

Sheep Herding Game - Design Document

Elevator Pitch

Embark on a whimsical sheep-herding adventure, where strategic thinking meets pastoral charm. Guide your flock through picturesque landscapes while overcoming challenges to lead your sheep to safety in this unique and accessible game experience.

Summary

With a top-down view of a shepherd's field, players take control of a sheepherding dog in a sheepherding competition. The player will have to work to wrangle the sheep into their pen as fast as possible. The sheep can be disobedient and willful, and so it takes planning and quick thinking to get the dog to herd the sheep in a competitive time. Additional challenges will see the player manoeuvring around obstacles such as rocks, trees, and streams - and as the player progresses in skill, more difficult environments to navigate the sheep through will become available.

Unique Selling Points

- **Canine Companion Perspective:** Immerse yourself in the role of a sheepherding dog, offering a fresh and captivating perspective on the herding genre.
- **Dynamic Challenges, Strategic Gameplay:** Navigate diverse landscapes with evolving obstacles, utilising the dog's unique skills for strategic herding, ensuring a dynamic and engaging gameplay experience.
- **Evocative Environments:** Herd sheep through beautiful, hand-crafted environments that challenge the player with obstacles such as rocks, trees, and streams, providing varied gameplay in a sequence of levels.

The Team

Name	Role
Ahmad Beshar (AB)	Game programmer, game design
Hadar Sharon (HS)	Game Programmer, Assistant Producer, Sound/Audio Design, Level Design

Hunor Szekacs (HSz)	Assistant producer, game programmer and tester
Kyle Holmquist (KH)	Game Design, Game programmer, Assistant Producer
Zoe Sklair (ZS)	Producer, Game programmer and tester, game design, UI

Prototyping

Stage 1: Planning the prototypes

To begin our development process, we started by identifying the key components of the game, and what elements would need to be developed and iterated on in order to fine-tune the user experience and game mechanics. The core of the game is the sheep and dog controls, so these are the primary programming challenges of the game to overcome.

The sheepdog is the player-controlled character, and so the user controls will determine the feel of the movement and the user experience. Whether the dog is moving or not, which direction it's moving in, as well as barking in the direction it faces, all need to be controlled.

The sheep move on their own, and we want a combination of them stopping to rest or graze, as well as moving around of their own will. The sheepdog's movements and barks will affect their movement too.

An important decision we discussed at this stage was how the movements of the characters should work - either using transforms or physics simulation.

Transforms:

- Easier to move the sheep
- Smooth movements with sheep moving at constant speed and immediate starts and stops - more chaotic feel which could be fun/challenging for the player
- However, game objects do not collide so collisions between sheep and objects will need to be handled manually

Physics simulation

- With sheep having the rigid body component, collisions can be dealt with by Unity's physics engine
- The sheep accelerate and decelerate in order to start/stop so the movement looks more natural
- The sheep will probably not move as much or as quickly if they are slowed down/stopped by collisions which may mean there isn't as much of a challenge of speed and chaos with regards to managing the sheep

Stage 2: First programming - initial prototypes

We started the game's programming by implementing prototypes of the sheep and dog movement. This was an opportunity to try different control and gameplay styles to see how they felt.

For the sheepdog's controls, we started with a combination of mouse and keyboard for the direction and movement respectively, with the dog pointing at the mouse cursor, and basic WASD for movement.

The sheep movement was first implemented using transforms with some experimentation on how quickly they should move and change direction. However, the main concern was how collisions would be managed, as part of the game involves sheep being blocked by fencing, obstacles, and even other sheep.

So, after discussing this, we decided that using the physics engine would be a better course of action, so we could focus on the relationship of the sheep and dog's movement, rather than manually detecting collisions.

The more realistic and natural-looking movement from using physics could also be more appealing to interact with.

Stage 3: Second programming - new sheep movement

The next round of programming focused on the new prototype of the sheep's movement. We attempted a more realistic version of sheep behaviour, to include periods of grazing, resting, and walking around when not near the sheepdog, rather than the more cartoonish behaviour of sheep constantly moving around chaotically. This was achieved using coroutines and random timers, to manage the periods of different types of behaviours.

We also developed the code for defining the "zone" the sheep all need to be in, in order for the player to complete the level, which involves tracking which sheep have entered or left the zone.

At this stage we also discussed how to keep the game challenging, and so future prototypes will need to include testing of parameters which control the sheep's level of disobedience, such as how far they will run after being barked at before returning to grazing, and how reliable the directions will be as the player tries to control the sheep movement.

Stage 4: Third programming - dog and sheep interaction and movement improvements

There were some issues with the previous sheep and dog movement; for instance, they would sometimes flip over when colliding with each other at certain speeds or angles, so the next stage of prototyping included fixing these issues by constraining the sheep and dog's rotations to the y axis and fine-tuning the movements further. Sheep movements are based on random timers, whose minimum and maximum bounds were landed upon iteratively by exposing the values to the team and having them tweak as necessary based on what felt best.

We also started the implementation of the dog and sheep interactions. The dog's bark affects sheep. The sheep get the dog's vector and move in the opposite direction, which forces the player to take care while using their bark.

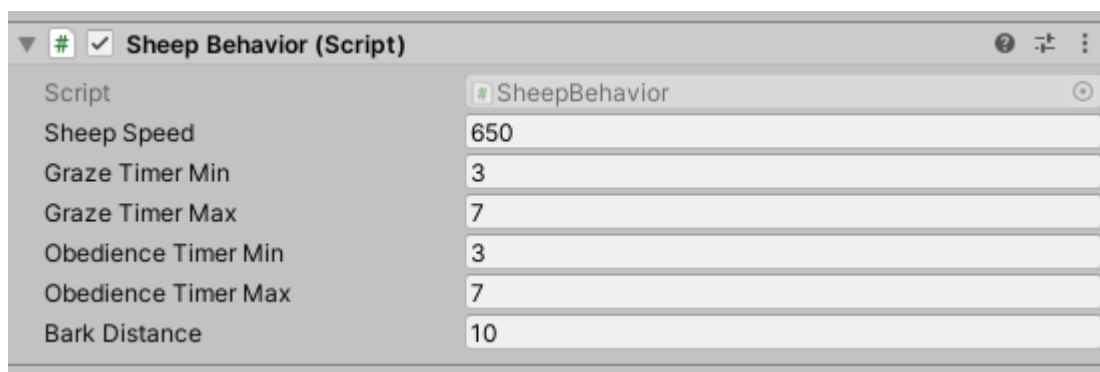


Image: Exposed Sheep Behaviour Variables



Image: Prototype Level - Game Start



Image: Prototype Level - Leading Sheep to Pen



Image: Prototype Level - Success State

Stage 5: Next steps

At this stage, we have the first implementation of the main game mechanics in place. Next steps will involve creating prototypes that we can release to external testers in order to gain feedback on controls, game feel, and user experience. For example, how challenging the process of herding the sheep is, and if it feels comfortable, intuitive, or frustrating to play.

We can also develop the level designs, and plan types of obstacles and how they will be incorporated into the environment. There are also other features of the game, such as the timer and points calculator for the high score/competition narrative of the game, as well as the UI and sound design.

Key Mechanics

- The key mechanics of the game are the sheepdog and sheep movement, and how they interact.
- What the player does during the game is control the dog's movement, direction, and barks, in order to manipulate the sheep's movement.
- The sheep will respond to the player's actions, by avoiding the dog and running away from its barks.
- The game's experience will depend on how the dog controls feel, and how the sheep behave - how difficult they are to control due to their individual level of predictability, and the task of getting them all into the correct "zone" at the same time.
- The win conditions of the game levels are met when the sheep are all in the correct area at the same time, and the points/score of the player are determined by the time taken to do so.
- The relationship between the player-controlled dog and the behaviour of the sheep will hopefully make the game the right combination of challenging and enjoyable, and provide satisfaction when sheep are herded into their zone.

Characters and Setting

The player character is a sheepdog, whose job is to herd the sheep (non-playable characters) using the dog's own movement and barking controls. The sheep are not your typical obedient creatures, and once away from the dog's vicinity, will go back to grazing. They are also unpredictable creatures, and may run off at random angles instead of following the dog's bark directly.

It's the sheepdog's job to work with the characterful sheep under its care, to guide them to their new pasture safely and in a timely manner.

The game is set in a shepherd's field, where the sheep roam and graze freely. It's time to get the sheep into their new pasture, so the sheepdog needs to herd them across the environment into the next fenced-in area, including over streams and around rocks and trees.

Not only do the sheep need to be herded safely across the field, it needs to be done fast! The sheepdog is being entered into a herding competition, and the faster the sheep are shepherded into place, the more points will be earned.

The player needs to help the dog do its job, herding the sheep to their new pasture, and doing it as quickly as possible, for the most points.

Beginning the Game

To start the game, we will have a simple introductory level which will act as a tutorial. There will be some hints on screen guiding the player on the controls and the goal of the game, but the level will be basic enough that the player should be able to pick up the requirements quickly.

First 2-5 Minutes of Gameplay

The game will start with a basic herding level, with only a few sheep, and a simple environment to navigate. This allows the player to focus on getting comfortable with the controls, and how the herding mechanics work, without it being too difficult to complete. This first level will include directions of how to start herding sheep using the dog's movements and bark control, and which part of the scene the sheep should be guided to. Once the first level is completed, the game can then progress to the next level, with more sheep and an environment that includes obstacles that need to be manoeuvred around. The later levels will increase in difficulty, with harder-to-navigate environments, more disobedient sheep, and a larger flock to manage.

Art Style

The art style will be simple and more cartoonish than realistic. The appeal of the game is the whimsical setting and impish sheep, so the art style will complement this with simple but cute and characterful designs for the sheep and sheepdog. The environments will match the non-realistic style, but will still be picturesque to capture the feeling of nature surrounding the sheep.

In particular, it will feature countryside visuals such as: fields, hills, valleys, steppes and other such vegetated areas.

Obstacles could therefore correspond to the respective environment - for instance, a grassland could feature rocks or a stream, occasionally blocking or hindering movement.

Audio

The game's audio design will be an integral part of the art as a whole, and will be heavily influenced by the player's current environment and landscape (or vice versa). The game will feature both background music and sound effects.

By binding the background music and sound effects to the environment, we could create a more immersive experience for the player. The visuals and audio should play together to combine a certain feeling or experience for the player to carry with them as they are playing.

For example - a pastoral, idyllic landscape could feature laid-back and pleasant music, to give the player a peaceful feeling corresponding to the calm environment. Likewise, we could design various soundscapes (or themes of such) to convey a range of different feelings, directly corresponding to the environment throughout the game: from menacing to ethereal.

The sound effects of the game, to be played on top of the background music, will comprise a collection of animal, nature, and other ambient sounds that are both immersive and responsive.

For instance, the sheep in the herd will freely interact with one another as they graze, to simulate the sound of a true herd, even without the player doing anything - however the dog (herder) will bark or howl to command the sheep, in direct response to the player's input.

Examples of sound effects:

- Dog Bark
- Dog Steps
- Dog Howl
- Sheep Bleat
 - Bleats will feature a variety of pitches, moods, volumes etc.
- Wind Blow
- Birds Chirps
- Insects Trills

In practice, music and sound effects will either be recorded ad-hoc by the audio/sound designer, or we will incorporate existing third-party assets as we see fit.

SWOT Analysis

- Strengths:
 - Fun, basic game that's easy to start and get going with.
 - Simple, evocative art style.
 - Cosy game with light-stress challenges.
 - Evolving obstacles and learning behaviour.
 - Extensible and modifiable.
 - Child-friendly, family-friendly, accessible.
 - Casual but still offers a fresh perspective.
- Weaknesses
 - A niche game
 - Top-down view might be a limitation
- Opportunities
 - Combine a compelling storyline with casual, laid-back gameplay
 - Great art style - eye-catching or aesthetically pleasing art goes a long way (also including background music and sound effects)
 - Interactive environment that's fun to explore (the environment reacts to you or has a life of its own)
 - Character arc(s) - standout or individualistic behaviour within the herd (e.g. some herd members are different to others or grow to display certain behaviours along the way)
 - Player achievements - make the player work for earning various progress and/or challenge-based prizes
- Threats
 - Game might not be special or compelling enough to play at first glance
 - Game might be repetitive or basic, leading to early churn
 - Environment might be either too hard or too easy to navigate, obstacles might be too lax or too tricky - therefore difficulty and playability should be well-coupled and well-adjusted, respectively
- Potential problems:
 - Making the game unique enough in comparison to other sheepherding games.
- How will we deal with problems if/when they arise?
 - Discuss with the team and make decisions about what to do next.

Production Schedule

Date/Week Commencing	Tasks
23 November	<ul style="list-style-type: none"> - Choose game - Decide on key mechanics - Decide on artstyle and sound design - Setup git repository (HS)
27 November	<ul style="list-style-type: none"> - Set up and learn basics of Unity - For those comfortable with Unity, start test implementations of dog and sheep movement (KH, ZS) - Start collecting assets for environment and characters (HSz, AB)
4 December	<ul style="list-style-type: none"> - Review prototypes of dog and sheep movement - Reimplement sheep movement script using physics/rigid bodies (KH, ZS) - Test dog controls (HSz, AB)
18 December	<ul style="list-style-type: none"> - Implement dog bark (KH) - Implement sheep zone/win conditions (ZS) - Test out music ideas (HS) - Fill out design doc sections (HSz, AB)
1 January	<ul style="list-style-type: none"> - Ensure design document is complete (All)
8th January	Midterm deadline / break
15th January	<ul style="list-style-type: none"> - Evaluate progress - Prepare prototypes for external testers - Come up with feedback questions - Release to testers
22nd January	<ul style="list-style-type: none"> - Level designs (HS, AB) - Build basic level environments (HSz, KH) - Begin UI designs (ZS) - Evaluate feedback
29th January	<ul style="list-style-type: none"> - Implement changes from feedback (dog controls, sheep movement) - Sound Design (HS) - UI Design (ZS) - Timer and points system (AB) - Character assets, animations, effects (KH, HSz)
5th February	<ul style="list-style-type: none"> - Advanced level designs (HS, AB) - Build level environments (HS, AB, KH) - Test levels (ZS, HSz)
12th February	<ul style="list-style-type: none"> - Get external feedback for levels - Continue polishing game/missing elements
19th February	<ul style="list-style-type: none"> - Implement feedback

26th February	Leeway week for delays / problems that need fixing
4th March	- Ensure game is ready for submission
11th March	Project Due

List of Planned Assets

Environment:

- terrain, fencing, trees, rocks, streams/ponds
- static models, some may have visual effects
- will use third party assets for these

Characters:

- sheep, sheepdog
- models with animations for movement
- may use third party assets or develop them ourselves if not too complex

User Interface:

- main menu, hints for tutorial level, timer during level, high score menu
- plan to develop these ourselves with Unity tools

Sounds/Audio:

- sound effects for sheep for grazing, moving, bleating
- sound effects for dog for moving, barking
- background/environmental/ambient sounds
- music for main menu and levels
- we can use a mix of third party assets and our own creations

Levels:

- created using environment assets and designed ourselves
- plan to create 5 levels of varying difficulty/complexity depending on the number and type of obstacles

Code elements:

- dog controls
- sheep movement/behaviour
- dog and sheep interaction
- sheep tracker (check for winning state)
- timer/points calculator

Bare essentials:

- levels/environments with fencing and obstacles
- the sheep and dog assets with implemented mechanics
- sheep tracker/win conditions
- the timer to track how quickly the level is completed
- sound of bark
- UI to display core information about time taken, score achieved, level selection, and tutorial hints

“Nice to have”s

- environments with visual effects
- different animations for sheep behaviour (e.g grazing and resting rather than just stopping movement)
- animations for dog (e.g tail wagging, head moving for bark, panting while idle)