



BSc Final Project
Computer Science Department
Reykjavík University
Making a Game
Operations Manual

Supervisor:	Examiner:	Instructor:
Jonathan Pierce - Lead Tools Engineer CCP jonathanp@ccpgames.com	Ingólfur Halldórsson - Lead Developer at Aska Studios ingolfurhalld@gmail.com	Steingerður Lóa - Reykjavík University steingerdur@ru.is

Students:

Ágúst Máni Þorsteinsson	230802-3190
Daði Rúnarsson	051290-2169
Hermann Helgi Þrastarson	310301-2720
Hugi Freyr Álfgeirsson	011299-2029
Júlía Ósk Tómasdóttir	110798-3189

Document description

This document outlines the development environment used to create the game called Cold Bargain. For more information on the methods used to develop the game see the Project Report. This document does not describe the game itself nor how to play it, for that information see the Game Manual.

System Requirements (Recommended)

The following are the recommended specs required for running the project.

OS: Windows 11 / MacOS 15 Sequoia

RAM: 16 GB

Storage Space: 10 GB

CPU: Quad-core 2.5 GHz

GPU: Dedicated (Nvidia GTX 3050 or newer)

Resolution: 1920x1080 minimum

DirectX: Version 11 (Windows)

Installation and Set-up Instructions

Start by downloading the [Unity Hub](#) and install the Unity Editor version 6 (6000.0.36f1)

Open a terminal and navigate to a folder where you'd like to store the repository.

Running the following command will download the source code:

[git clone <https://github.com/HermannHelgi/making-a-game-G13>].

This is assuming you have already installed [git](#) onto your device.

Once you have cloned the repository, open the Unity Hub and press 'Add Project from Disk'. From there select the file location of the repository you have just downloaded.

This will generate the rest of the necessary files required by the unity engine.

Launch the project from the Unity Hub to begin editing the project.

Core scripts and their functions.

Unity File Structure

Within the Assets folder are two main folders. These folders were made to follow the Git Flow architecture, for more information see the Project Report.

First, there is the features folder. Within it are several folders, each of which created specifically for the development of a feature throughout development. When a developer created a new feature branch, the first thing they did was to create a new folder within the feature folder in Unity and name the folder the same as the branch or feature. All work that was done on that feature was within that folder. That is to say prefabs, levels, scripts, scenes, etc.

Secondly, there is the development folder. This folder contained all finished features which were ready to be used for development use. This folder was updated when updating the develop branch.

Aside from these folders there is the packages folder. This folder contains all of the asset packages used for the project. The game used two scenes, the Main Menu Scene and the Game Scene.

AI Architecture

The enemy, the “Wendigo,” was designed and implemented using a state machine for its core behavior. In addition, it uses the NavMesh package for pathfinding, and various predator-prey logic following the integrated states.

Animation & Rigging

Characters were rigged and animated using 3rd party software and assets, but Unity's animator system was also used.

Audio

For audio, the team created and used a custom made sound manager to set up and instantiate all sounds within the game.

Known Issues

Some font files might get corrupted and create merge conflicts. Git LFS was used during development and might create issues as well, such as causing problems with scene loading. These issues were either out of the developers hands or the team couldn't find a resolution to the problems.