Lucid Design Document

Designed by Big Dreamers

Team

Brian Eisenberg - Design Lead and Programmer
Jacob Frommer - Scrum Manager and Environment/Shader Designer
Nevo Mantel - Lead Programmer and AI Designer
Diego Holguin - Procedural Generation Programmer
Connor Dean - Lead Level Designer and Shader Designer
Colin Winders - Lead Character Artist
Manny Gomez - Lead Texture Artist and Environment Artist
Nick Pinero - Lead Animator and Character Artist

Overview

Lucid is a roguelike, top down, combat game charging the player with exploring the possibilities and limitations of lucid dreaming. You will be charged with journeying through your subconscious in order to make sense of the world. There is more than meets the eye in this realm as a masked figure seeks your help, and a dark entity pursues you, attempting to torment your dreams.

The player will progress through dreams, combating enemies with unconventional tactics. There is a delicate balance to the dream state. Overexerting your control risks breaking the fragile balance of the dream and causing it to crumble. Players will have to consider their strength and patience to overcome the ever growing challenges with each new area.

We strive to remember our dreams. We claw at those that slowly fade away. The dreams you cannot complete, whether through death or crumbling, will be lost to the recesses of your mind. The dreams you complete, will be logged and can be returned to at any point. Lucid expands on the roguelike genre by giving the player the chance to save dreams and return to them without the risk of losing the progress forever. In Lucid, you will learn the balance of the dream state, and master control of your mind.

Game Terminology

Dream - A level in Lucid set in the dreamscape of the main character. A dream is a grid of procedurally generated areas (**See: Area**). The player starts in one, and must travel to the center area to complete the dream.

- **Saved dream -** A dream that the player completes is saved in the player's dream journal
- **Lost dream -** A dream that the player destroys by overpowering the dream state with his abilities

Dream Journal - A list of saved dreams the player can replay from the bedroom.

Area - The Environments that make up a dream. They are separate from one another but have connectors linking to adjacent areas.

Equilibrium - The balance of the dream state. It is shifted through overexertion of your abilities. Too much overexertion leads to **crumbling**

Crumbling - The destruction of the current dream. **See: Lost Dream**.

Nightmare - Each area has a chance of becoming a nightmare. A nightmare is grimmer and harder. An imposing enemy will assault the player as well.

Bedroom - The main menu of the game. Contains the **dream journal.**

Tethers - Interactable objects located in some areas that rebalance equilibrium

Gameplay

In Lucid, the player enters a dream state, where he or she fights various enemies, while trying to find his or her path through a procedurally generated dream world. Each dream consists of a grid of areas. The player begins the dream with two randomized abilities, with which he will be required to create a strategy to fight enemies and survive the dream without being defeated or overexerting the mind and crumbling the dream.

The player's abilities are used to combat enemies. They can be applied directly to the enemy, though numerous object in the environment can also be affected. Abilities can be charged to an upper limit. The greater the charge, the higher the ability intensity. Higher intensity attacks have greater effects on enemies and the environment but will also affect equilibrium. Too much exertion creates an imbalance in equilibrium and leads to crumbling.

The goal of the player is to find his path through the dream. Challenges await in each area. After completion of an area the player will travel through a connector to the next. Upon entering a connector, the player will be given an overhead view of the dream to better understand what direction the center is in. If the player completes the center area, he can exit the dream. This is the equivalent of allowing a dream to run its natural course. Completed dreams, are saved in the journal, and can be replayed at any time. Returning to dreams offers the player a chance to relive fond battles and ability combinations.

There are two lose states through gameplay. First, death in combat results in being respawned in the beginning of the dream. Enemies will be stronger and you will have to play more wisely. Second, disrupting equilibrium and crumbling a dream destroys it forever.

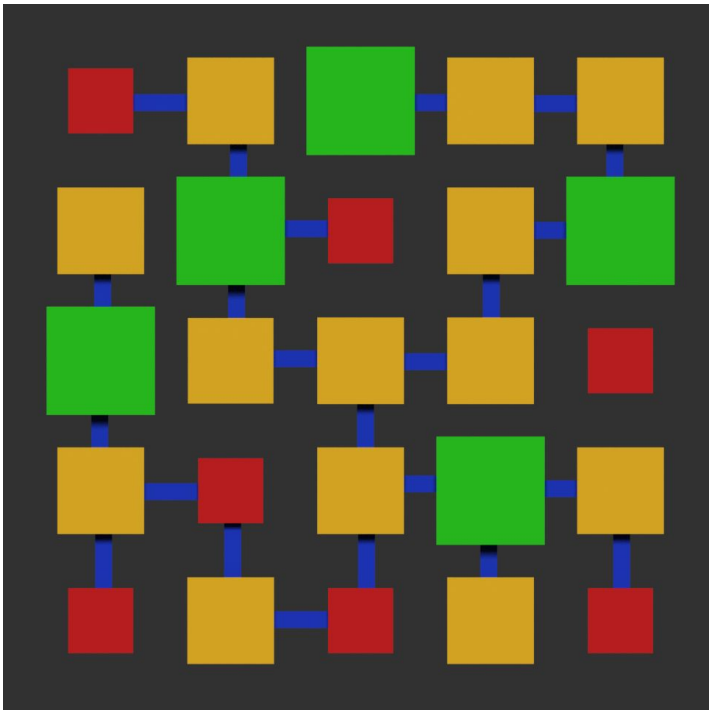
Game Elements

Dreams

- Entering a Dream
    - The hub menu of the game will be the main character's bed
    - Beginning a dream will assign the player two abilities and procedurally generate the level. The abilities given will determine the possible layouts of areas.
- Saving a Dream
    - Upon completing a dream, the dream is saved in your dream journal. Once a dream is saved, you can return to the dream from the journal.
- Losing a Dream
    - If the player's abilities become too powerful for the player to believe in their dream, the dream world will begin to crumble. Once the dream's structure is no longer sustainable, the dream world will be lost. The player will be exited out of the dream back to the bedroom. This dream is **gone** and the player must go back to sleep and start a new dream.
- Dream Journal
    - The dream journal contains all of the dreams that the player has saved. The journal is located in the sky of your bedroom. Upon exiting a dream, the player wakes up in his bed and looks up at his journal.

Areas - Areas are what dreams are made of.

- Size
    - Areas can be normal size (x1.0), small (x.75), or large (x1.25) As Illustrated in the image below
- Format
    - Each area will offer the player an objective to engage in either an environment puzzle, combat, rescue mission, or nightmare. Alternatively, some may act as neutral rest areas.
- Connectors
    - Areas can be linked to adjacent areas by connectors as illustrated in the image below



Equilibrium - The balance of the dream state

- Disrupts equilibrium
    - Increasing the intensity of your abilities
    - Killing neutral npcs.
- Balances equilibrium
    - Slowly balancing over time
    - Power is inversely proportional to balance. The stronger you are, the less balanced the dream.

Player Abilities - As the player uses his abilities, they will grow in strength and in influence on the environment.

- Push/Pull
    - The player can push pieces of the environment at various speeds. This will be used to reveal parts of the map, in combat to squish enemies, and move platforms.
- Gravity control
    - Used to change raise or lower the gravity in a certain area, this would allow the player to raise and lower platforms as well as the ground itself.
- Dash
    - An omnidirectional burst of momentum granting immunity during the motion
- Mindcontrol
    - Turn an enemy into an ally and have him target other enemies for a limited time

- Telekinesis
    - Grab and throw objects in the environment
- Traps
    - Explosive traps that enemies can activate, sticky traps that force things together, attraction trap that pull things toward it.
- Quick Draw/Shot Lock
    - Lock on to several enemies then launch projectiles at them all
- Sacrifice
    - Use neutral npcs to destroy enemies/traps or to create some
- Spikes
    - Summon spikes in an area

Enemies

- Enemies will have different behaviors with the aim to kill the player, protect an area, or flee. Their size and physical attributes will represent the behavior of the enemy, and will have different Types (color/texture differences) to mark power and abilities of the enemies
    - Large enemies will patrol an area. If they see a player they will pause and then charge at the player. They cannot be damaged from the front.
    - Medium enemies will guard specific locations or track the player through an area. They can be ranged or melee, and are the most common type.
    - Small enemies will try to hide from the player behind cover. They are ranged if they can attack, and will only do so if they aren't spotted behind cover. They often run away from the player and limits the abilities.
- The nightmare entity is a unique boss enemy who appears when your dream becomes a nightmare. He relentlessly pursues the player and is the most dangerous enemy. He always appears in the final room of a dream.

NPCs

- The Guide: an npc that the player will meet across different dreams to give the player an end goal/advice.
- Neutral NPC: can be affected by abilities but don't aim to harm the player/ have major interactions with the player.

Interactable Objects

- Most objects in the world can be upended or destroyed by abilities. The larger and heavier they are, the more powerful the player will have to be to move or destroy them. These objects combined with the player's abilities will be the key defeating enemies and therefore advancing the player towards the end of the level.
- Tethers that help ground the player within the dream, to prevent him from waking up.

Artistic Style

Lucid is a 3D game with a point of view that is a mix between top down and isometric. Environments and character models will be deceptively low poly. Cartoonish textures in combination with distortion shaders will be used to add definition to meshes. We aim to find a synthesis of HP Lovecraft horror with cartoonish overlays. The dream shader will be reminiscent of the visual overlay in the Lord of the Rings films when the one ring is used.

Characters take forms of creatures that only dreams and nightmares would create. They will be deformed, misshapen, or abstract. Furthermore, each character has a mask. Masks are the defining characteristics of models. Upon death, models will fade away while their mask remains behind to mark their defeat.

The environments will consist of a ground level and the assets generated on top. Procedural generation will handle the color of the ground and the location of assets. The ground will then be passed through a noise map in order to give a gradient and non-uniformity to its color. The theme of the game reflects dream-esque environments. Assets will avoid total realism. While structures may mimic real world environments, they will be misshapen or distorted. Global particle effects will be used with varying colors and settings to help the player mark where he has been.

Process

MVP: A non-procedural dream of 3 pre-made areas to fight 1-2 types of enemies using 2 abilities. Equilibrium will play a part in combat. The game will enter a lose-state if the player dies or the dream crumbles. An enemy and player model will be visible but will lack animations. The areas and assets will have an experimental shader attached.

Procedural generation: Lucid will spawn a dream consisting of a fixed amount of areas in a grid. The procedural generation will not only spawn these areas in a random fashion, but will have the ability to spawn a specified amount of NPC’s for every area.

The first milestone to be reached is generating the grid of areas with connectors that allow for at least one point of entry to the center area. Varying sizes of areas will be implemented next with more refinement to the connector system. Once this is complete, focus will move to procedurally adding assets to the world without overcrowding or hindering the ability to reach connectors. This process will also begin with simple shapes until such a time that environmental assets are available.

Expected challenges will stem from the ability to create a guaranteed path to the center of the dream. Ample time will be given to this aspect. If completion will take too long, an alternative, non-procedural system for connectors will be implemented.

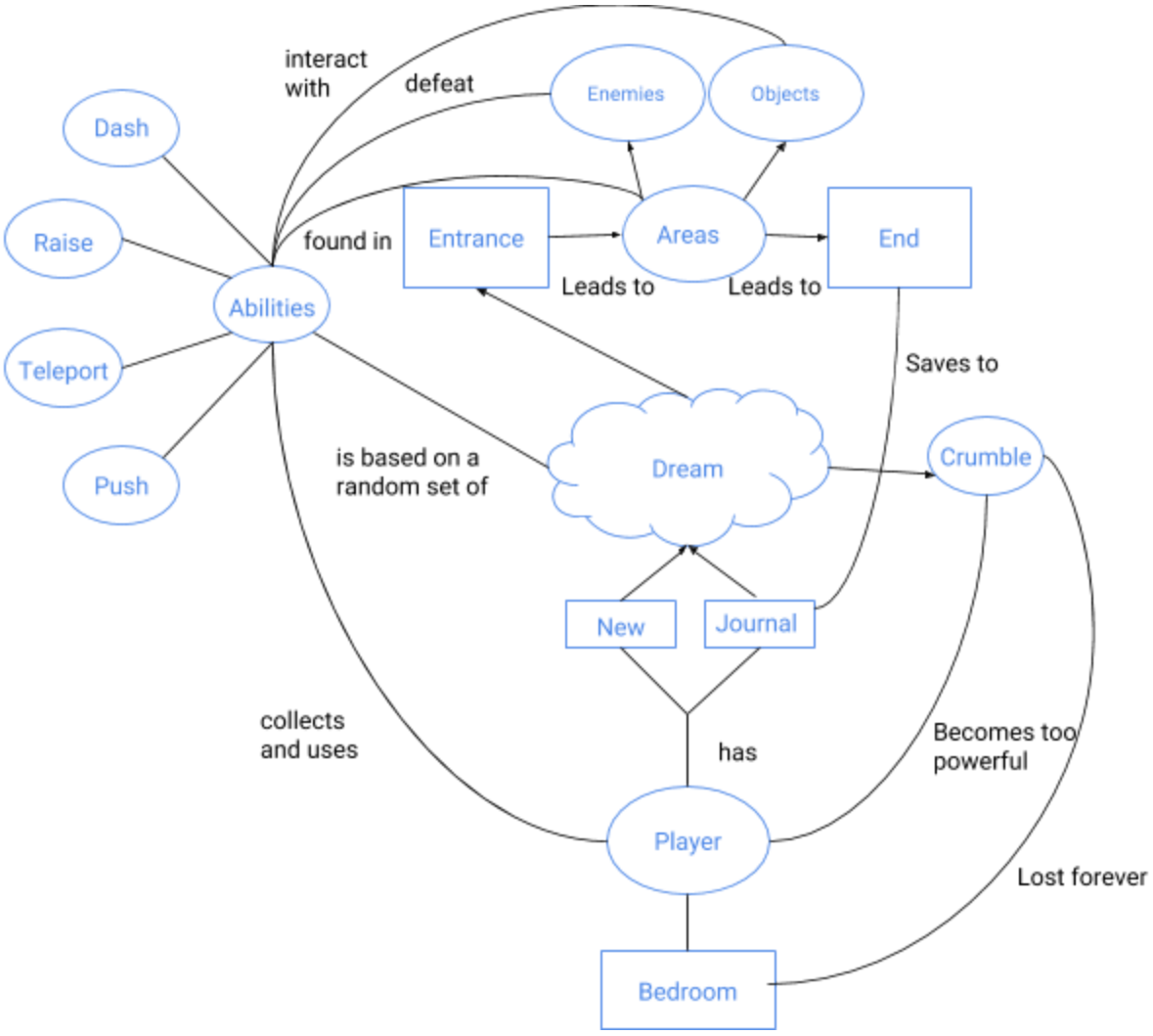
The character creation pipeline: Each of the characters in the game will be first modeled in Maya by the character artist. They will then be unwrapped and textured by the texture artist. Lastly they will be handed over to the animator to be rigged, skin weighted, and animated with with three standard animations: idle, walk, attack. The pipeline is initially expected to take 1.5 weeks to produce a standard enemy. This process is expected to be hastened after some successful runs. This will be done for the initial three enemies. Afterwards the pipeline focus will move towards the player for two weeks before returning to enemies for the remainder of the project.

The enemy AI and capabilities will be designed alongside this process by the AI programmer. After completion of the initial AI functionality, new elements for unique elements are also expected to take 1.5 weeks.

Expected challenges will stem from the efficiency of the three artists working on a character. To counter this, artist's waiting for new products will enhance prior characters or textures.

Shader development: The various shaders will be designed using the Unity plugin “Shader Forge” and with code when necessary. Shaders will be used to convey distortion during crumbling, the general type of enemy being faced, the death animation of characters, and the overall tone of environments.

Appendix


interact with
defeat
Enemies
Objects
Dash
Raise
found in
Entrance
Areas
End
Abilities
Leads to
Leads to
Teleport
Saves to
Push
is based on a random set of
Dream
Crumble
New
Journal
collects and uses
has
Becomes too powerful
Player
Lost forever
Bedroom