

LAND OF DESIRE

Introducción.

Land of desire es un simulador de sexo similar a otros títulos del género como 3D Sex Villa o Honey Select. Pero con la particularidad de ser gratuito y de código abierto y basado en un motor que comparte dichas características. Con la esperanza de que este juego sea fácil de modificar y con suerte, sobrevivir al paso del tiempo si se forma una comunidad alrededor de este. Como así también, ser una base para aquellos que quieran crear sus propios juegos del mismo género.

Importante

Si vas a modificar el juego, crear ropas, animaciones o lo que sea, te recomiendo que uses la misma versión del motor para evitar problemas de compatibilidad que este pueda llegar a tener.

Tecnologías utilizadas.

- **Motor:** UPBGE 0.36.1
- **Modelado:** El mismo motor del juego es un Blender modificado.
- **Iconos.**

Los iconos del juego fueron creados con el programa LibreOffice Draw de la suite LibreOffice versión 24.2.7.2

Estructura de carpetas

```
Land of desire/  
├── Characters/  
│   ├── Female/  
│   └── Male/  
│       ├── Clothes/  
│       │   ├── Pants/  
│       │   ├── Shoes/  
│       │   ├── Textures/  
│       │   └── Torso/  
│       ├── Exported/  
│       ├── Eyebrows/  
│       ├── Eyewear/  
│       ├── Hairs/  
│       ├── Piercings/  
│       └── Skin_Tectures/  
└── Data/
```

- Icons/
- Python/
- Scenes/
- icons.odg
- all_game.blend

Programación

El juego tiene scripts escritos en Python y hay partes que están hechas con el sistema 'Logic Bricks' y 'Logic Nodes' del UPBGE trabajando en conjunto.

Sistema de edición de personajes

Al seleccionar el tipo de personaje que se quiere crear (hombre o mujer) Se carga el correspondiente .blend 'Female_char_Editor.blend' o 'Male_char_Editor.blend' de la carpeta 'Characters' Estos archivos al arrancar, cargan el contenido de 'Female_char_template.blend' o 'Male_char_template.blend' según corresponda. Recorriendo los shape keys de cada malla y el contenido de las carpetas de ropa para crear en tiempo real la interfaz visible en los editores.

Modificación corporal (Shape Keys)

El personaje base tiene múltiples shape keys para modificar la forma del cuerpo (grosor, estatura, etc.). Cada barra de desplazamiento en el menú ajusta el valor de una shape key específica mediante el script de Python 'shape_key_value_change.py'.

El nombre del shape key que se modificará está definido en el objeto Empty 'script_holder' dentro de la propiedad ShapeKeyToChange.

Color de piel

El color de piel se modifica cambiando el tono, la saturación y el brillo de la textura de un material específico. Estos tonos se modifican con el script 'change_skin_color.py'

Carga de ropa

Cada prenda es un archivo .blend separado con una malla que se ajusta al personaje.

Sistema de carga:

- Al seleccionar una prenda, se carga el archivo y se instancia el objeto.
- Se posiciona automáticamente en el cuerpo.

- Las prendas deben tener las mismas shape key que el cuerpo del personaje ya ajustadas al cuerpo del mismo, cuando se importa la malla de la ropa automáticamente se le asignan el valor que las shape key del cuerpo tengan. El script que se encarga de esto es: 'load_clothing_or_hair_or_eyebrows.py' el mismo script se usa para cargar el pelo.
-

Guardado de personaje

El sistema usa el script 'save_character.py' para guardar en tiempo real:

- Las shape keys modificadas.
- El color de piel.
- Las prendas y cabellos seleccionados.

Esto permite retener los cambios entre sesiones si se integra con un sistema de guardado posterior.

Todo se guarda como un .json dentro de las carpetas 'Exported' dentro de 'Female' o 'Male' según lo que estés trabajando.

Scripts principales del editor del personaje

- shape_key_value_change.py: Modifica el valor de una shape key específica.
- load_clothing_or_hair_or_eyebrows.py: Carga y ajusta una prenda.
- change_skin_color.py: Cambia la textura del material del cuerpo.
- save_character.py : Guarda al personaje como un archivo .json.

Los scripts se llaman mediante Logic bricks por botones y la carga de la escena (esto último en el objeto 'scrip_holder').

Ropa

La ropa de ambos hombres y mujeres se guardan en sus respectivas carpetas. 'Torso' para ropa que cubre el torso. 'Pants' es la misma para prendas que cubran piernas o ingle siendo pantalones o ropa interior (solo puede vestir una a la vez. Un personaje no puede tener pantalones y calzoncillos. Por ejemplo) Lo mismo para los zapatos. La carpeta 'Shoes' es para guardar zapatillas, medias o pantuflas.

Extensibilidad

Agregar nuevas shape keys

Crear una nueva shape key en el personaje base 'Male_char_template.blend' o 'Female_char_template' según corresponda.

Agregar nuevas prendas

1. Modelar la prenda ajustada al personaje base.
2. Esa prenda guardarla en la carpeta correspondiente como .blend (archivo de blender/UPBGE)
3. Asegurarse que la prenda tenga las mismas shape keys que el personaje base, cada una ya modificada y acomodada a la malla en los distintos estados de la shape key.

(Recomiendo simplemente copiar el personaje base al que se la va a poner la prenda y modificar esa malla.)

4. Las mallas de las prendas deben llamarse 'torso_up' si es prenda del torso o pechos, 'pants' si es pantalones, calzoncillos o tanga y 'shoes' si es zapatos, pantuflas, chanclas, etc.
5. Las mallas de las prendas deben estar rigueadas a una copia de los Armatures del personaje al que corresponda, como así también, guardados en las carpetas correctas de su respectivo personaje. Eso es importante para que el juego pueda ubicar los objetos tanto en los archivos del juego como los objetos dentro de cada .blend.

Notas técnicas

- La ropa no usa simulación de colisiones. Se crea con un margen extra para evitar intersección.
- El sistema detecta si una prenda ya está cargada y la elimina de la escena para evitar duplicados.
- El HUD está hecho completamente con objetos 3D planos.

Animaciones

Si quieres crear nuevas animaciones lo único que tienes que hacer es abrir el archivo 'animations.blend' dentro de 'Characters' Ahí encontrarás unos emptys planos y cúbicos dispuestos a modo de marcadores y una pareja (hombre y mujer) con los mismos Armatures que los de los templates. Simplemente anima esos personajes usando el marcador que quieras como referencia (si uno está sentado, apoyado, etc.) A ambos personajes ponles el mismo nombre de animación cuando apliques el NLA. No te preocupes si a uno el blender el pone .001

.002 o etc al final ya que el juego después tiene como trabajar con ese tema.

El nombre de las animaciones siempre debe tener la nomenclatura 'pose': 'area' siendo 'area' el nombre exacto del marcador donde se ejecuta esta pose. 'bed/floor' para que la pose sea en la cama o en el piso. 'table' para que sea sobre una mesa. 'wall' para la pared y así.

Otro tema, Jamás modifiques un marcador, ni su forma ni su tamaño ni su nombre, estos siempre deberán tener las mismas dimensiones en todo los archivos del juego donde estén presentes.

Escenarios

Para crear un escenario importa los marcadores (los empty) del .blend 'animations.blend' (ignora a la pareja.) Estos emptys los tienes que usar como referencia de tamaños para cada cosa de la escena. Jamás modifiques el tamaño ni la forma de un marcador. Sí puedes rotarlos y ubicarlos como y donde se te dé la gana. Pero si por ejemplo modelaste una cama y esta resulta o muy grande o muy chica comparado con el marcador 'bed/floor' pues modifica la malla de la cama, NO el marcador.

Una vez que creaste la zona. Guárdala como un archivo .blend dentro de la carpeta 'Scenes' junto a una captura de la misma en .png, esa imagen posteriormente se mostrará (aún no implementado en esta versión) en el menú de selección de escenas. Eso es todo, el botón para elegir tu escena se crearán automáticamente cuando se cargue ese menú al usar el programa.

Juego (Archivo all_game.blend)

El launcher lo único que hace es cargar en tiempo real el blend 'all_game.blend' el cual es importante que ya se encuentre en la escena del menú principal y con la perspectiva desde la cámara. (Recuerda esto si te metes para modificar algo)

Este archivo tiene:

1. El menú principal.
2. El menú de selección de personajes y escenarios
3. La escena de juego.

Configuración.

La configuración se aplica y guarda con un logic nodes 'VideoSettings' asociado al objeto 'ScriptHolder' del menú principal. Y se aplican con el logic nodes 'AppySettings' Asociado al botón 'ApplySettingsButton' de la misma escena, que a su vez guarda esa información en un archivo .json 'variables.json' en la carpeta 'Data'.

El botón de iniciar partida solo cambia de escena a la de selección de personajes y escenas y aplica el HUD de la misma, una 'overlay scene'

Selección de personajes y escenarios.

Esto se lleva a cabo en la escena 'character_selector' (en un principio se iba a usar solo para los personajes, pero al final quedó para ambos).

En esta escena hay 2 marcadores. Uno para hombre y otro para la mujer, la idea era usarlos para ver los 'preview' de los personajes que se seleccionan. Los marcadores quedaron por error, ya que no se usan. Los botones para seleccionar a los personajes se generan en el script 'create_scene_buttons.py'.

Cuando ya se han seleccionado los personajes masculino y femenino junto a una escena es cuando se habilita el botón de iniciar partida. Que lo que hace es cambiar de escena a 'game_scene' y aplicar el HUD de la misma.

No todos los aspectos del juego están cubiertos aquí. Y pido disculpas por ello. Llevo 7 meses trabajando en este juego yo solo con ayuda de chatGPT. Pero espero que sea suficiente punto de partida para aquellos interesados en modificar mi juego o empezar a construir a partir de aquí.