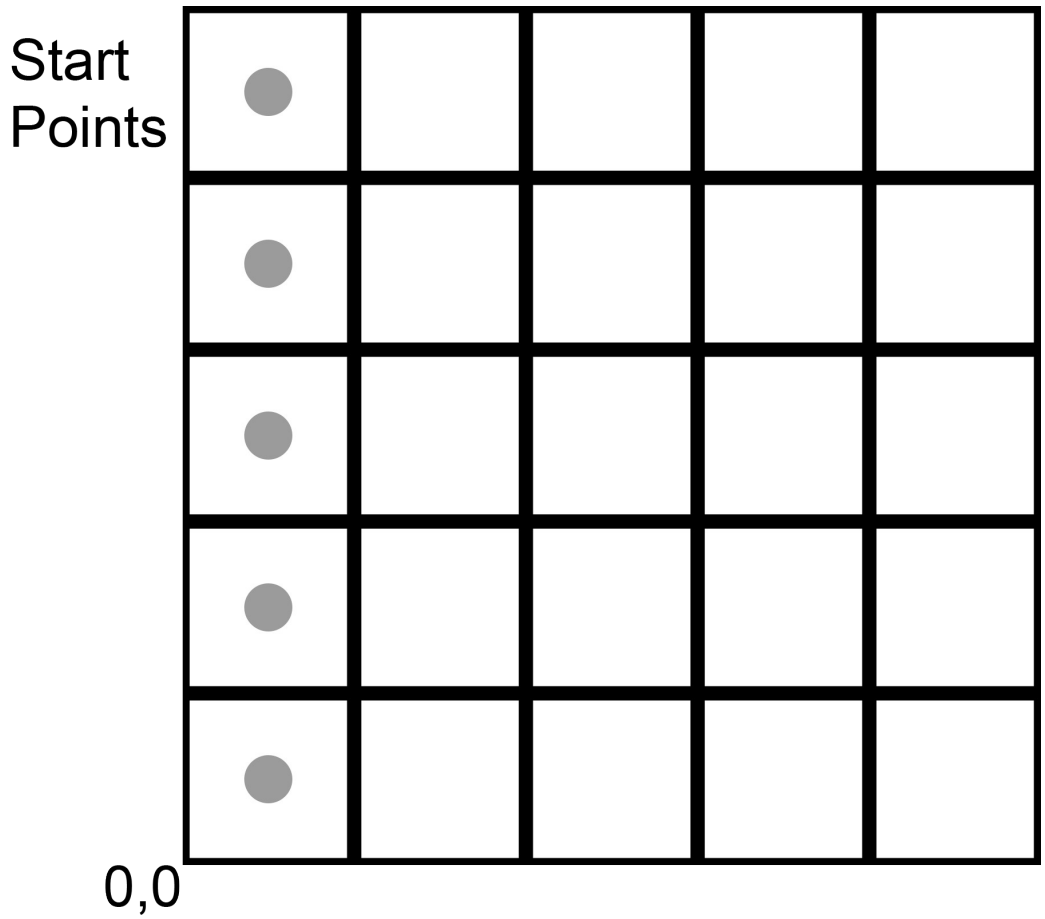


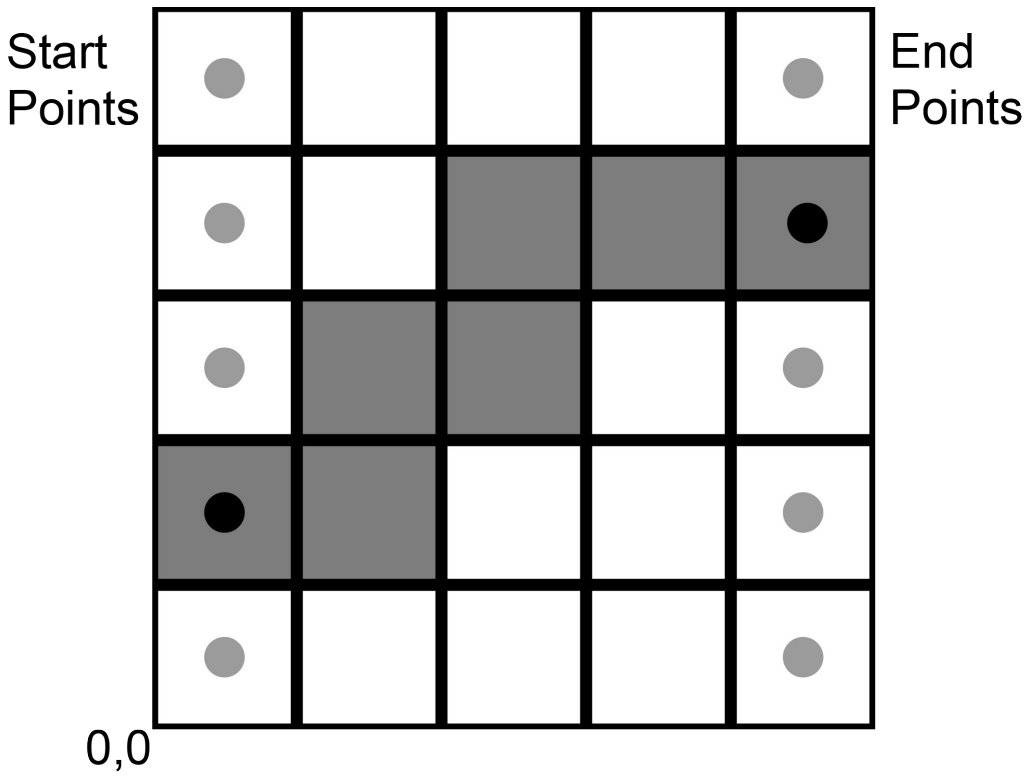
Diagram of outer wall tiles in black



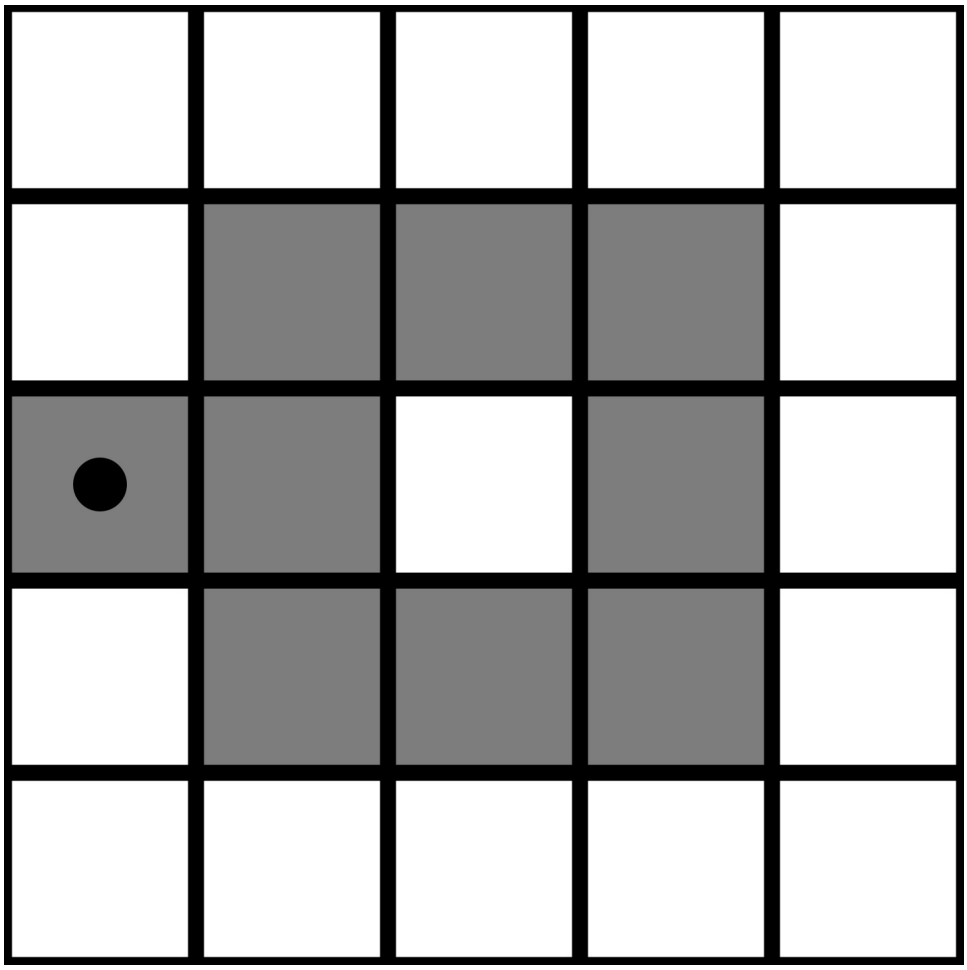
A diagram of the outer wall tiles enclosing the dungeon in black



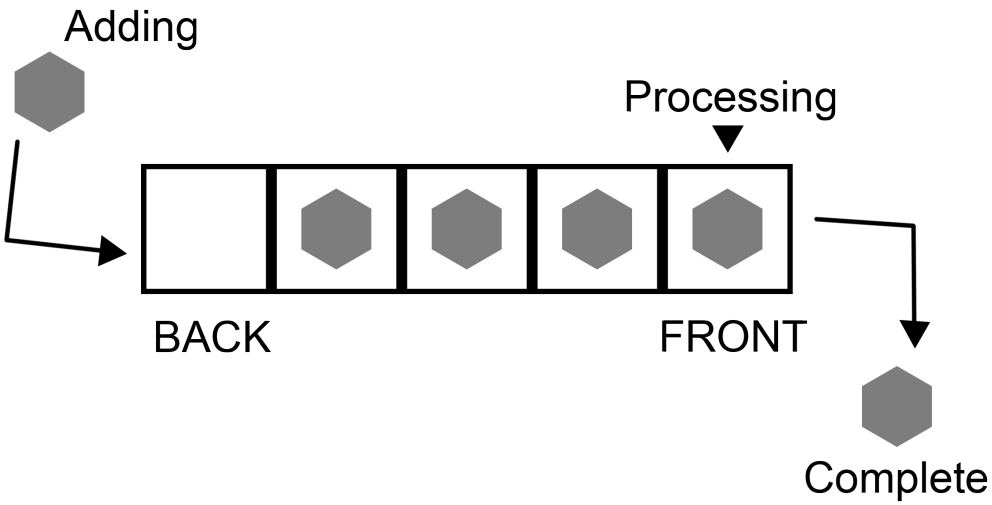
A diagram of possible starting points



A diagram of the placement of exit based on the path



A diagram of a path cycling back



A diagram of a queue

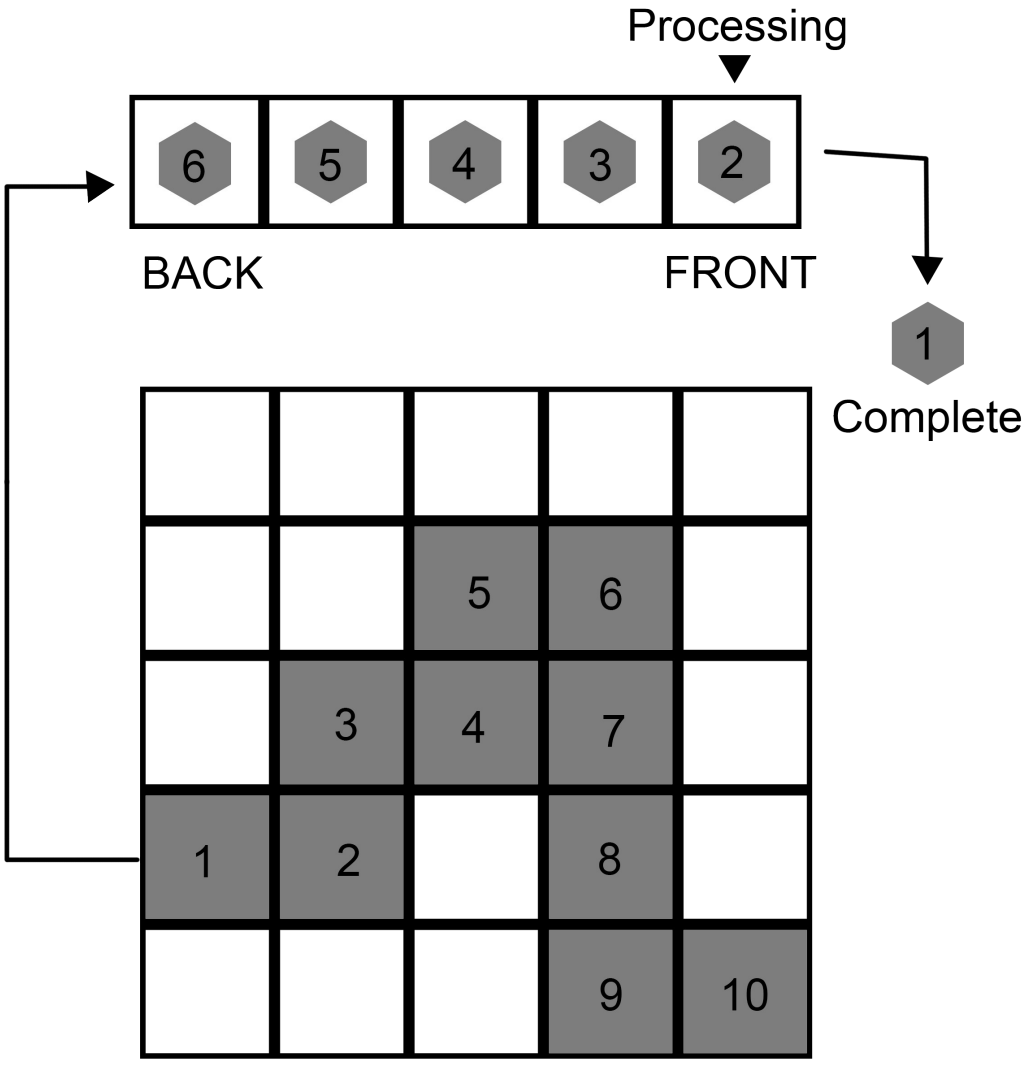
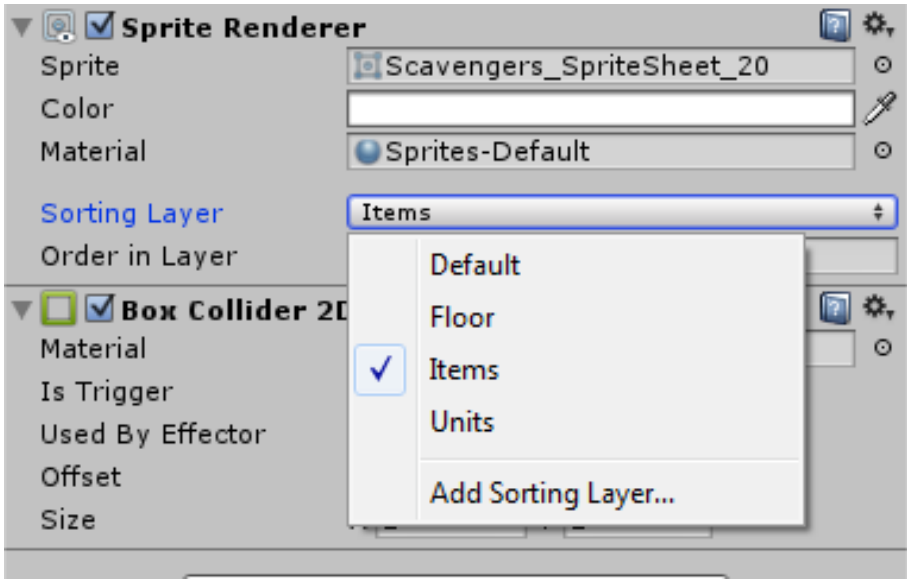
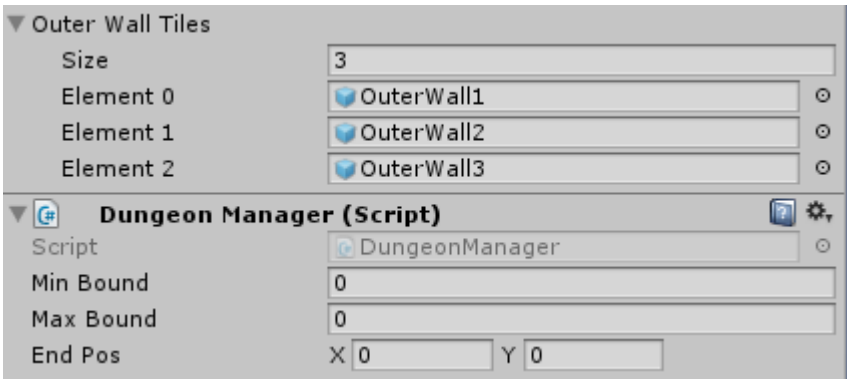


Diagram of essential path in queue

Image of exit tile sprite



An image of sorting layer order



New options in the GameManager prefab

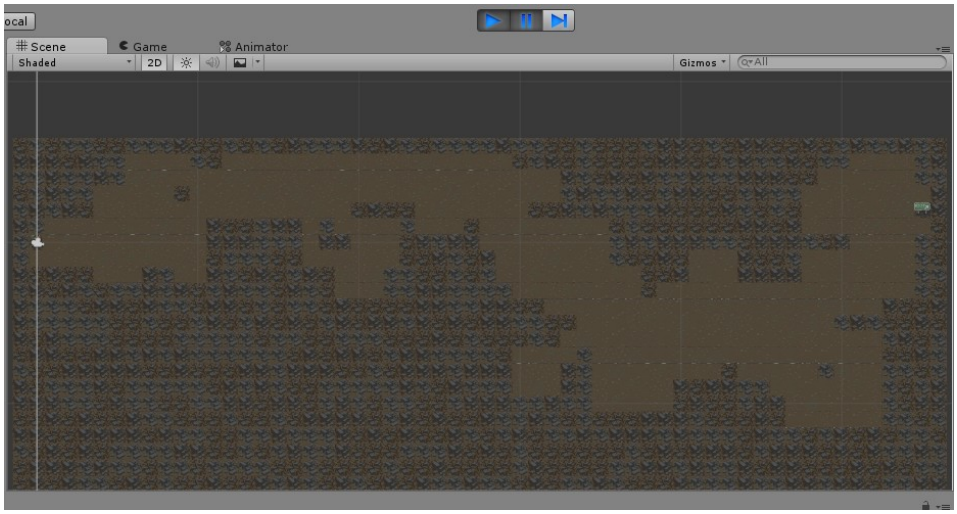
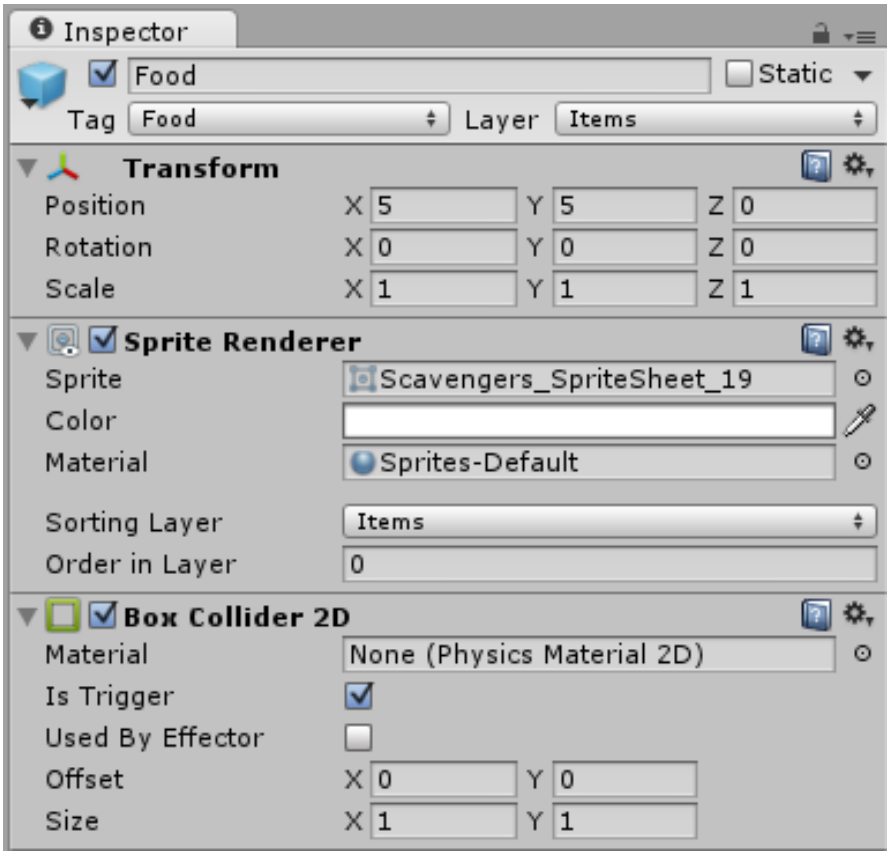


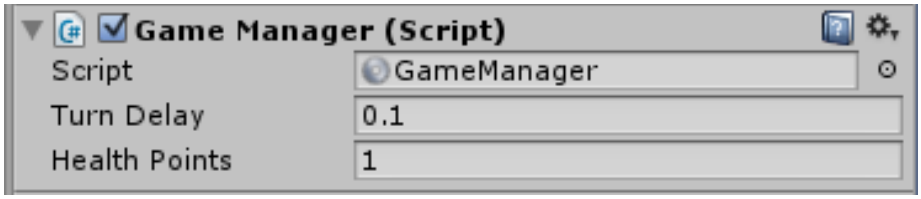
Image of dungeon overview

Chapter 5: Randomized Items

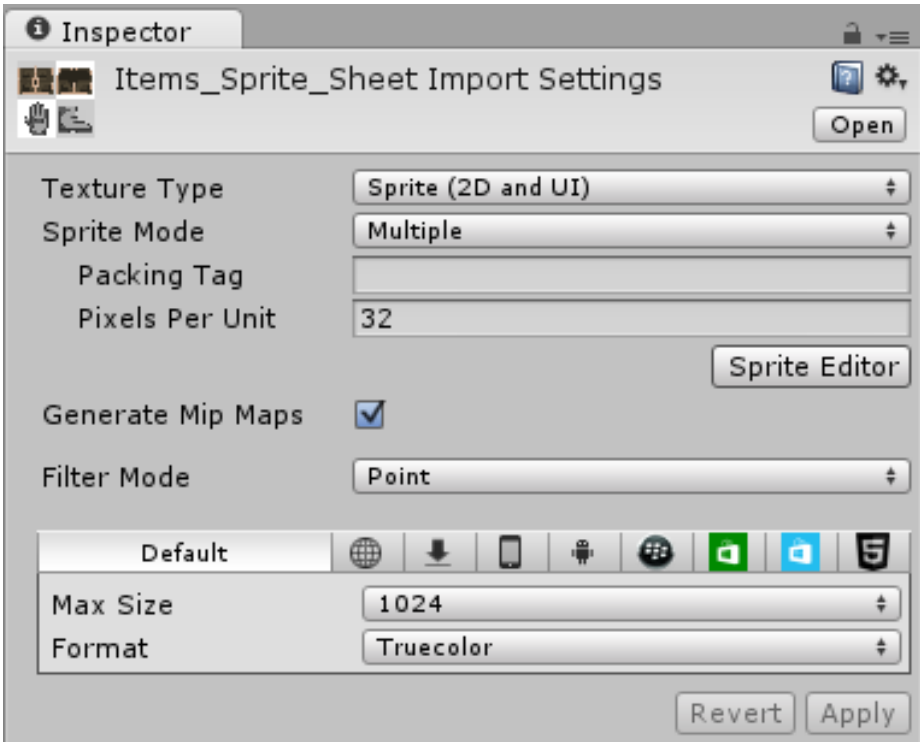


Food item setting

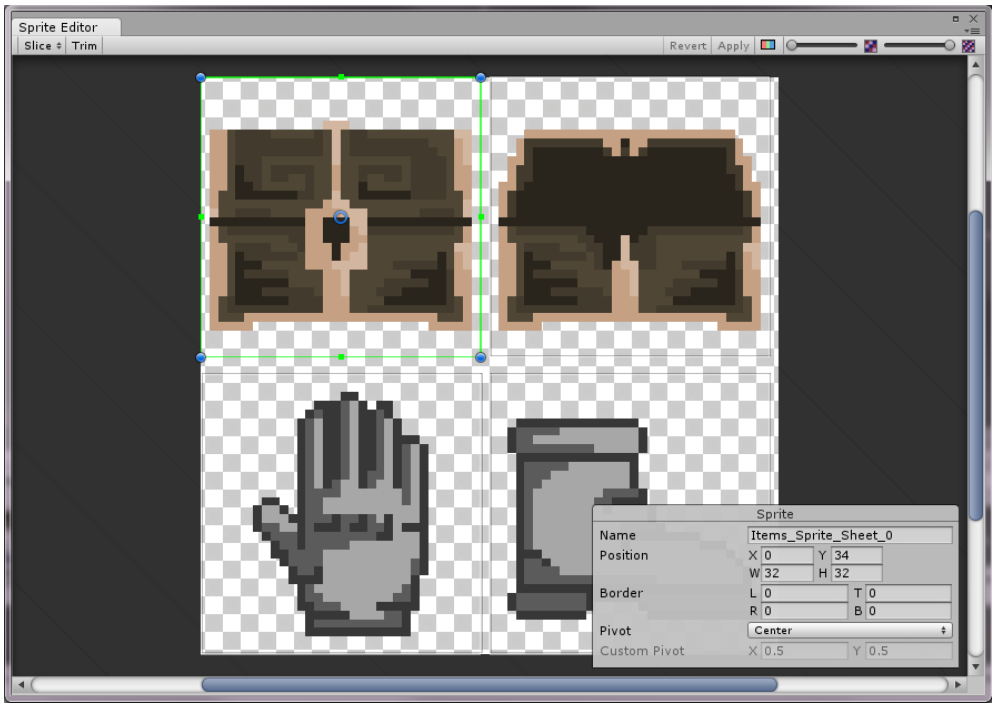
Health item appears when a wall is destroyed



Where to find and set health point



Imported sprite sheet settings



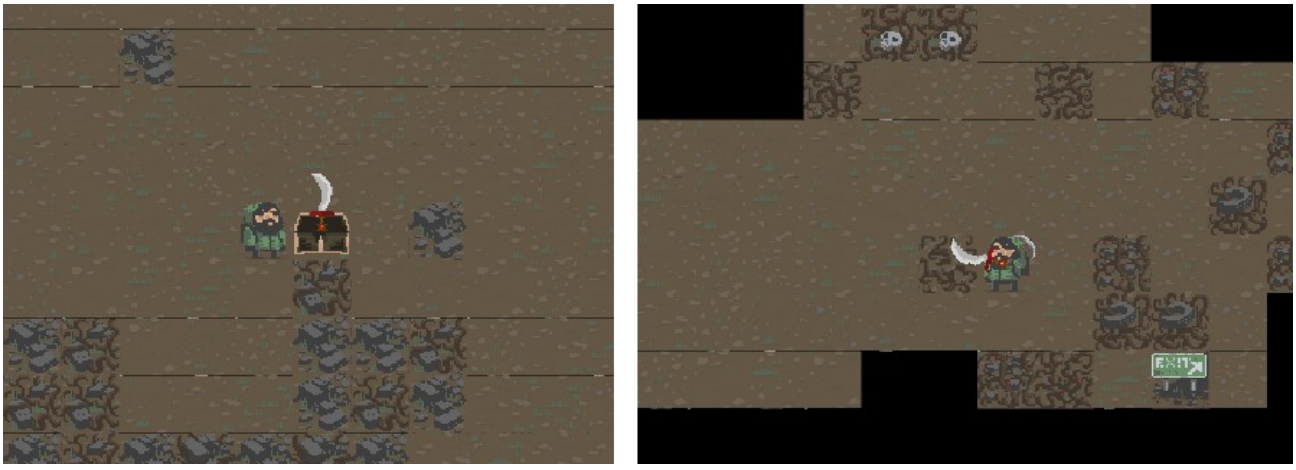
Sprite Editor

A randomly spawned chest

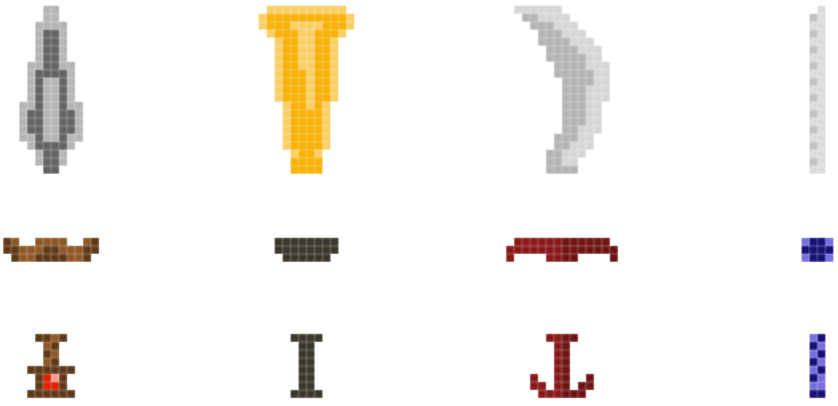
chest and item but no visible inventory

Picked up item shows in inventory

Chapter 6: Generating Modular Weapons



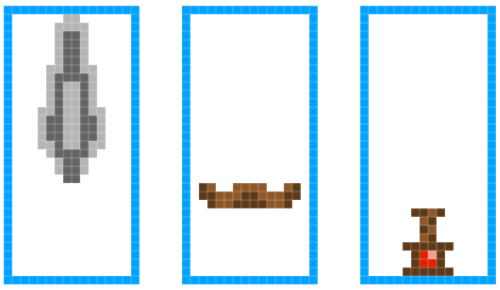
The final result of the modular weapon implementation



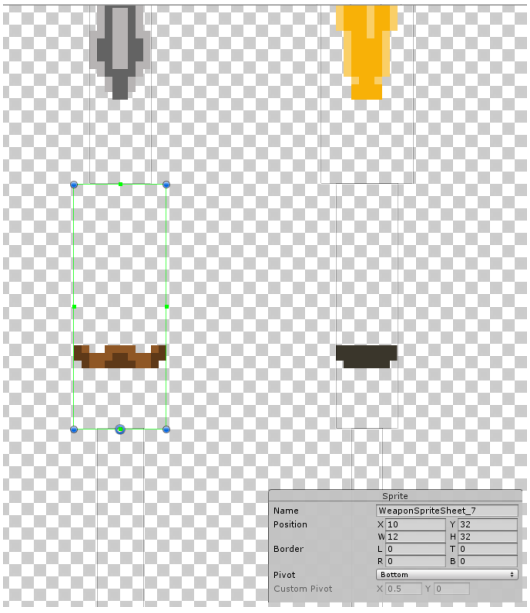
Weapon module sprites

$$4 * 4 * 4 = 4^3 = 64$$

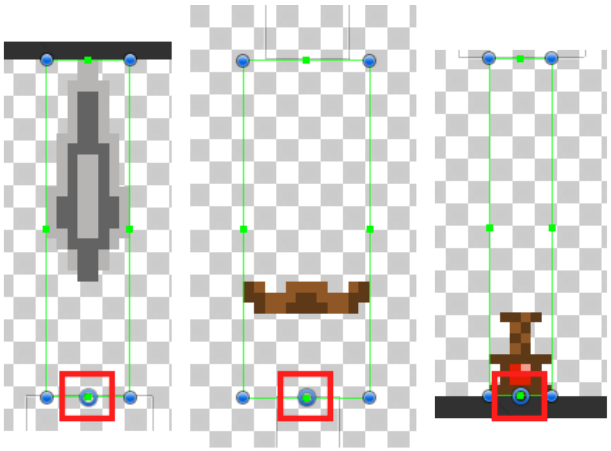
$$5 * 5 * 5 = 5^3 = 125$$



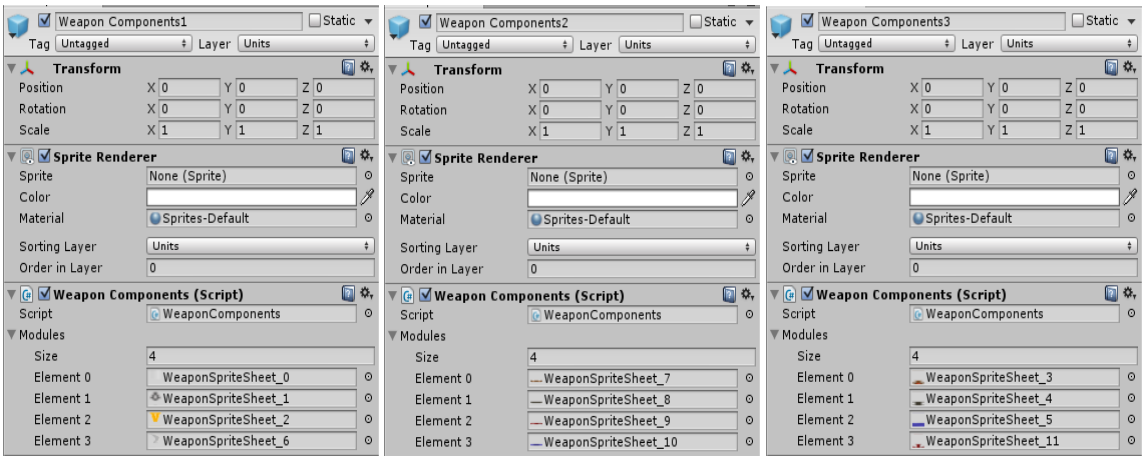
Weapon modules in their respective bounding boxes



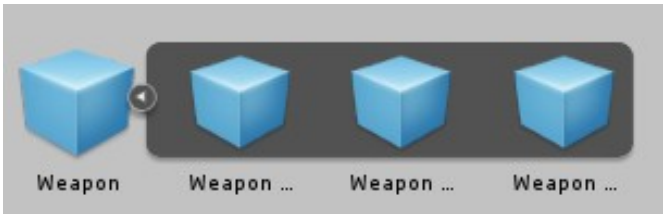
Measurements of a module



Pivot points for each type of module



Weapon Component's settings



The expanded Weapon prefab

Chest spawning a weapon

The weapon is unhidden and follows the player, while there is a blank image to the right

The sword is hidden and the icon to the right appears

Weapon animation

The player faces to the right and the sword swings to the right as well, while the wall on the left is damaged

The player can now face the left

The Player can swing the sword toward the left

Chapter 7: Adaptive Difficulty

Player among enemies!

Enemy Sprite

Enemy walking on black space

Enemies on world board

Enemies on Dungeon Board

Attacking an enemy

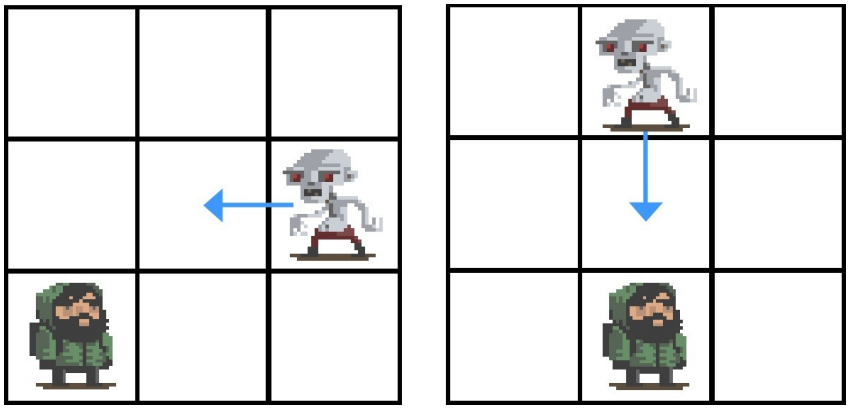
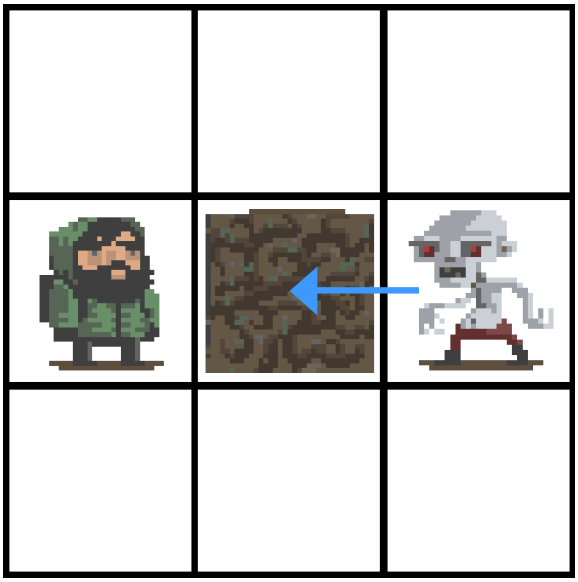
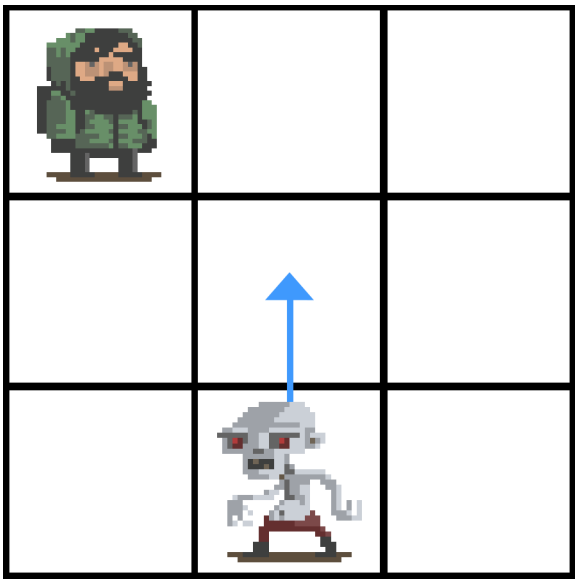


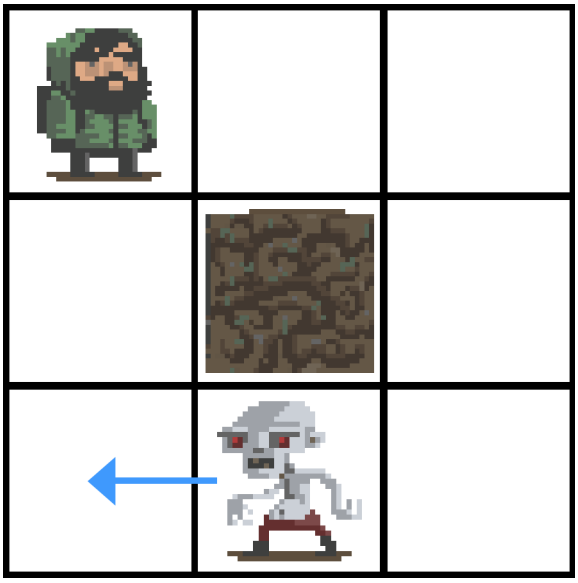
Diagram of simple enemy movement



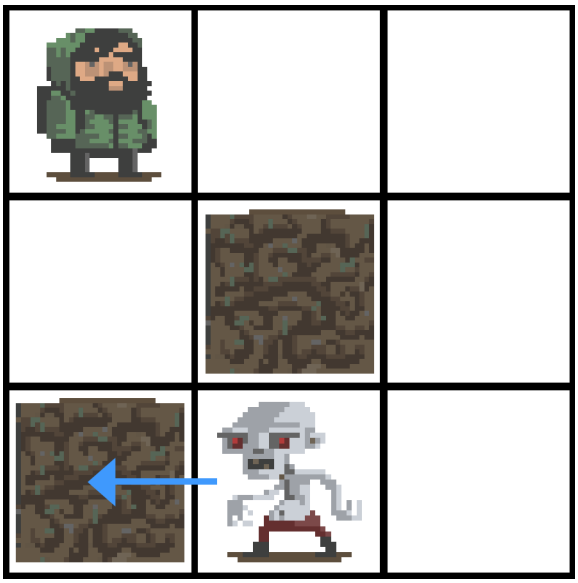
Enemy stuck on wall



Player is 1 space away horizontally and 2 away vertically so enemy moves vertically

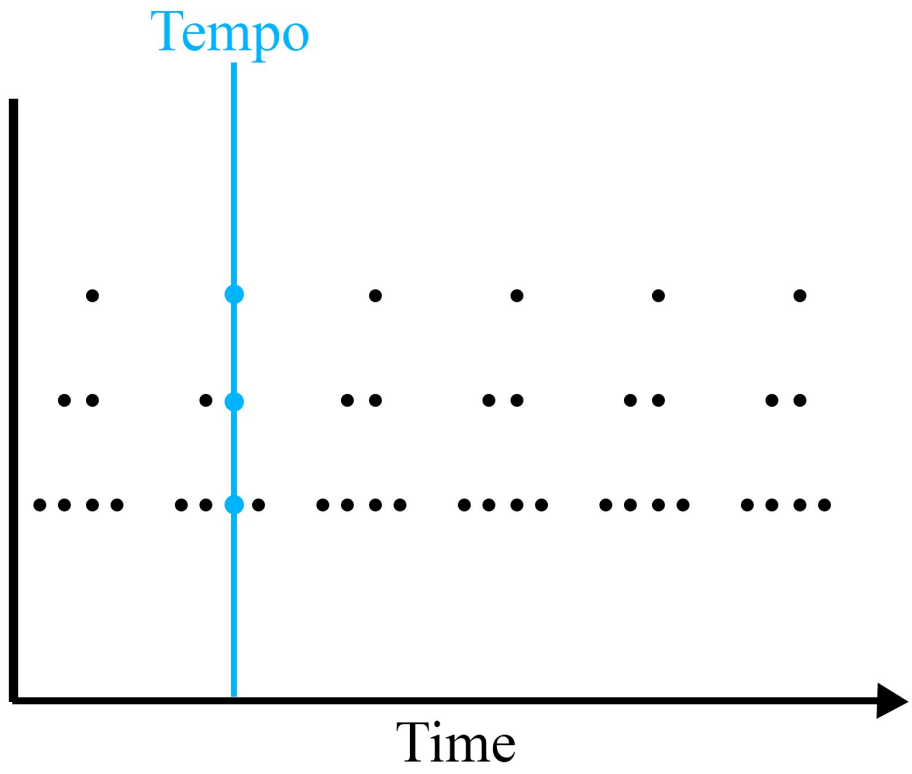


Enemy going around wall

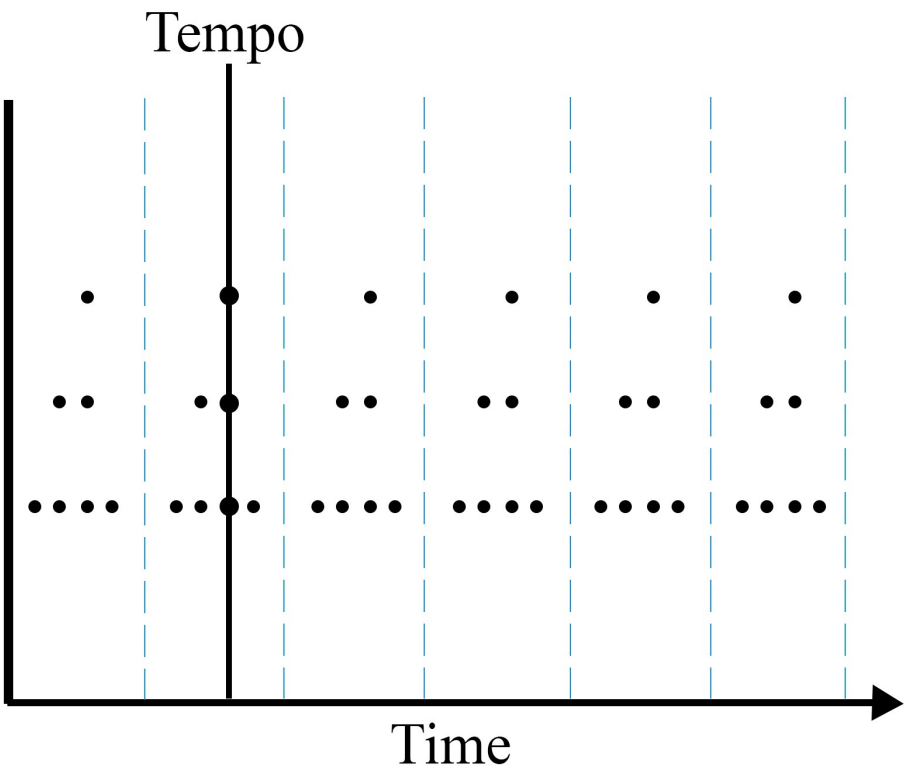


Smarter enemy getting stuck on a wall after two attempts of picking the best path

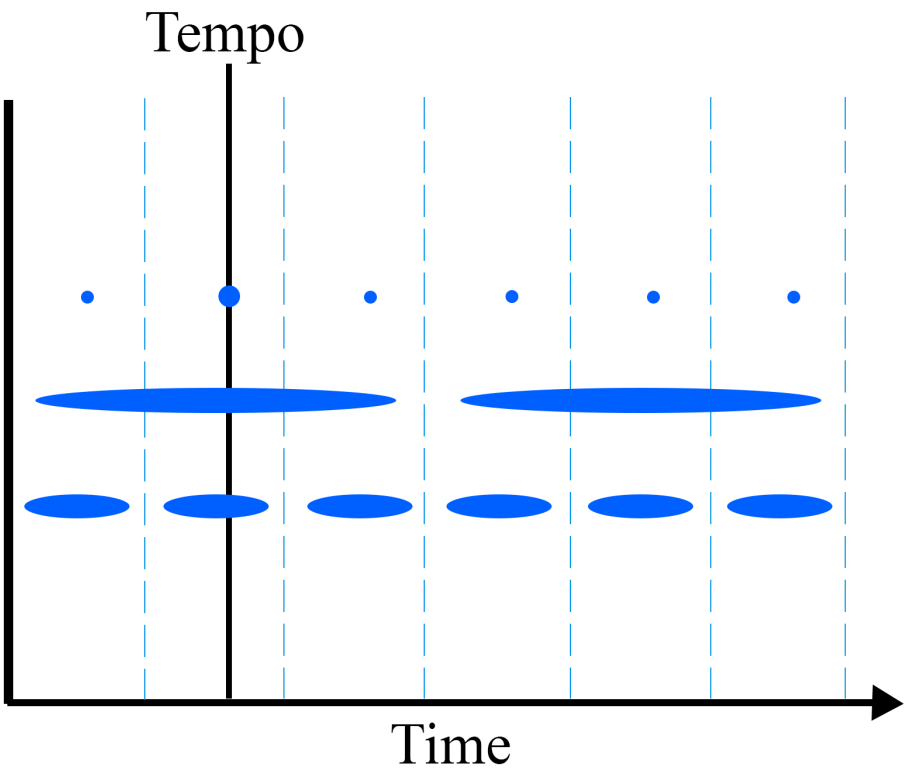
Chapter 8: Generating Music



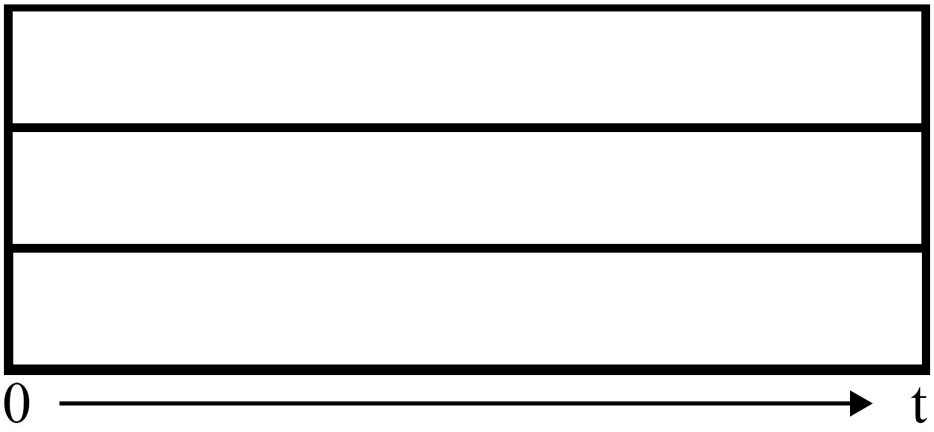
Visualization of tempo



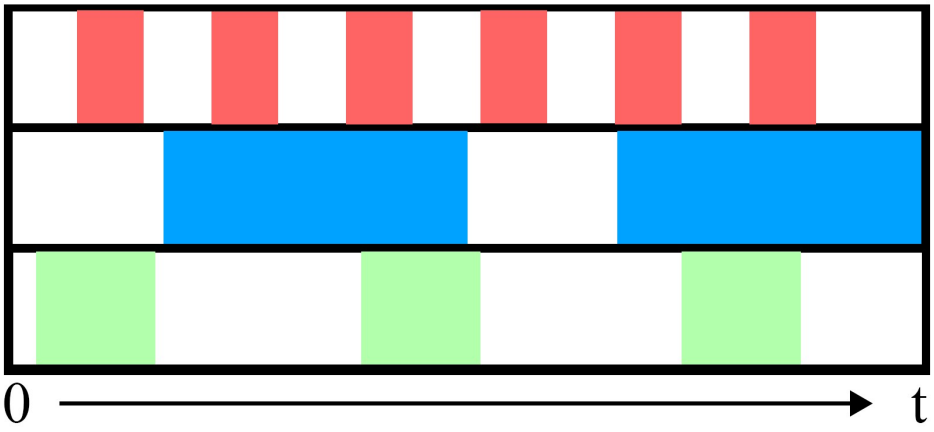
Visualization of tempo divisions



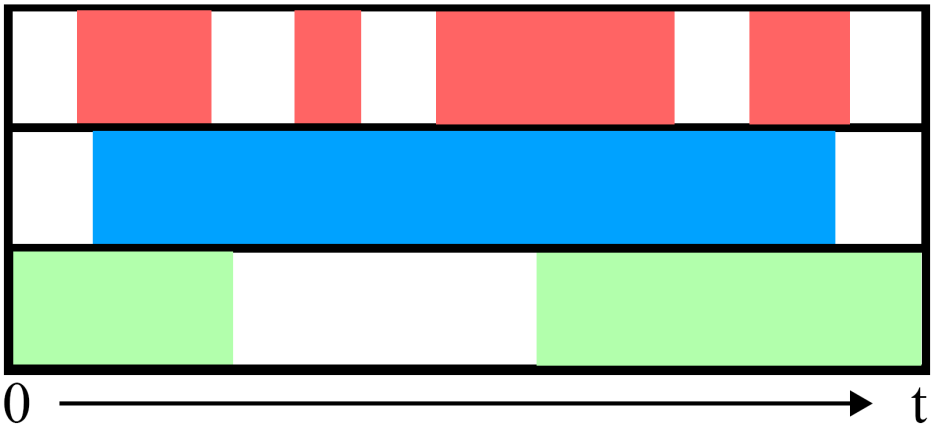
Visualization of sound length variations



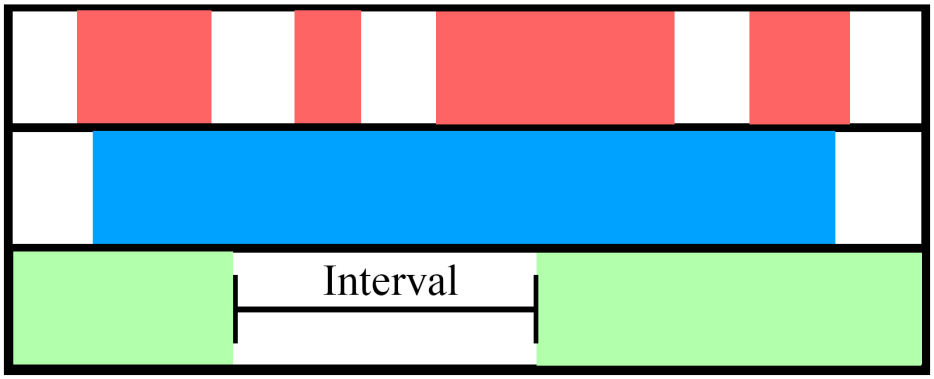
Visualization of the measure with sounds



Measure with sound division

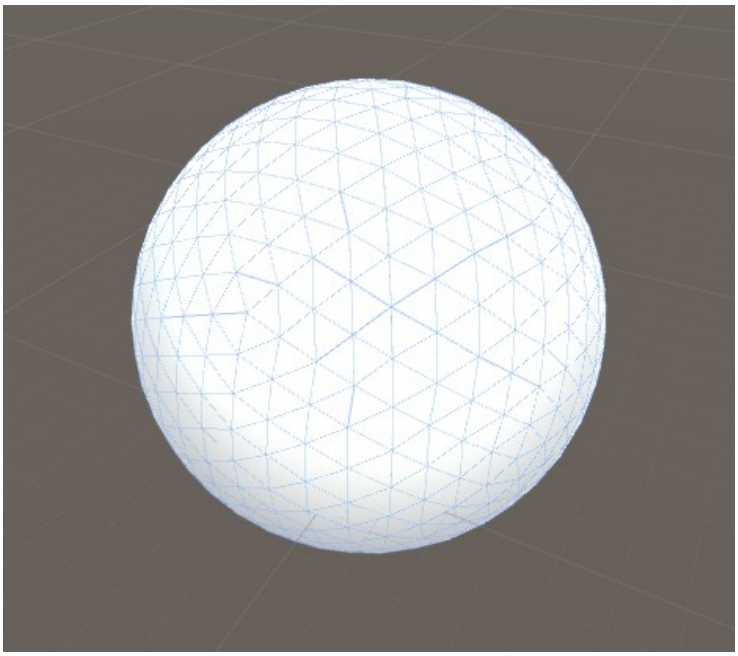


Measure with random sound lengths

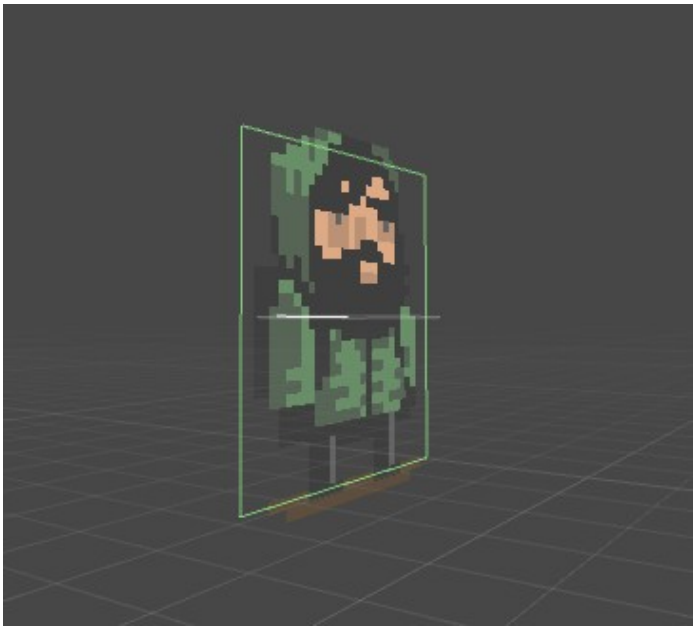


The interval in which sound is not played

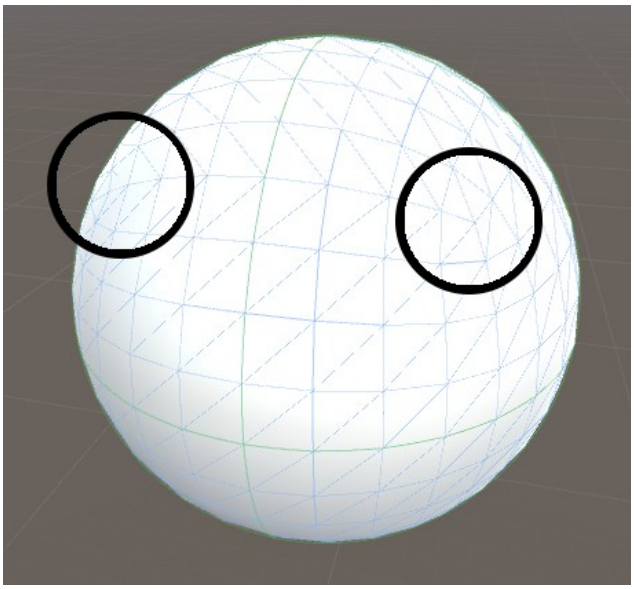
Chapter 9: Generating a 3D Planet



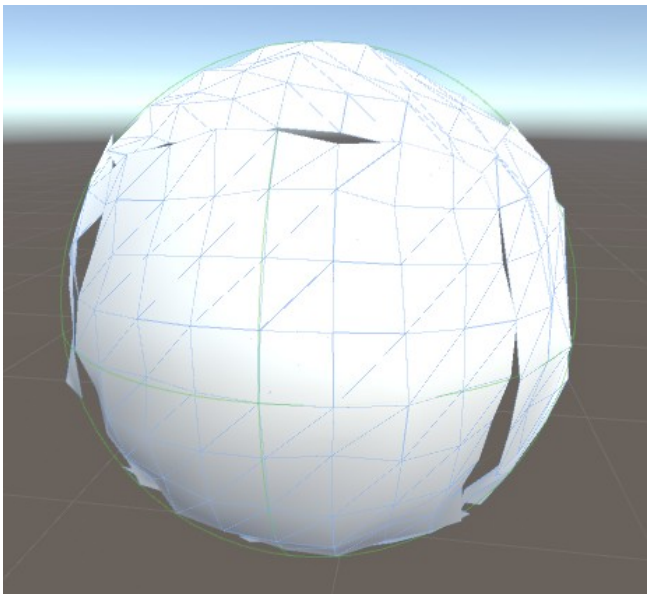
A procedurally generated sphere



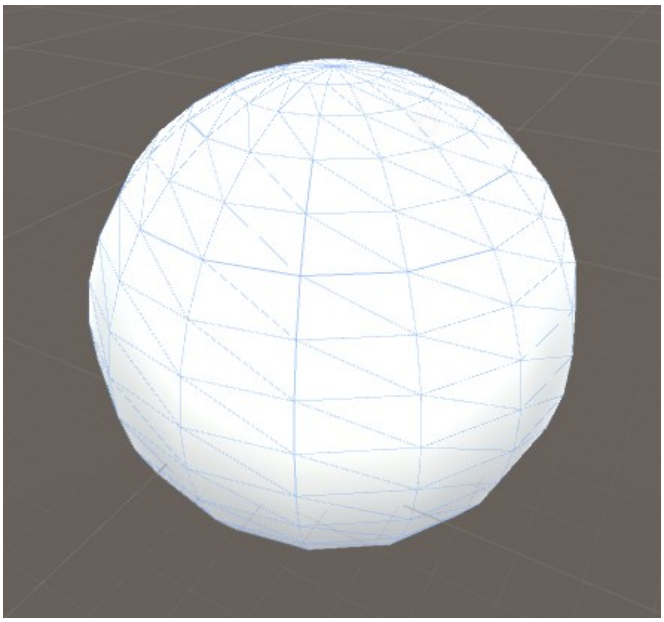
The 2D quad a sprite is drawn on to



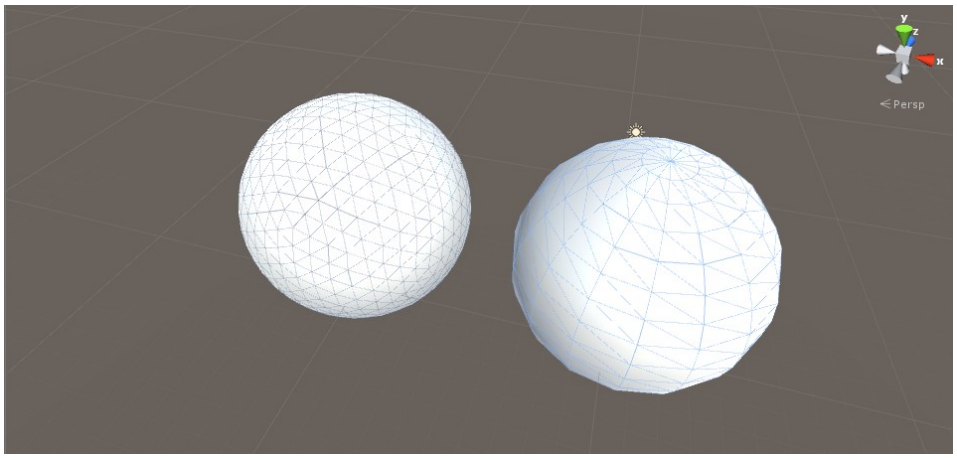
3D cube made into a sphere



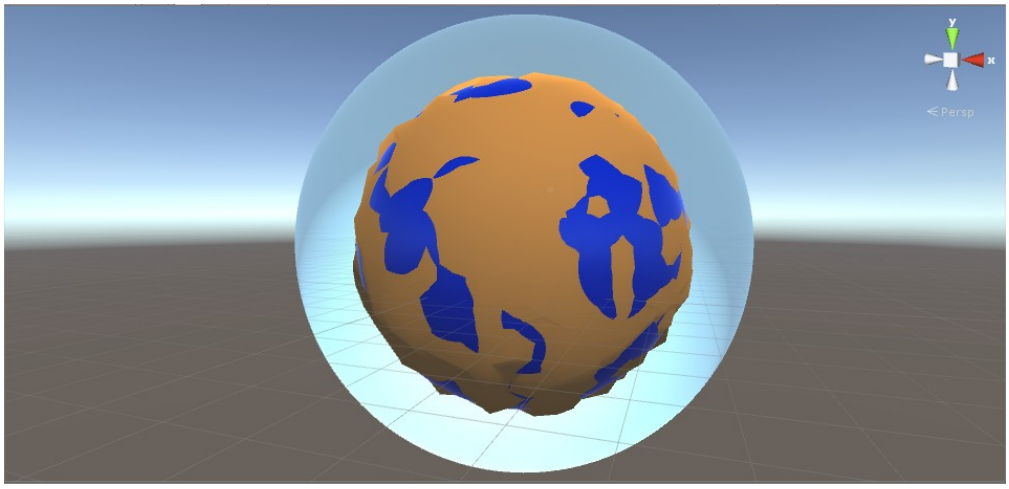
Sphere triangles splitting when moved



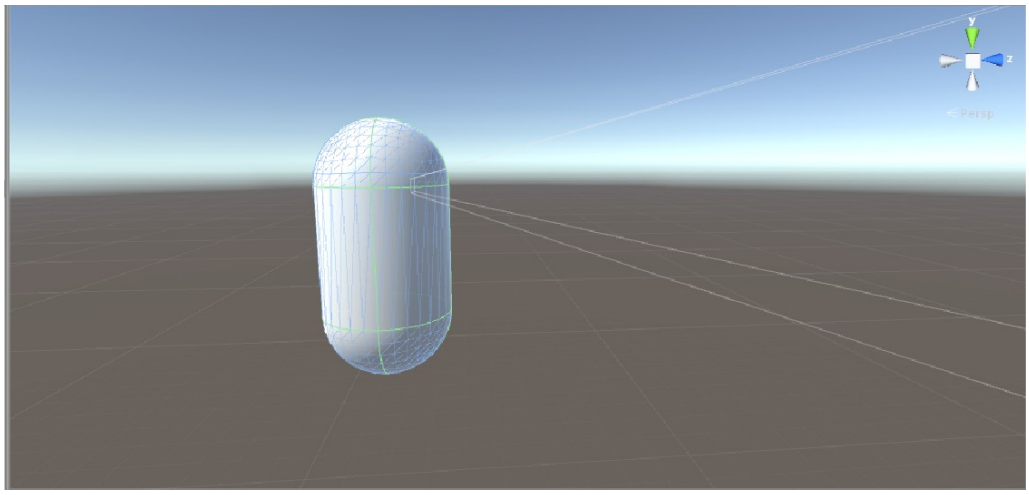
Procedurally generated sphere



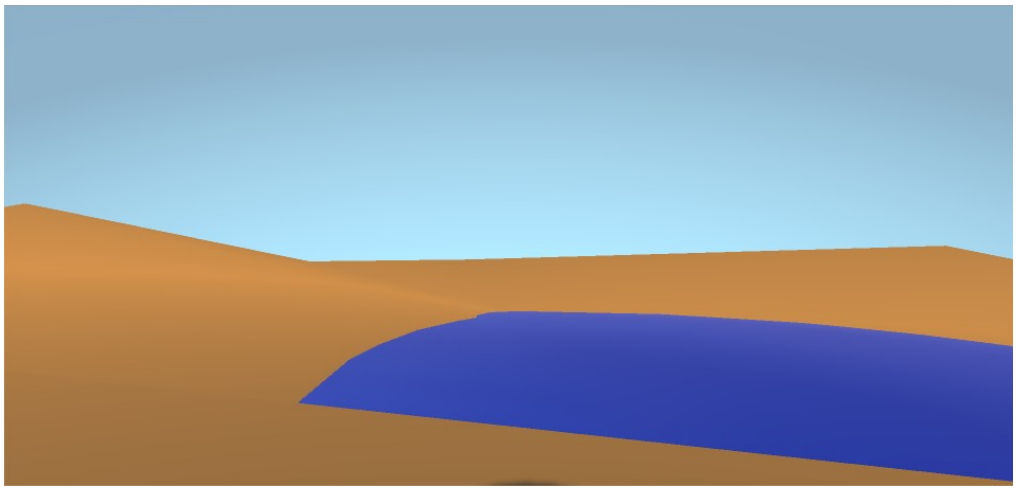
Icosahedron sphere on the left and polar sphere on the right



Procedurally generated planet

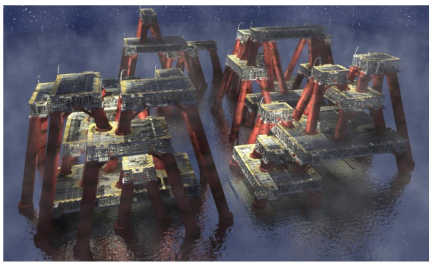


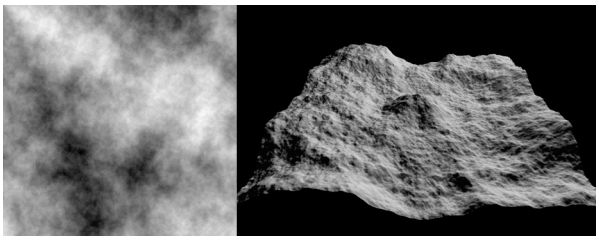
The first-person camera placement

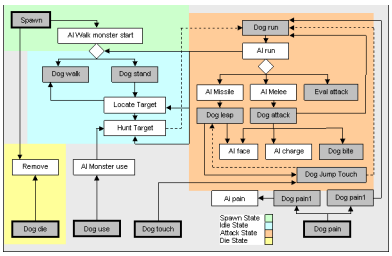


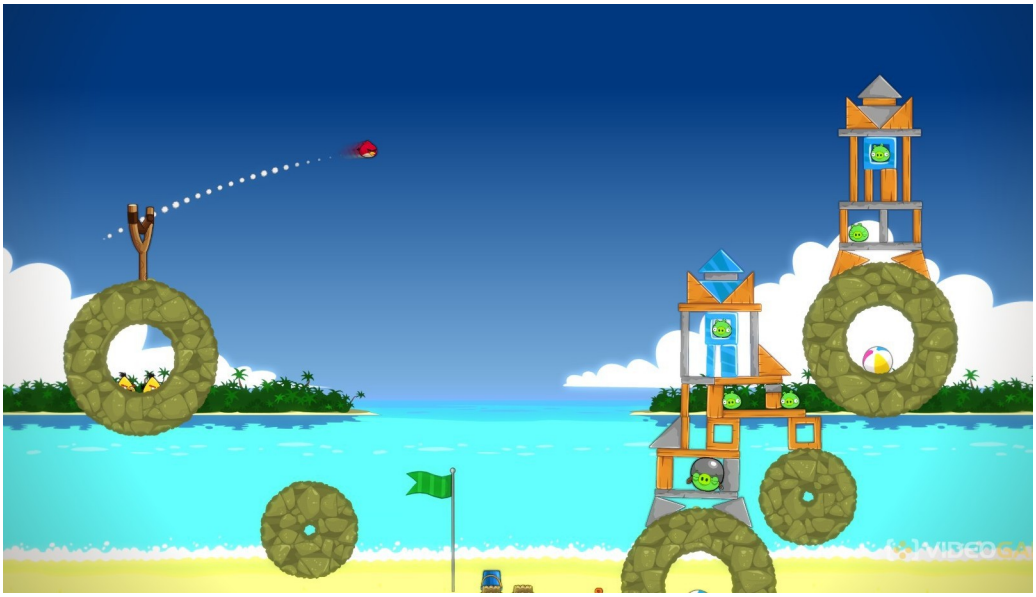
The first-person view of the planet

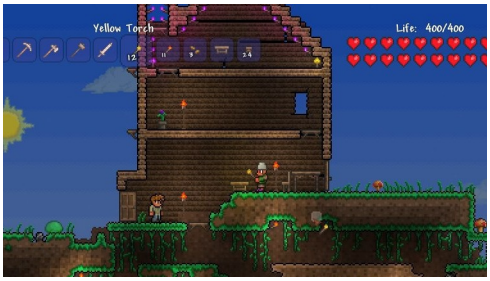
Chapter 10:

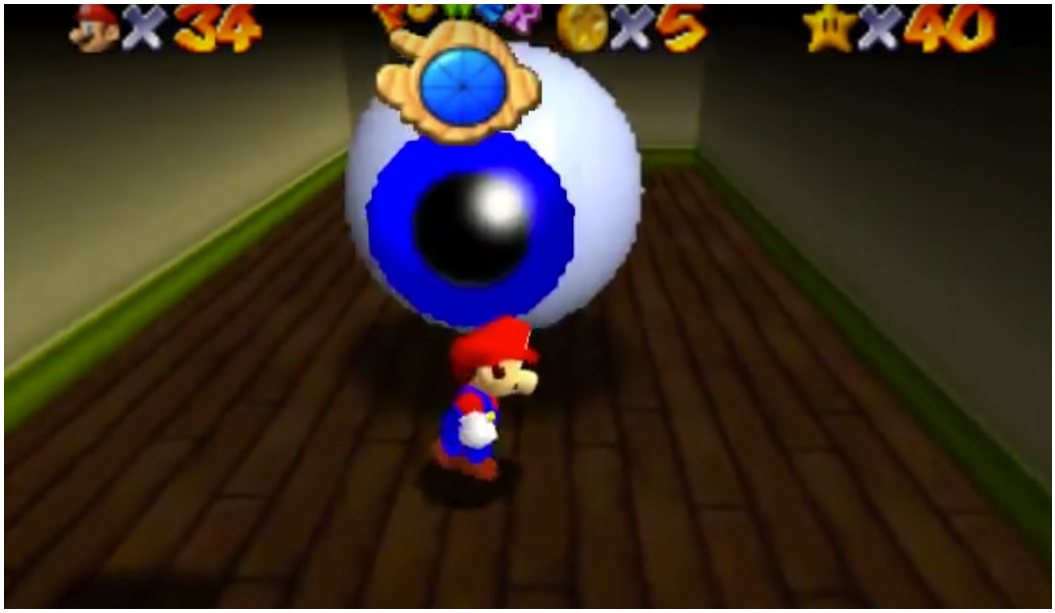





Spawn
AI Walk monster start
Dog walk
Dog stand
Locate Target
Hunt Target
Remove
AI Monster use
Dog die
Dog use
Dog touch
Dog run
AI run
AI Missile
AI Melee
Eval attack
Dog leap
Dog attack
AI face
AI charge
Dog bite
Dog Jump Touch
AI pain
Dog pain1
Dog pain1
Dog pain
Spawn State
Idle State
Attack State
Die State




Yellow Torch
Life: 400/400


X 34
X 5
X 40