

Midnight Odyssey

Art and Making

Introduction

Midnight Odyssey is a 2D platformer puzzle game developed and designed by Yangying Ren. Player needs to control two toys and helps them get out of the house. This is a game about loneliness and help. What I want to present is that we can always find someone nearby who can support us. In this art and making guide I will present the design process of my works including character settings, UI, game scenes, game mechanism and animation storyboards. Also the development calender and some code samples or other game information can be found in the appendix.

Index

002 Art Design	
Character Setting.....	002
The shepherdess.....	002
The little prince.....	003
Three views.....	004
Unused settings.....	005
UI Design.....	006
Game Interface.....	006
Game Interface.....	007
Option and Chapter Menus.....	008
009 Game Design	
Flow Chart.....	009
Game Mechanism.....	010
Chapter1: Loft.....	010
Chapter2: Hall.....	012
014 Animation	
Storyboard.....	014
Opening Animation1.....	014
Opening Animation2.....	016
018 Appendix	
Development Calendar.....	018
Code Samples.....	019

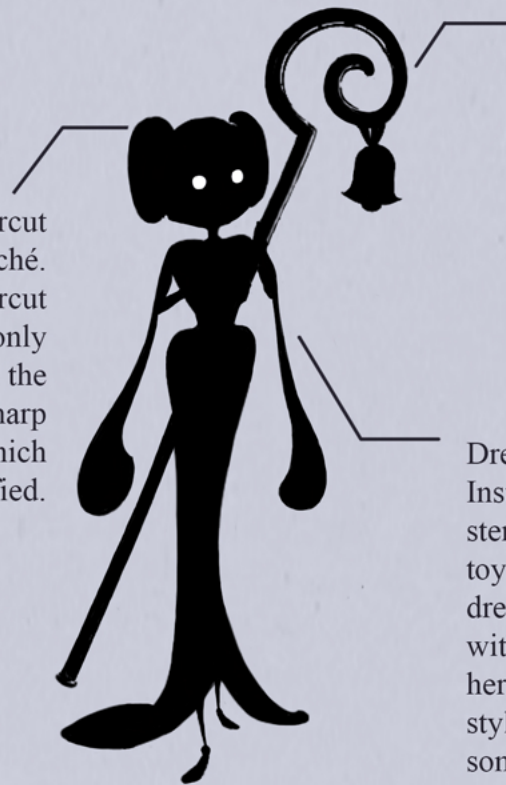
Character Setting

The shepherdess

When decided to make a game about toys adventure the first character that I thought is a classical and elegant shepherdess. Her body is made by wood and dresses in silk long dress. She is sensitive and pessimist, but also adamant.

Hair:

I want my character's haircut keeps classical but not cliché. Therefore Princess Leia haircut is a perfect choice. It is not only very elegant but also makes the shepherdess in to a sharp inverted triangle shape which makes her easier to be identified.



Cane:

The design of cane is referenced from the question mark and treble clef. When the shepherdess shakes it, the bell on it can make ringing sound. It also can help shepherdess to reach further.

Dress:

Instead of the traditional stereotype of shepherdess toy with large bubble style dress, I choose a long rope with kimono sleeves to make her more slim and tall. The style also is referenced from some medieval English style rope.

In game action:



Standing



Sleeping



Turning switch



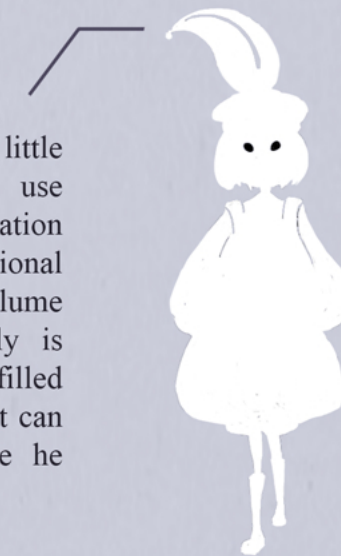
Character Setting

The little prince

The little prince is the inverse of the shepherdess. He is a fluorescent toy abandoned in the loft, and before the arriving of shepherdess most of time he was asleep. He is cute, innocent but also very childish, like a baby. Unlike the shepherdess who has a very strong ego, the little prince actually does not know the meaning of personality.

Hat:

This is the icon of little prince. I want to use more oriental decoration instead of traditional crown. The huge plume on the hat actually is made by silk and filled with cotton. The hat can keep shaking while he keeps moving.



Bloomers:

The bloomers are reference from Renaissance male clothes. Create a soft and pear shape figure, which is also the opposite shape of the shepherdess. The bloomers are made by silk fabric and fullfilled with cotton.

In game action:



Standing



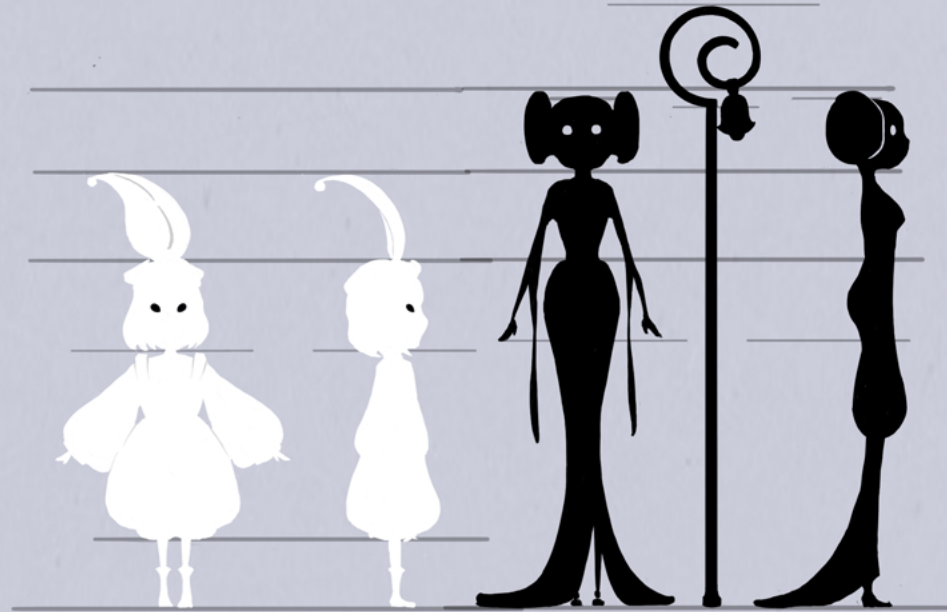
Sleeping



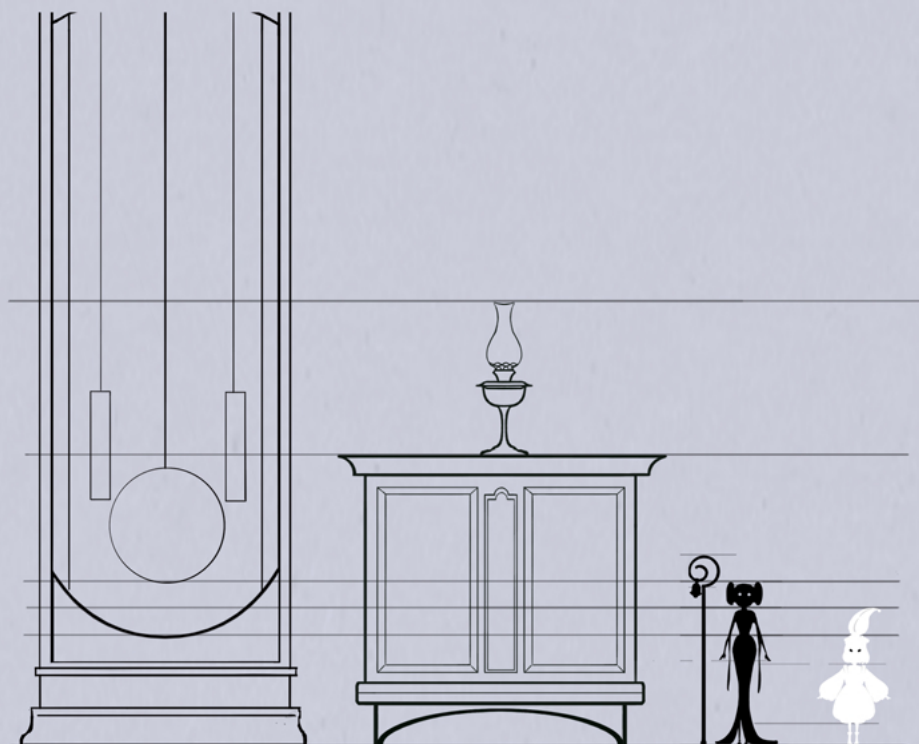
Turning switch



Character Setting

Three-view drawing

Compare with in game furnitures:



Character Setting

Unused Settings

At first I designed some more traditional figures with large bubble dress and early 20th century haircut. My ideas was from several movies such as *Toy story*, *The sound of music* and many designs from *Gone with wind*. But the finally design actually is referenced from *Star War*, a more mature and stylish figure.



The design of second character should have a distinct comparison with the first character thus at beginning the figure of the little prince is more close to a gentleman, like Jack Skellington from *Nightmare before Christmas*, tall and thin. But since the final design of the shepherdess is already slim, I want the little prince to become more lovely, giving player a feeling of soft and innocent.

UI Design

Start Interface

The start interface is what player will see first after the opening animation. The design of it is referenced from old style candy advertisements and cartoon posters (such as *Tom and Jerry*, *Looney Times*) and Christmas cards.



I choose purple as theme color to create an atmosphere of fantasy and a little bit gothic, giving player a feeling of mystery and fairy tale. The sun and moon at two side and the sunflower and epiphyllum at the bottom of interface are the metaphor of two characters. One should in the light and the other one should in the dark. Also to fit the style of interface the font has been chosen as old English style.






When player clicks the title it will change to the main menu. Player can choose whether start a new game or select specific chapter to play. When player moves the cursor to selected button there would be a frame around it to emphasize where is selected.

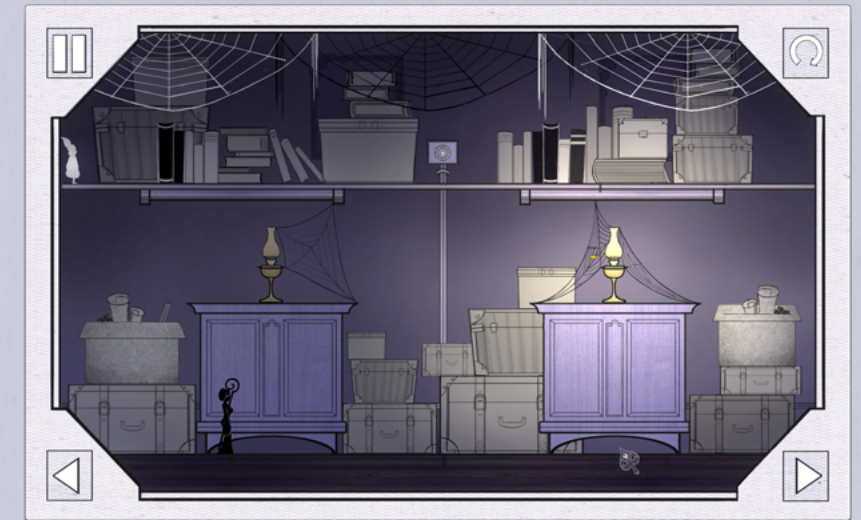


UI Design

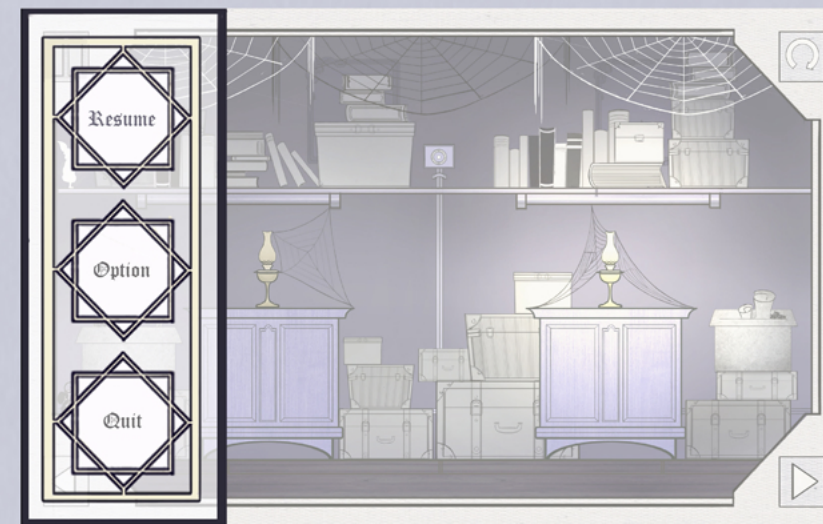
Game Interface

The design of game interface is referenced from old mirror frame and radio button. I want player can have the feeling of prying while playing, like watching the old style television with very thick border. It is also referenced from some 19th century book's layout such as dictionary and encyclopaedia.

-  Pause: By clicking it player can pause the game and the pause menu will show up.
-  Restart: By clicking it player can restart the level.
-  Check: Player can click it to preview the rest of map if available.
-  Return: By clicking it camera will back to previous position.
-  Cursor: Mouse cursor will turn to black spade while playing.

Pause Menu

The pause menu's design is referenced from Middle East stained glass windows. At here player can go to option interface by clicking "option" or back to start interface by clicking "quit".

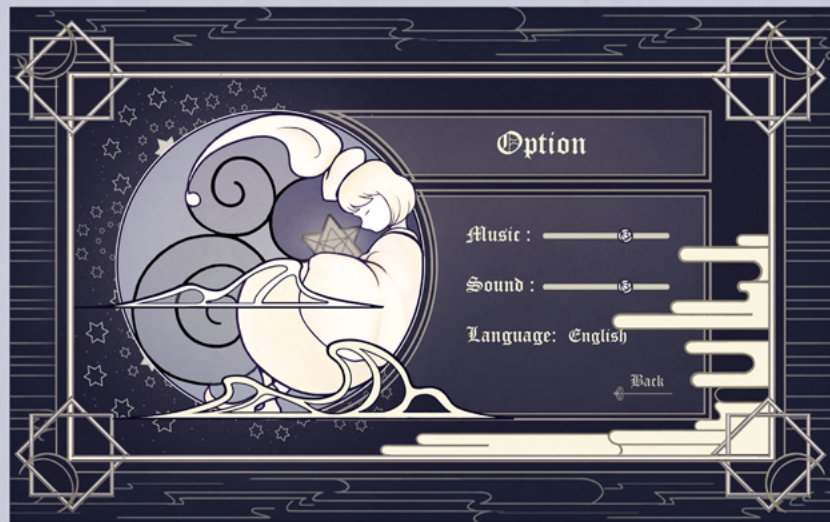


UI Design

Here are the option and selection menus which player can adjust game settings or chooses different chapter to play. The art style of menus is referenced from illustrations and advertisements of Alphonse Mucha.

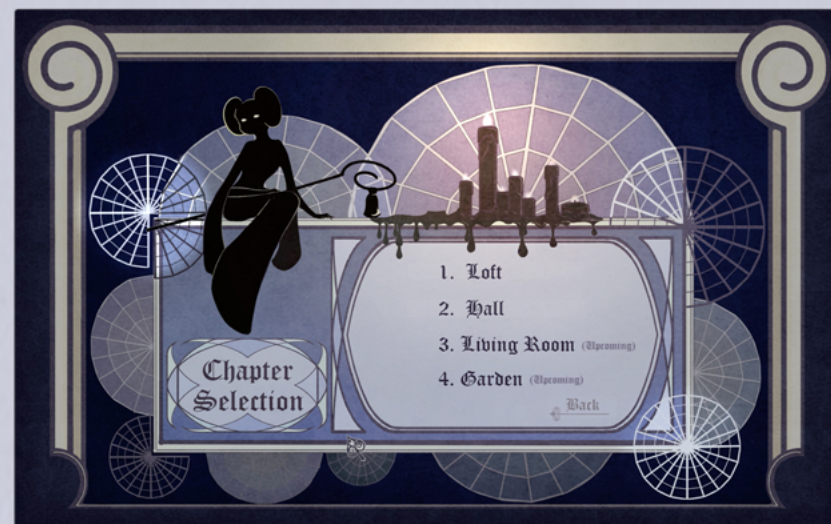
Option Menu

The Option menu can be found on either start interface or every level's resume page. To fit the theme of midnight I have chosen star and moon as main decorations and curly lines around as mist and cloud. Also the button on the scroll bar has been designed into rose shape to match the romantic and gothic style.

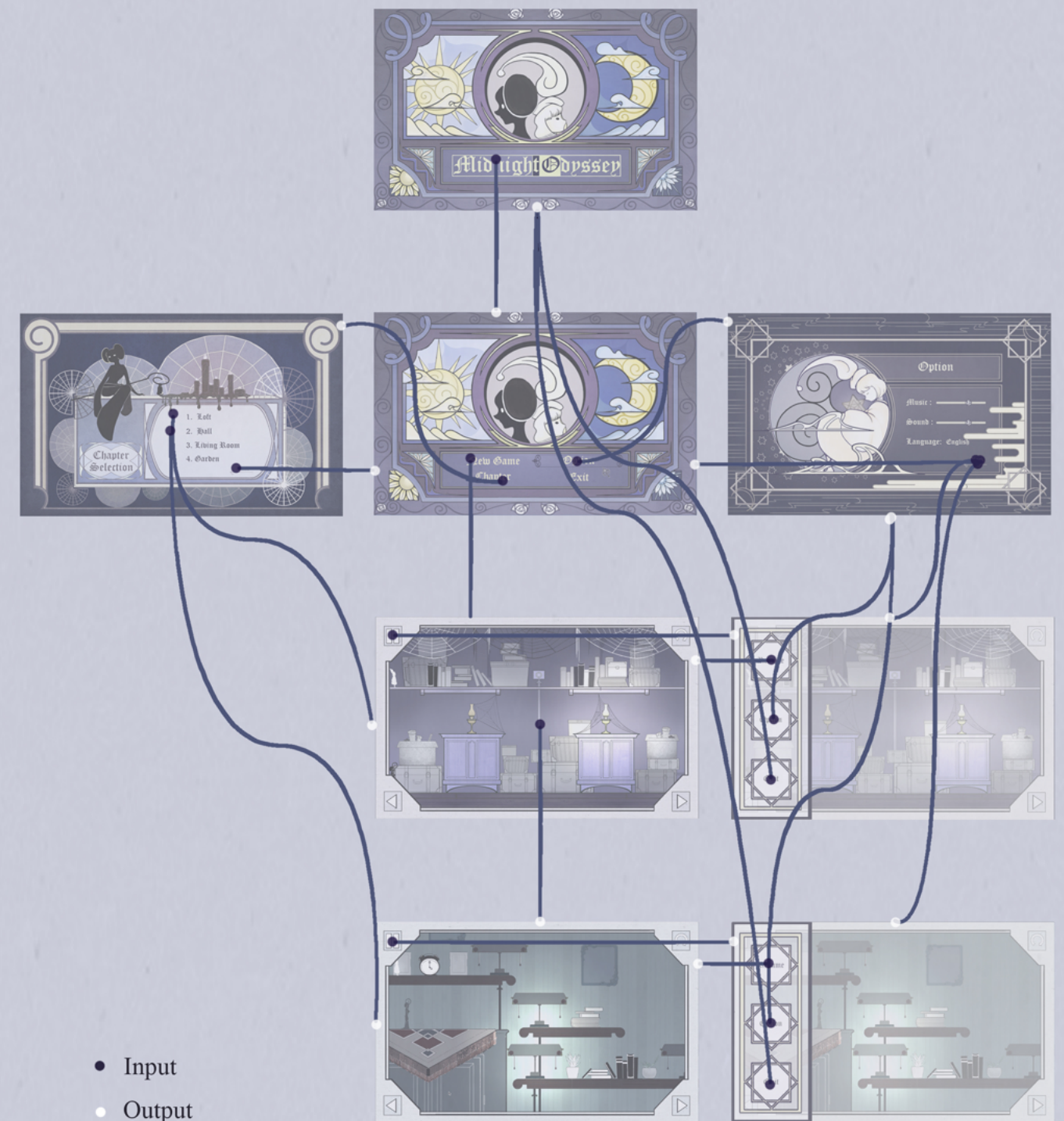


Chapter Selection Menu

The chapter selection menu can be found on start interface. Player can choose any chapter they want to play here. The rounds and candles with sparkled light around are metaphor of spider web and lamps, also create a dreamy and quiet atmosphere.



Flow Chart



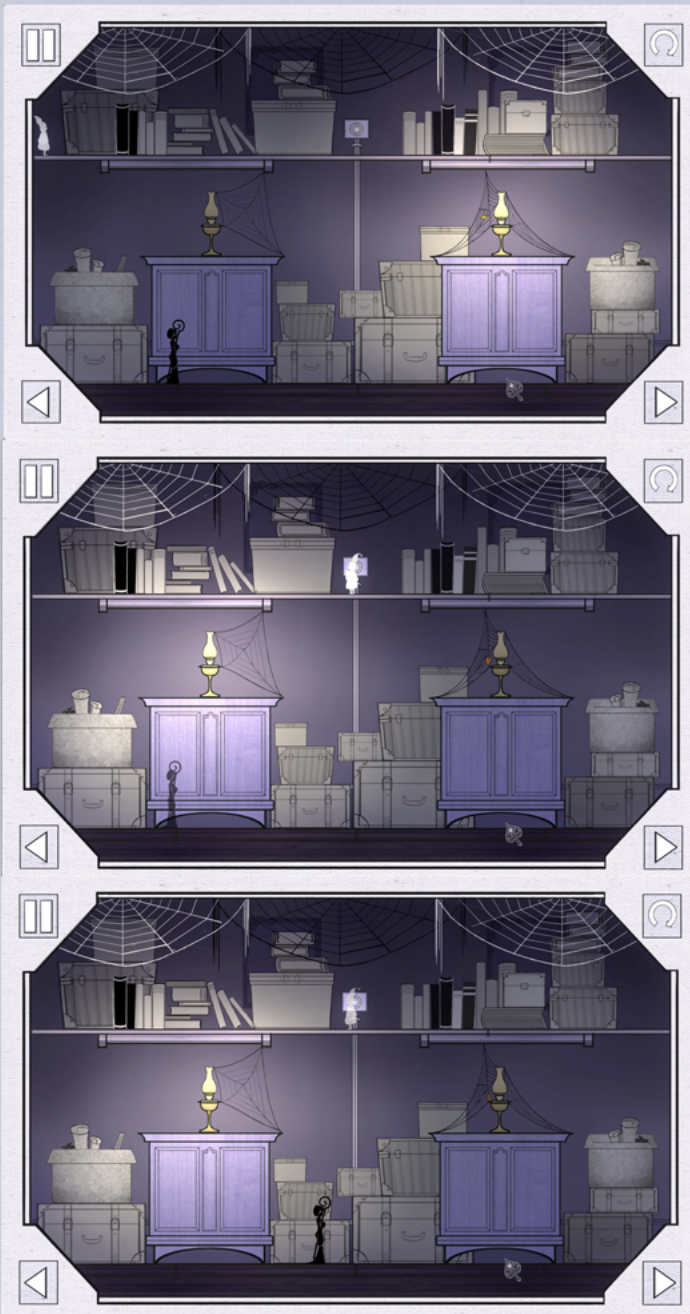
- Input
- Output

Game Mechanism

The basic mechanism of *Midnight Odyssey* is finding a solution to let both characters move from the left side to the right side. In different maps there are different type of lights and switches so player needs to try them and figure out the right route.

Chapter 1: Loft

In this chapter switches and lamps are distributed separately in the map and the relationship of switches and lamps are not one to one correspondence. Here is an example of how to play the game.



I
When shepherdess approaches the lamp but lamp's condition is turning off, she cannot pass it and the screen will keep shaking. On the contrary if the lamp was turning on, the little prince cannot pass through.

II
Change to another character and find the right switch to change the light condition. Please pay attention that this action may change the condition of other lamps too.

III
Change again the character and pass through.

Game Mechanism

Items and Maps

Kerosene Lamp:
Enable to light very small range.
Enable to affect the action of only one character.



Lamp:
Enable to light large range.
Enable to affect the action of all characters.



Switch:
Control the light of lamps.

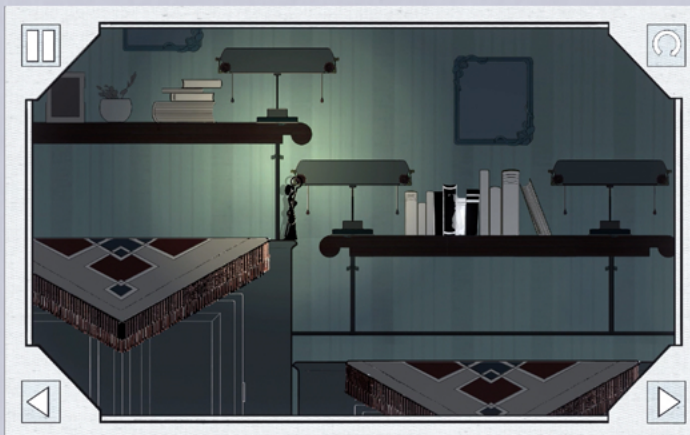
Scene Examples:



Game Mechanism

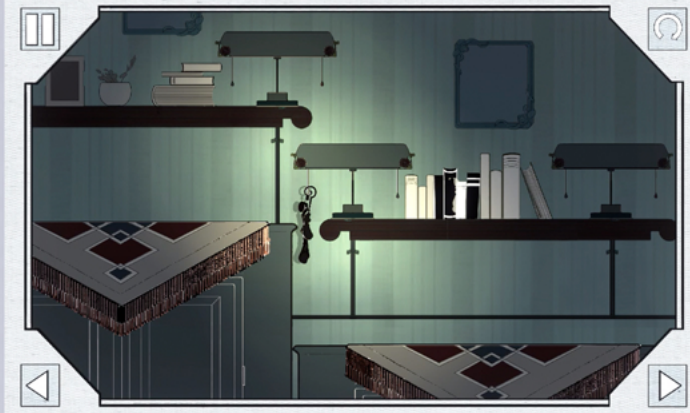
Chapter 2: Hall

In this chapter one lamp has two switches on each side of it. Both of them can control the lamp. Player need to transport two characters from the left top to right bottom. Here is an example of how to play.



I

The range that the lamp can affect is double deck. As shown in the picture the lamp can also affect the platform under it.



II

Characters need to get the switch on another platform's border in case of reaching lower platform. At the same time when character pulls this switch, the lamp's condition will change.



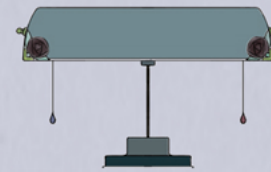
III

Arriving at the lower platform and pass through. Please pay attention that characters cannot return to the last platform again.



Game Mechanism

Items and Maps



Desk Lamp:
Only lamp type in chapter 2. Its light not only affects its own position but also affects the below platform.

Blue Switch: Can only change the light.

Red Switch: Can change the light and also play a function of dragline for characters to reach below platform.

Scene Examples:



Storyboard

Scene	Action	Music	Time
Opening Animation 1			
	Count down from 5.	Clock tik tok sound	00'
	First person view point. Open the door.	Glockenspiel followed with piano arpeggio.	04'
	Show up the room.	Another high key piano arpeggio joins in	07'
	Hold the storage box. Showing up from the bottom of screen		09'
	Enter room, move closer to bookshelf.		09'

Scene	Action	Music	Time
	Move in front of the bookshelf.		10'
	Open the bookself.		10'
	Take out toys. Put in the storage box.	One high key chord after every 4 repeated low key arpeggio	11'
			12'
			12'
			13'
			13'

Storyboard

Scene	Action	Music	Time
	Change to left hand.		14'
	Take out the shepherdess.		15'
	Staring her closely for a while.	Key turns down when hand put in bookshelf.	16'
	Put in storage box. Camera follows the shepherdess.		19'
	Camera focuses on box inside. Put other items in the box.		20'
			22'
			24'
	Push camera closer to shepherdess' face	Key turns up again when the camera focuses on shepherdess eyes.	26'
	Focusing on her eyes		28'
	Change the viewpoint to shepherdess. Show the reflection of the box holder		29'
	Rotatet the camera while close the box.		30'
	Screen turns all black when box closed.		31'
	Camera turn to third person view. Shepherdess blinks in the dark.	End with a continuous arpeggio.	34'
	End		37'

Storyboard

Scene	Action	Sound	Time	Scene	Action	Sound	Time
Opening Animation2							
	Box shaking.	Shaking	00"		Squat down. Start searching something.		12"
	Box falls down, some objects get out of box.		06"		Find her cane. Get it out box.		13"
	Shepherdess climb out.	Walking	07"		Check whether there are any other toys alive.		15"
	Stand up		09"		No response. deep breathing again.	Shaking again	20"
	Deep breathing		09"		Stand up and turn back again.		23"
	Turn around, facing the box		11"				

Storyboard

Scene	Action	Sound	Time	Scene	Action	Sound	Time
			23"				40"
	Leave, be out of screen.	Walking	25"		Wake up.		42"
	Screen becomes darker.		27"		Leave, be out of screen		45"
			28"				47"
	Panning out, until the little prince showing in the screen.		36"				49"
			39"				51"

Appendix

Here is other information about the game including development calendar, development tools, code samples and credits.

Development Calendar:

Time	Work	
07/27/2019-08/01/2019	Art Design	Character Design
08/01/2019-08/15/2019		Opening Animation 1
08/16/2019-08/20/2019		Character Poster
08/21/2019-09/03/2019		Opening Animation 2
09/04/2019-09/15/2019		UI
09/16/2019-09/20/2019	Level Design	Level Design
09/20/2019-10/02/2019	Art Design	Scene Design
10/03/2019-11/15/2019	Programming, Test	

Development Tools: Adobe Photoshop, Unity 3D, Adobe After Effects

Credits:

Direction: Yangying Ren
 Art Design: Yangying Ren
 Level Design: Yangying Ren
 Programming: Yangying Ren
 Animation: Yangying Ren
 Special Thanks: Peng Huang
 Xuan Qi

Cover Design and Layout: Yangying Ren

Download Link:

<https://www.yangying-ren-gameandart.com/midnight-odyssey>

<https://yangying-ren.itch.io/midnight-odyssey>

Code Samples

```

Users > renyangying > Downloads > MidnightOdyssey > Assets > switch_control.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class switch_control:MonoBehaviour{
7
8     public GameObject light_switch;
9     public Animator s_animator;
10    public bool IsIn;
11
12    void Update(){
13
14        if(Input.GetKey(KeyCode.W)&&GameObject.Find("f_moving").GetComponent<character_switch>().enabled==true){
15            s_animator.SetInteger("switch",1);
16        }
17        else{
18            s_animator.SetInteger("switch",0);
19        }
20
21        if(GameObject.Find("f_moving").GetComponent<character_switch>().enabled==true){
22            if(Input.GetKeyDown(KeyCode.W)){
23                GameObject.Find("f_moving").GetComponent<AudioSource>().Play();
24            }
25        }
26
27        if(IsIn==true&&GameObject.Find("f_moving").GetComponent<character_switch>().enabled==true){
28
29            if(Input.GetKeyDown(KeyCode.W)){
30                gameObject.GetComponent<AudioSource>().Play();
31                light_switch.SetActive(false);
32                gameObject.GetComponent<switch_control>().enabled=false;
33                gameObject.GetComponent<switch_reset>().enabled=true;
34            }
35        }
36    }
37

```

```

Users > renyangying > Downloads > MidnightOdyssey > Assets > trigger.cs
6 public class trigger:MonoBehaviour{
7
8     public Transform playerTransform1;
9     public Transform playerTransform2;
10    public bool IsIn;
11
12    public Camera s_camera;
13    private Rect changeRect;
14    private float shakelevel = 1.5f;
15    private float shakeDelta = 0.002f;
16
17
18    void Awake(){
19        changeRect=new Rect(0.0f,0.0f,1.0f,1.0f);
20    }
21
22    void Update(){
23
24    }
25
26    private void OnTriggerStay2D(Collider2D other){
27        IsIn=true;
28        if(IsIn==true){
29            if(other.tag=="Player2"){
30                if(gameObject.GetComponent<light>().enabled==true){
31
32                    changeRect.xMin = shakeDelta * (-1.0f + shakelevel * Random.value);
33                    changeRect.yMin = shakeDelta * (-1.0f + shakelevel * Random.value);
34                    s_camera.rect = changeRect;
35
36                    if(playerTransform2.localScale==new Vector3(1,1,1)){
37                        playerTransform2.position=new Vector3(playerTransform2.position.x-1f,playerTransform2.position.y,playerTransform2.position.z);
38                    }
39                    else if(playerTransform2.localScale==new Vector3(-1,1,1)){
40                        playerTransform2.position=new Vector3(playerTransform2.position.x+1f,playerTransform2.position.y,playerTransform2.position.z);
41                    }
42                }
43            }
44        }
45    }
46

```