

CMPS 160 - Spring 2019

Final Project

Members: Thanut (Art) Parkeenvincha, Terence So

# Advanced Methods Implemented

# Hit Collision:

* Using the algorithm known as axis aligned bounding box (AABB vs AABB), each geometry in the scene is equipped with a center point variable and a half width variable.
* These two variables are used to calculate a bounding box around the object.
* To detect collisions, it must match its own bounding box with the other geometry’s boxes.
* If a collision has occurred for the player, we send out “ghost states” that check to see where the player can (up, down, left, right) be a frame later. If those ghost states are still colliding with the object, then the player cannot move in that direction.
* If a collision has occurred for a villager, they simply face 180o from where they were standing and walk in the opposite direction.

# HUD:

* The bottom most layer is our WebGL component, while the two layers on top are our HUD canvases.
* The reason for separating them into two layers is because one is reserved for art and assets while the top most layer is used for text.
  + Due to how canvases are rendered, this separation is required.

# Dynamic Music and Filters:

* The time, music, and colors of the game are all determined by the date and time in the real world.
* Music:
  + Depending on the hour of the day, the class AudioController picks the music that corresponds with that hour.
  + This feature was implemented in the original Animal Crossing.
* Color filters:
  + Throughout the day, the colors of the game are normal.
  + But at 5 - 6AM and 5 - 8PM in the real world, a sepia filter is applied throughout the entire game.
  + And at night, a dark blue filter is applied as well.
  + This was done by a weighted matrix that converts each RGB component to its respective filter output.

# Linear Interpolation of Movement and Camera:

* Linear interpolation is used throughout the game to make it feel smoother.
* Turning the player around does not happen instantaneously, instead, the angle at which the player is facing is linearly interpolated to its target direction.
* Linear interpolation is also used for controlling the camera during dialogue with villagers.
  + When starting conversation, the camera linearly interpolates the zoom and the y axis to create a smooth pan down.

# Particles:

* A transforming object is used to create the illusion of a volumetric particle.
* The object’s position/scale/rotation are tied to input states to mimic the dust particles that appear when the player runs in Animal Crossing.
* The same particle is seamlessly recycled, which makes the system relatively inexpensive.

# 