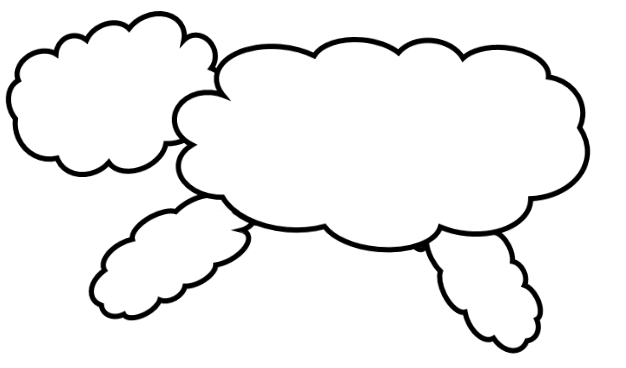
Project Sheepherder Technical Design Document



This page: Table of Contents and Team Member Listing

|  |  |  |
| --- | --- | --- |
| Table of Contents  Executive Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Game Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Technical Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Equipment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Hardware . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Game Engine . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Target Platform . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Scheduling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Development Plan . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Milestones . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Updates, Maintenance & DLCs . . . . . . . . . . . . . . . . . . . . . . . . . . .  Work Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Remote Collaboration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  File Formats & Naming convention . . . . . . . . . . . . . . . . . . . . . . . .  Levels . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Level 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Asset List . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   |  | | --- | |  | | Game Development Team Members    PRODUCER  Garrison Price    PRODUCTION MANAGER  Garrison Price    PRODUCTION COORDINATOR  Jacob Golden-Needham  GAME DESIGNERS  Blake Myers, Garrison Price, Jacob Barger, Jacob Golden-Needham  SYSTEMS/IT COORDINATOR  Jacob Golden-Needham  PROGRAMMERS  Balke Myers, Garrison Price, Jacob Barger, Jacob Golden-Needham  TECHNICAL ARTISTS  Blake Myers, Jacob Barger  AUDIO ENGINEERS  Jacob Barger, Garrison Price  UX TESTERS  Blake Myers |

|  |
| --- |
|  |

# 1 Executive Summary

Game Overview

Herding hysteria is a third person herding game where the player is a shepherd trying to collect and guide his flock. The shepherd guides his flock with the aid of his faithful dog away from dangers ranging from environmental challenges like cliffs and divides to predators, and even the wrath of the gods. The player must command his dog to control the flock and shout to scare off predators to prevent members of the flock from being killed.

Technical Summary

Herding Hysteria will be developed in approximately one month by four people using the Unity game engine. For 3D asset creation Blender 2.81 will be used, with FireAlpaca for texture work. The total production cost of the game will be $0.

The game will be deployed for PC.

The minimum requirements include:

Nvidia Geforce 940m or Equivalent

Intel i5 or Equivalent

4GB of RAM

XGB of Disk space

# 2 Equipment

Hardware

Members of the team will be utilizing different personal equipment for all game development and asset creation.

Software

All software utilized during production will be able to produce high end visuals and effects while still being able to deploy across different platforms should we choose to do so. Additionally, all software being utilized is either free to acquire, or was already owned by a member of the team.

# 3 Evaluation

Game Engine

The game engine utilized for the development of Herding Hysteria is Unity because it allows us to implement basic functionalities of a 3D game with relative ease and then build off of that with more advanced game development techniques. It helps facilitate the beginnings of game development while not placing a ceiling on the complexity or optimation possibilities later on. Unity also allows us to deploy to many target platforms should we choose to do so later on.

Target Platform

Herding Hysteria will be deployed to PC only. This helps narrow the focus of development and allows us to not have to implement multiple control schemes, or having elements of the design work equally well across multiple platforms(ie UI working well on PC or console, but being difficult to read on mobile). This is also the platform where the bulk of the game’s audience would likely be located due to its somewhat nontraditional nature. It is not a pure puzzle game, and does not feature standard forms of combat. Game’s with a somewhat unique premise like this typically can do better on the PC market where games without a clear or traditional genre often shine.

# 4 Scheduling

Development Plan

Original development plan was to have the core systems(procedural levels, sheep boid ai, and companion dog) working by the end of March. Then the focus would be to have a networked multiplayer component where the second player either controls the dog, or becomes a god that can choose to torment or aid the player. From there enemy ai could be added, and art assets could be generated. However, this schedule has been drastically thrown off by changes due to COVID-19.

Milestones

4/21/2020 - Prototype testing

Updates, Maintenance & DLCs

# 5 Work Environment

Remote Collaboration

Due to social distancing and self isolation requirements all collaboration for this project has become remote. The team has been utilizing discord to communicate, and was using gitlab from day one to work on different branches of the same project to safely iterate on a common project remotely.

# 6 File Formats and Naming Convention

The team has chosen to stick with the standard naming conventions for development in C# for Unity

# 7 Levels

The levels of Herding Hysteria will ideally be procedurally generated and utilize forms of perlin noise to generate unique landscapes. The levels will include spawn points for sheep to collect, hazards to avoid, and enemies to fend off. However, if procedural level generation becomes a problem, design could be shifted to include at least 3 pre generated levels that utilize procedural techniques like perlin noise in their initial construction. Regardless the levels will be natural terrains(hills, mountains, rivers, canyons).

Asset List

Players: Shepherd, God(potentially)

Enemies: Wolf

Allies: Dog, Sheep

Props:

Environment: ground, sky