CS4620 | CS5620
Fall 2024

Creative #1 Report

ABU'S HAPPY CLEANING DAY

JOANNA YOO (SY797) & LIAN LIAO (LL987)

Table of Contents

Project Overview

Our team developed a game titled "**Abu's Cleaning Day**," where the main character, Abu, uses a vacuum to clean up trash randomly scattered across the map. The player navigates Abu around the environment, collecting trash using the vacuum system. The primary objective is to collect all trash items on the map to clean the entire area and achieve victory. To implement this, we utilized collision detection and a particle system technique, building upon concepts covered in our class lectures and assignments.

Design & Development Process

Sprint Planning → Defined the game concept, core mechanics, and scope

Sprint 1 *Prototype Development* → Built basic player movement, trash items, and collision detection.

Sprint 2 *Feedback & Iteration* → Collected feedback, improved collision accuracy, and refined vacuum mechanics.

Sprint 3 *Particle System & Optimization* → Added particle effects for the vacuum and optimized performance.

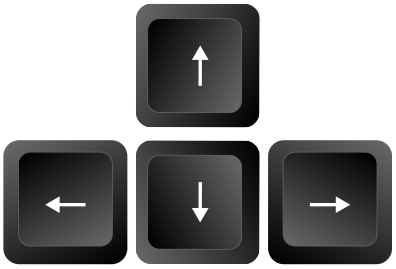
Sprint 4 *Testing & Final Refinements* → Adjusted win conditions, improved UI, and polished the game.

Our team followed the **Agile methodology** for the game design and development process. In each sprint, we held regular meetings to incorporate feedback and ensure that each phase focused on incremental improvements. Continuous refinement was a key aspect of our approach, allowing us to build and enhance the game step by step.

Core Gameplay Mechanics

(Note: All key inputs are case-insensitive.)

Keyboard Components



Direction Keys- Move Abu around the environment

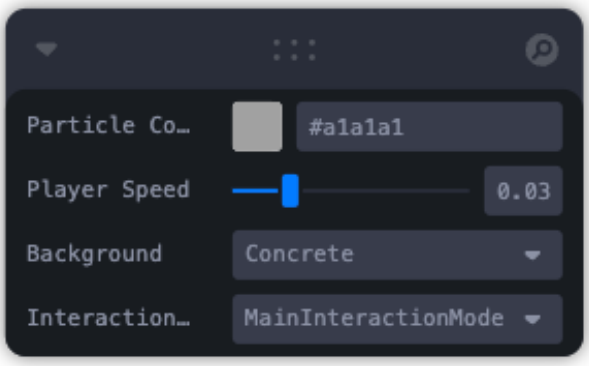
V- Activates the vacuum while held, allowing Abu to continuously collect nearby trash.

C- Changes Abu's Outfit

M- Toggles background music on/off

E- Empties the vacuum when near a trashcan

R- Resets the game



Interaction Slider Component

- **Particle Color**: Allows you to change the color of the vacuum particles.
- **Player Speed**: Adjusts Abu's movement speed.
- **Background**: Lets you change the game's background.

Technical Features

Collision Detection

- Collision detection ensures trash is only collected when Abu is within the vacuum's range.
- An invisible rectangle, attached to the vacuum head, simulates the vacuum range and detects collisions with trash for accurate collection.
- When near a trashcan, pressing the "E" key empties the vacuum, requiring accurate collision detection to ensure Abu is within proximity to the trashcan for this action.
- We used code inspired by the Catamari example from class to implement this feature.

Particle System

- The particle system visually represents the vacuum's suction power by emitting particles when Abu activates the vacuum.
- To make the emitted particles form a cone shape, we applied trigonometry, calculating angles to direct the particles based on Abu's facing direction.
- To optimize performance, every time the vacuum is deactivated, we removed old or inactive particles from the system to maintain smooth gameplay.

Scoring System

- The game environment contains 50 pieces of trash. As Abu collects trash, the amount currently in the vacuum is displayed in real time.
- Players can empty the vacuum when near a trashcan, which updates the displayed count. Additionally, as more trash is placed in the trashcan, a secret graphic change occurs.
- The main objective is to collect all 50 pieces of trash scattered across the environment, regardless of whether the vacuum is emptied.

Customization Features

- Change Outfit: Pressing the "C" key allows players to change Abu’s outfit.
- Particle Color: Players can adjust the color of the vacuum particles through an interaction slider.
- Player Speed: Abu’s movement speed can be customized using a slider.
- Background: The game’s background can be changed through the interaction slider for a more personalized experience.

Challenges

Performance Optimization:
One of the main challenges was optimizing performance due to the accumulation of particle objects over time. This issue was addressed by implementing a system that flushed out inactive particles, ensuring smooth gameplay.

Vacuum Collision Box:
Creating an invisible box to follow the vacuum head presented a challenge, particularly because Abu has both left- and right-facing character designs. In left-facing mode, the vacuum is positioned on the far left side of Abu, while in right-facing mode, it shifts to the far right, requiring precise adjustments to the collision box.

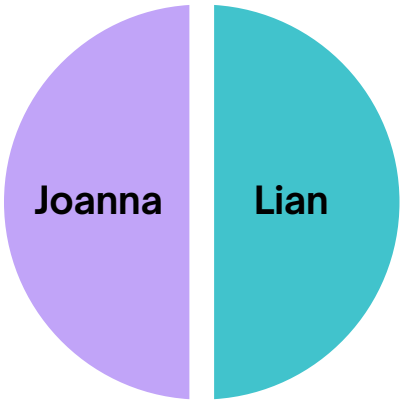
Contribution Map

Joanna

- Planned and designed all game assets.
- Developed trashcan collision detection to enable trash emptying functionality.
- Contributed to the particle system, including the design of particle behavior.
- Led particle system performance optimization.
- Worked on the customization features for background and player speed slider

Lian

- Led the implementation of vacuum collision detection.
- Contributed to brainstorming and planning the core game mechanics.
- Developed the trash collection mechanics.
- Worked on the particle system to enhance visual effects.
- Implemented the customization features, including resetting and particle color changes.



Asset List

Sound:

[MapleStory BGM] Kerning City: Bad Guys) © Copyright Wizet, Nexon

- Bgm.mp3:

Pixabay: *pixabay.com/sound-effects/search/game/*

- trashbag.mp3
- vacuum.mp3
- victory.mp3
- ding.mp3

Image:

Pexels: *www.pexels.com*

- conBG.jpg

Canva: www.*canva.com*

- moonBG.jpg
- litterboxBG.jpg
- woodBG.jpg

Graphic Assets:

- All trash and trashcan assets were created by Joanna Yoo using ***Canva resources***.
    - *Learn more about Copyright ownership of designs made in Canva: www.canva.com/help/copyright-design-ownership/*
- All player images are original artwork, created by Joanna Yoo using Adobe Illustrator.
- The Labcat image was used from the example 2 starter code.