3D Character Creator

Game overview & core features

A Nintendo 3DS-style 3D dress-up game where players can fully customize a stylized anime girl character. All 3D models and textures were created by the developer. Players can modify appearance, save/load characters, export screenshots, and randomize outfits.

- Color changing
    - Choosing color of hair and eyes on a color picker
    - Skin color from a gradient
- Makeup
    - Eye style
    - Blush
    - Lips
- Hairstyles, tops, bottoms, shoes and backgrounds chosen from premade options
- Gameplay utilities
    - Randomize a full character
    - Save and load
    - Export a screenshot
- Character blinks for a more alive look
- Rotating the character left and right
- Sounds played when pressing buttons

Technical design

1. Texture Changer

- Handles **runtime application of colors and textures**:
    - Hair, skin, eyes, lips, blush
    - Uses gradient-based skin system and color pickers
- Tinting is applied via texture manipulation at runtime
- Works closely with Face Texture Combiner to rebuild combined face textures

2. Face Texture Combiner

- Combines facial features (eyes, lips, blush) into a single texture for performance
- Manages blinking animation and eye styles
- Ensures facial textures are updated consistently when features or colors change

3. Mesh Manager

- Handles swapping 3D meshes (hairstyles, clothing, shoes, backgrounds)
- Tracks currently active mesh per category
- Provides methods to access the current renderer and textures

4. Color Picker

- UI managers for selecting hair, eye, and skin colors
- Sends color updates to Texture Changer and Face Texture Combiner

5. Save/Load System

- Serializes the current character state to JSON
- Stores mesh indices, colors, and facial feature selections
- Allows full restoration of character presets on reload

6. Randomizer

- Generates random characters by picking random options for hair, clothes, facial features, and colors
- Works through the same managers to apply selections consistently

7. Screenshot System

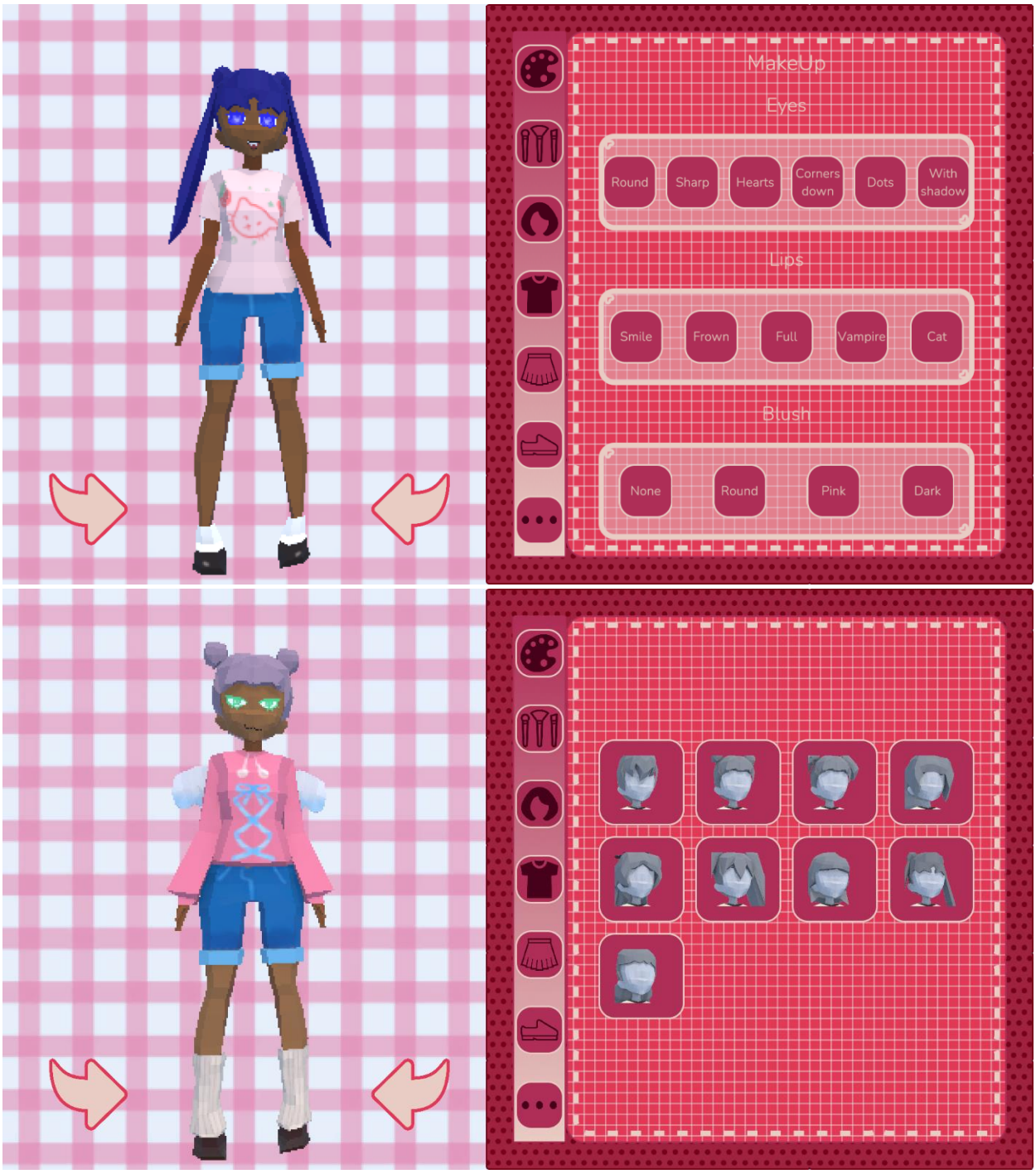
- Captures only the character area excluding UI
- Uses Read Pixels from the main camera viewport
- Matches in-game resolution and preserves pixelated art style

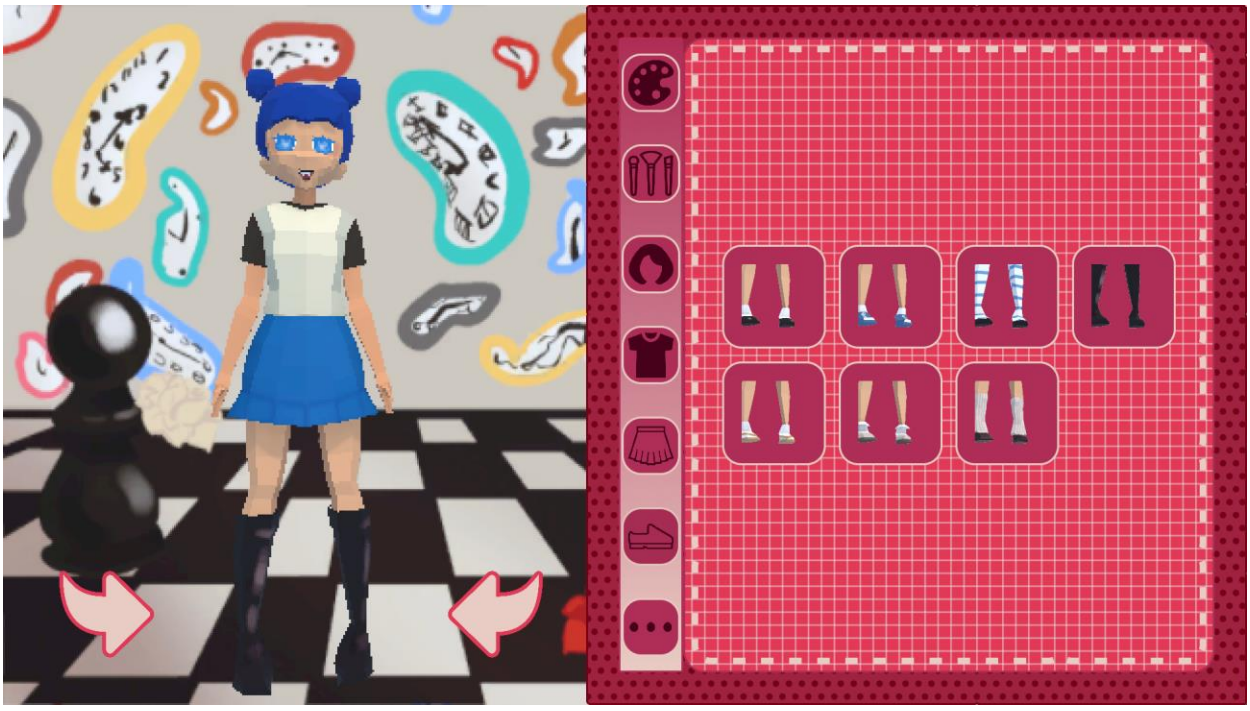
Asset creation & workflow

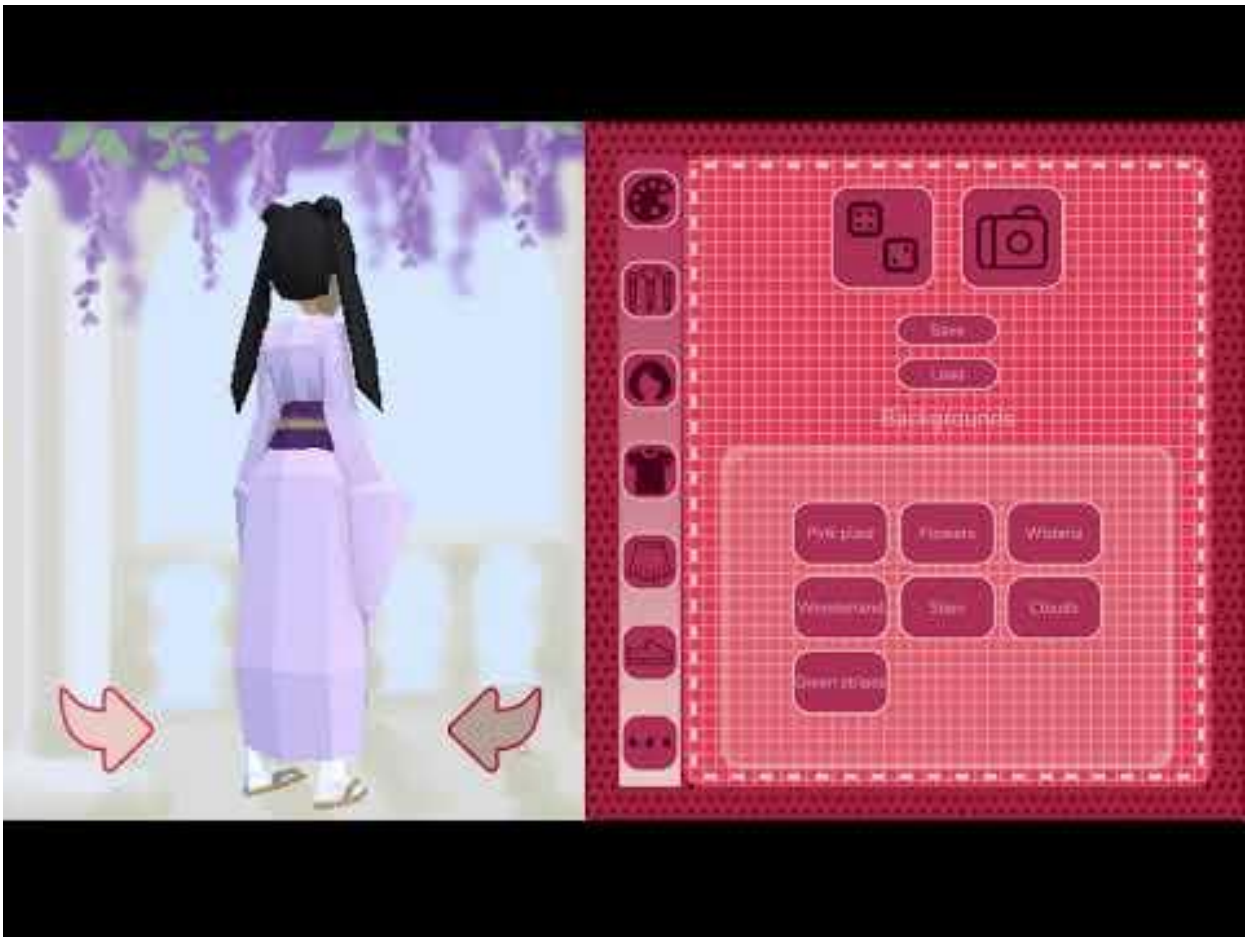
1. Designed the core character concept inspired by Nintendo 3DS-style visuals.
2. Modeled a fully customizable base character in Blender, optimized for Unity.
3. Created UV-layouts suited for modular texture blending and color tinting.
4. Designed hairstyles, eye styles, lip textures, and blush variations.
5. Painted textures in Krita, using layered workflows to support recoloring.
6. Modeled clothing pieces and accessories, ensuring consistent proportions.
7. Prepared assets with clean topology and efficient polygon counts.
8. Planned the UI layout (color pickers, style selectors, save/randomize buttons).
9. Created UI sprites and icons.
10. Imported UI into Unity and configured it to scale responsively without stretching (using sliced sprites / 9-slice where needed).

11. Built managers for hair, eyes, skin, clothes, and backgrounds.
12. Implemented a texture-combining system for makeup and skin details.
13. Added color-changing logic.
14. Implemented randomization logic for all customizable options.
15. Set up lighting and camera angles for consistent presentation.
16. Implemented character rotation.
17. Added blinking animation via texture swapping.
18. Added screenshot capture.
19. Implemented save/load of character customization via a serialized state.
20. Added button sounds
21. Created Git setup for version control and project tracking.

Screenshots


MakeUp
Eyes
Round
Sharp
Hearts
Corners down
Dots
With shadow
Lips
Smile
Frown
Full
Vampire
Cat
Blush
None
Round
Pink
Dark





Challenges & solutions

Challenge	**Solution**
Textures flickering	Adjusting the meshes so that no faces are in the same place
Parts of models disappearing	Switching materials from Backface Culling to Double-Sided Rendering
UI stretching	Changing sprite mode, enabling 9-slicing, defining borders, applying tiled mode where needed
Colors of eyes and hair being too dark	Reworked blending mode from just multiplying to a custom mix of overlay and multiply
Preview didn’t accurately display the colors	Changing target RenderTexture format to a non- HDR one