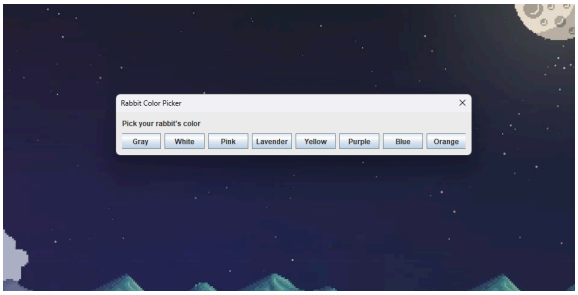
Readme.pdf for A3- Bunny Business

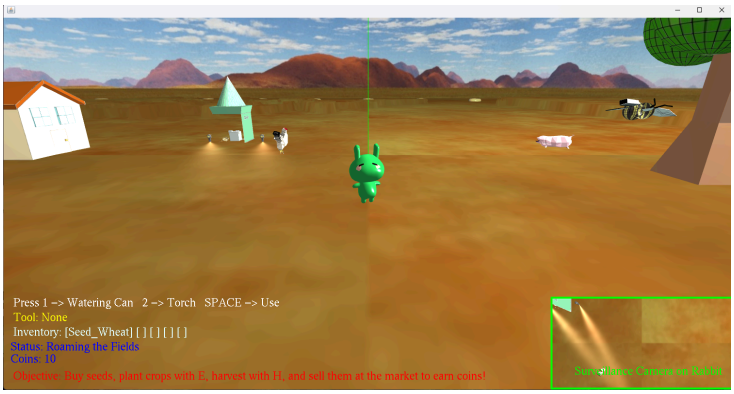
1.) Bunny Business,
   a.) Team Members & Section: *Isabel Santoyo-Garcia & Danica Galang*, CSC165-02
   b.) Farming by day and fending off unexpected threats from sudden bee attacks and roaming NPC animals.
2.) Screenshot of typical screen of the game

Image Explanation:

(1)



(2)



(3)

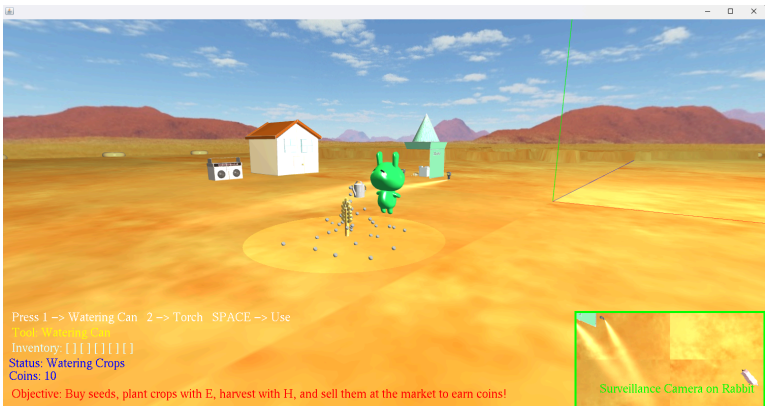


Image Explanations:

(1) When running the game, it first provides a pop out asking the player what rabbit color they want to play. This will be sent to all the ghost avatars.

(2) This is how the game will look once you spawn in.

- Animal NPCs are roaming (bee, pig, and chicken)
  - You can hear them all make noise, if they are near the player's avatar
- The bee will begin orbiting the tree and will harm the player if they get near
- The HUD message is there to provide information: objective, status, inventory/tool indicator, and coin collection.

Additionally, this image captures an overhead view of the avatar.

(3) This image shows a grown plant that became a crop (wheat) from the player watering them. The inventory is updated and more buildings are seen.

3.) instructions for compiling and running your game, including the network server

We did a majority of the compiling & ran the game through VSCode. On command prompt:

- (1) clearTAGEclassfiles
- (2) buildTAGE
- (3) compile
    - javac -Xlint:unchecked a3/*.java
- (4) server
    - javac -cp .;tage.jar tage/networking/server/NetworkingServer.java
    - java -cp .;tage.jar tage.networking.server.NetworkingServer 6028 UDP
- (5) run
    - java --add-exports java.base/java.lang=ALL-UNNAMED --add-exports java.desktop/sun.awt=ALL-UNNAMED --add-exports java.desktop/sun.java2d=ALL-UNNAMED -Dsun.java2d.d3d=false -Dsun.java2d.uiScale=1 a3.MyGame [insert IP address] 6028 UDP

But, you can also click in that exact order on the .bat programs. I made the server a batch file, to make it have easier access.

4.) How to Play & Score

Objective: Start with 10 coins and improve your farm by buying seeds, planting crops, harvesting, and selling produce at the market.

How to Play:

- Buy seeds at the market (press M near the market → choose “Buy” → select Carrot or Wheat).
- Plant a seed you’ve bought by pressing E.
- Wait for it to grow (Wheat takes 30 seconds while Carrot takes 45 seconds). A green spotlight appears over any fully grown crop.
    - Watering the plant will reduce the time
- Harvest ready crops by standing within range and pressing H. Harvested items go into your inventory.
- Sell at the market (press M → “Sell”), choosing to sell everything or just your crops.

Warning & Collisions:

- Animals: Chickens and pigs roam freely, if they bump into you, you’ll momentarily be pushed back.
- Bees: If the player gets too close to the tree, the bee AI will pursue you, triggering a knock-back and temporary “stun.”
- Environment: Buildings (home, market, radio) and terrain borders use collision detection, so you and the NPCs can’t walk through them.

Scoring (Coins):

- Initial balance: 10 coins
- Seed purchase: 2 coins each (Carrot or Wheat)

Selling:
- Wheat → 5 coins
- Carrot → 10 coins

Selling unused seeds: 1 coin each

Your “score” is simply your coin balance.

5.) Keyboard & gamepad controls for the game

Avatar Controls:

Input	Action
Keyboard: W Gamepad: Left Stick Up	Move the rabbit forward
Keyboard: S Gamepad: Left Stick Down	Move the rabbit backward
Keyboard: A Gamepad: Left Stick Left	Turn (yaw) the rabbit left
Keyboard: D Gamepad: Left Stick Right	Turn (yaw) the rabbit right
Keyboard: Q	Toggle X,Y,Z axis lines on & off
Keyboard: 1 Gamepad: Button Z (7)	Equip watering can
Keyboard: 2 Gamepad: Button Z (7)	Equip torch
Keyboard: SPACE Gamepad: Button Y (3)	Use selected tool (water or torch) or used for selecting an option in market
Keyboard: E Gamepad: Button X (0)	Plant a seed
Keyboard: H Gamepad: Button B (2)	Harvest nearest ready crop
Keyboard: M Gamepad: Button A (1)	Open / interact with the market’s menu & interact with buildings/radio

Keyboard: ESC	Quit the game

Camera Controls:

Input	Action
Keyboard: Arrow UP	MAIN Camera viewport: - Move camera up (orbit elevation)
Keyboard: Arrow DOWN	MAIN Camera viewport: - Move camera down (orbit elevation)
Keyboard: Arrow LEFT	MAIN Camera viewport: - Rotate orbit camera left (azimuth)
Keyboard: Arrow RIGHT	MAIN Camera viewport: - Rotate orbit camera right (azimuth)
Keyboard: I Gamepad: D-Pad UP	Mini (side) Camera viewport: - Pan overhead camera up
Keyboard: J Gamepad: D-Pad LEFT	Mini (side) Camera viewport: - Pan overhead camera left
Keyboard: K Gamepad: D-Pad DOWN	Mini (side) Camera viewport: - Pan overhead camera down
Keyboard: L Gamepad: D-Pad RIGHT	Mini (side) Camera viewport: - Pan overhead camera right
Keyboard: P Gamepad: RY axis (press & release)	Mini (side) Camera viewport: - Reset viewport camera offsets
Keyboard: U Gamepad: C-Stick RIGHT	Mini (side) Camera viewport: - Zoom in (decrease orbit radius)
Keyboard: O Gamepad: C-Stick LEFT	Mini (side) Camera viewport: - Zoom out (increase orbit radius)

6.) Description of the Lighting Used

- **Global Ambient Light:** A constant low level ambient sets the base illumination for the entire scene. Every object in the scene receives this to prevent totally pitch black shadows.
- **Chase Light:** A red-tinted light is attached to the rabbit when the AI NPC bee is actively pursuing. It's disabled by default and enabled only while chasing, turning off again when the bee backs off.
- **Market Lamps:** Two warm lamps placed on either side of the market. These are always on once the market scene is built.

- **Torch Light:** A positional, flickering style torch light that follows the rabbit. It's off by default and toggles on/off whenever you equip and use the torch.
- **Crop Spotlights:** Whenever a planted crop matures, a spotlight is spawned above it. That spotlight is removed when the crop is harvested.

Dynamic Enabling/Disabling:
- Chase Light turns on/off with AI NPC pursuit.
- Torch Light toggles with the torch tool.
- Spotlights appear for matured crops and disappear on harvest.

7.) Summary of Changes to the Network Protocol

We are still using the **GameServerUDP**.java.
The message types includes:
- join , which adds the clients to its list, then sends the client all existing avatars & current skybox index
- create , spawns a new ghost avatar with its chosen color and broadcasts each new avatar's creation so everyone can spawn a ghost rabbit.
- move , server broadcasts avatar movement to all other clients
- rotate , broadcasts avatar orientation when turning
- bye , broadcasts to everyone that a client is leaving and removes that ghost avatar
- skybox , synchronize time of day, so all clients switch to the same skybox texture
- water , synchronize the player's watering can on/off across all clients
- torch , toggles the torch in everyone's scene
- plant , harvest , grow , keeps every client's crops in sync
- beeAttack (via the AI server), delivers NPC knock back impulses when the bee attacks

We ran an AI server (**GameAIServerUDP**) alongside the main game server that periodically broadcasts whether the bee is close to the avatar, if so it decides to strike one client.

On the client side, **ProtocolClient** is the client class that handles all message traffic:
- Sends the message types to the main GameServerUDP
- And communicates with GameAIServerUDP for the isnear/beeAttack flow.

8.) Changes/Additions to TAGE
- Expanded the HUDmanager.java
    - Grew the HUD system from 4 to 12 distinct display slots for the market HUD options, viewport, and just to have instruction on the screen.
    - Each slot now has its own color property (HUDnColor) and setter method (setHUD(text, color, x, y))
    - Updated the drawHUDs() method to render the each HUD string on the screen

- Added globalYaw to GameObject.java
    - Allows rotation around the global y-axis instead of local up vector
    - Implemented in TurnAction.java

9.) A statement indicating the genre, theme, dimensionality, & activities utilized in the game

Genre: Multiplayer Simulation
Theme: Farming Adventure

A 3D farming adventure where we are nurturing growth through the risk and reward of agricultural exploration. Farming by day and fending off unexpected threats from sudden bee attacks and roaming NPC animals.

Dimensionality: 3D with multi viewport support
Activities:

- Farm lifecycle: plant, water, grow, and harvest with a supportive spotlight once the crop matures
    - Observing and interacting with roaming NPC animals
- Economy: buy/sell seeds and produce at the market
- Exploration: avatar can roam around varied terrain with texture mapping and changing skyboxes within every 50 seconds.
- Camera navigation: Orbit cameras, pan/zoom mini viewport
- Tools: toggle watering can and torch that interacts with lighting
- Hostile mobs: fending off sudden bee attacks launched by the AI driven NPC bee

10.) List of where each Project Requirement is visible

External models (custom OBJ + UV-wrapped textures)

- Rabbit.obj + texture attached
    - Before starting the game, the player gets to choose the rabbit color texture they want to be spawn in with. That rabbit's texture is used to load the model in the game. Can be shown once the player spawns in.
- Plant.rkm/rks, carrot.obj, & wheat.obj + texture attached
    - When the player plants seeds from their inventory, the plant model is then placed on the ground with a small animation. Once sprouted or matured, it takes around thirty to forty five seconds, it becomes either a carrot or wheat model depending on the seed they planted.
- Home.obj + texture attached
    - Player's home is shown right next to the market behind the avatar. It's the player's home base which is used to refresh the skybox & terrain texture in the morning.
- Market.obj + texture attached
    - Located right behind the avatar. Once near, the player can buy and sell items.
- tree.obj + texture attached
    - The tree is located on the right side of the avatar, it's used as an anchor point for the bee's orbit
- Torch.obj + texture attached

    - When the tool is activated (on keyboard click 2 then SPACE to use), the player can use the torch to light up the area
- Wateringcan.obj + texture attached
    - When the tool is activated (on keyboard click 1 then SPACE to use), the player can use the watering can to water plants.
- Lamp.obj + texture attached
    - Two of these models are shown right next to the market model.

Networking Multiplayer

- Ghost avatars: when two (or more) players join, you'll see their rabbit avatars roaming around in real time, each rendered with the texture they picked
    - Avatar's movement and rotation is synced
    - When the client is disconnected, it removes the avatar as needed.
- When planting each player can see it and when harvesting, each client can harvest no matter if they planted it or not.
    - It disappears and renders as it should
- Tools & their active physics can be shown across clients
- Skyboxes and terrains are synced

Note:

- Coins, attacks, NPC spawning are all local to your machine and not shown in multiplayer.

Skybox and Terrain

- *Skybox***:** There's eight different cube map textures that's in a loop. It takes 50 seconds to cycle through one to another. However in the eveningTwoTerrain.jpeg (the last one in the cycle) it stalls until the avatar interacts with the home model to reset the skybox cycle.
- *Terrain*: The ground is tiled with a hills.jpg texture which creates hills on the terrain plane.

Lights

- *Global Ambient Light***:** A constant low level ambient sets the base illumination for the entire scene. It's the main directional light that every object in the scene receives to prevent totally pitch black shadows.
- *Chase Light***:** A red-tinted point light attached to the rabbit when the AI NPC "bee" is actively pursuing. It's disabled by default and enabled only while chasing, turning off again when the bee backs off.
- *Market Lamps***:** Two warm lamps placed on either side of the market. These are always on once the market scene is built.
- *Torch Light***:** A positional, flickering-style torch light that follows the rabbit. It's off by default and toggles on/off whenever you equip and use the torch.
- *Crop Spotlights***:** Whenever a planted crop matures, a spotlight is spawned above it. That spotlight is removed when the crop is harvested.

HUD

- *Starting to from the bottom left to up:* States the objective of the game, coin indicator, status report, inventory indicator, tool indicator
- *Menus:* When you press M on the keyboard near the market, home or radio you get a HUD menu with options that the player can interact with.

3D Sound

- *Bee buzzing*: as you approach within 15 units of the bee NPC, you hear a 3D buzz that moves with its position
- *Watering can droplet*: toggling the watering tool plays a looping water pour sound
- *Torch fire noise*: toggling the torch tool plays a fire sparkling sound
- *Animal sounds*: pigs oink and chickens cluck when they move around.
- *Background music*: flip the radio on/off to start/stop a looping background stream from the radio's location

Hierarchical SceneGraph

- *Tools parented to rabbit:* The watering can and torch are attached as children of the rabbit. So whenever the rabbit moves or turns, the tools follow.
    - To activate, press 1 or 2 to select the tool and then SPACE to activate
- *Bee orbits around the tree*: The bee is attached as a child to the tree, which represents that it's moving and rotating the tree.

Animation

- *Bee.rka*: the bee model plays a looping "FLY" animation around the tree from the very start. It looks like a side to side animation.
- *Chicken.rka & pig.rka*: Both chicken and pig use keyframe animation for idle and walking. It cycles through both, so when the game can start with either one with any speed.
- Plant.rka: when planted, the plant will show a very subtle animation that mimics a ruffling of leaves.

NPCs

- *Bee NPC*: the AI controller makes the bee circle then pursues the avatar on sight when they are near the tree, with the chase light following.
- *Roaming animals*: the pig and animal are driven by their animation to wander around the field

Physics

- *Water droplets*: when you water, each droplet is a physics object, so the drops fall, bounce off the ground or plants, then disappear after a short time.
- *Bee knockback*: if the bee hits you, a force is applied to your avatar's physics body, causing a shove/stun before you recover.
    - This concept is similar when the roaming animals run into the avatar

- *Collision bounds*: you can't walk through the home or market. If you try, the avatar is knocked back.

11.) A list of the requirements that you weren't able to get working

All features are included! The only issue is that the game is refusing to run on any PCs… I was only able to get it to work on multiple laptops, but when I try to run it on any PCs, including the lab machines, it runs into errors and the game just crashes.

Because of this, I recorded a gameplay of myself playing the game and testing all the features including multiplayer.

- Single Player POV, using the keyboard (which is displayed) to show all the features of the game. All NPCs are animated when moving, including the plant that flutters its leaves once in a while.
    - https://youtu.be/1U8hqmJMSOs
        - NOTE: since the servers weren't running, the AI NPC bee didn't send a knock back message to the rabbit, but this is later seen in the other two videos.
- Single Player POV, using the gamecube controller. The youtube video includes both the gameplay and the gamepad in action! Features are the same as the first one, but it shows how accurate the input handling is on both devices.
    - https://youtu.be/EE1yK9EfayI
- Multiplayer POV, it shows how your client can see the interactions of the ghost avatar. From being able to plant & collect the crops without any issues & tools being shown across scenes.
    - https://youtu.be/dU4VA2lTTHA

12.) Any technique you used in your game that goes beyond the requirements

- I really like how the physics for the watering can droplets look! It took us a while to make it look smooth, mimicking the technique of realistic water droplets.
- Also we made sure the terrain's textures matched each cycle of the skybox's textures to maintain consistency in the game's ambient environment.

13.) Contributions of Team Members

Isabel

- I implemented the skyboxes, physics, networking, terrain mapping, NPC AI, sound, & lights logic.
- I created these models & their textures: bee.obj, carrot.obj, chicken.obj, home.obj, lamp.obj, market.obj, pig.obj, plant.obj, radio.obj, torch.obj, tree.obj, watercan.obj, wheat.obj
    - And animated these: beeFly.rka, chickencurious.rka, chickenwalk.rka, pigcurious.rka, pigwalk.rka, plantmove.rka

Danica

- Updated HUD messages & adapted the gamepad controllers
- I create this model & the following textures: rabbit.obj, bluerabbittx.jpg, brownrabbittx.jpg, grayrabbittx.jpg, greenrabbittx.jpg, lavenderrabbittx.jpg, orangerabbittx.jpg, pinkrabbittx.jpg, purplerabbittx.jpg, rabbittx.jpg, yellowrabbittx.jpg

14.) <u>List of assets that we created</u>

We created the following assets:

Animation

- bee.rkm, bee.rks, beeFly.rka
- chicken.rkm, chicken.rks, chickencurious.rka, chickenwalk.rka
- pig.rkm, pig.rks, pigcurious.rka, pigwalk.rka
- plant.rkm, plant.rks, plantmove.rka
    - These were created using blender's timeline & dope sheet

Models

- bee.obj, carrot.obj, chicken.obj, home.obj, lamp.obj, market.obj, pig.obj, plant.obj, rabbit.obj, radio.obj, torch.obj, tree.obj, watercan.obj, wheat.obj
    - These were created using blender

Skyboxes

- dayFour, dayOne, dayThree, dayTwo, eveningOne, eveningTwo, night, sunset
    - These were created using Terragen

Texture

- beetx.jpg, bluerabbittx.jpg, brownrabbittx.jpg, carrottx.jpg, chickenttx.jpg, dayFourTerrain.jpg, dayOneTerrain.jpg, dayThreeTerrain.jpg, dayTwoTerrain.jpg, eveningOneTerrain.jpg, eveningTwoTerrain.jpg, grayrabbittx.jpg, greenrabbittx.jpg, hills.jpg, hometx.jpg, lamp.jpg, lavenderrabbittx.jpg, markettx.jpg, nightTerrain.jpg, orangerabbittx.jpg, pigtx2.jpg, pinkrabbittx.jpg, planttx.jpg, purplerabbittx.jpg, rabbittx.jpg, radiotx.jpg, torchtx.jpg, treetx.jpg, wateringcantx.jpg, wheattx.jpg, yellowrabbittx.jpg
    - A majority of these files were from blender UV mapping & colored in using microsoft paint or Terragen for skyboxes

15.) List of assets that we did not create + evidence of permission

Sounds

- backgroundMusic.wav
    - Under creative commons, it states that we can copy, modify, or perform the sound without the need to ask permission from another.
        - https://freesound.org/people/josefpres/sounds/659017/?

- beebuzz.wav, chickennoise.wav, fireSound.wav, pigoink.wav, water.wav
    - These are under the website called pixabay.com, where it states we can use the sounds for free if it's not for prohibited uses.
        - https://pixabay.com/service/license-summary/

16.) which RVR-5029 lab machines (at least two – it's networked!) on which your program was tested

I tried testing on ECS_SONIC & ECS_SPACEINVADE, but it wasn't working. There seems to be an error with the game running on PCs.

But, it works on laptops!

As stated before, because of this, I recorded a gameplay of myself playing the game and testing all the features including multiplayer.

- Single Player POV, using the keyboard (which is displayed) to show all the features of the game. All NPCs are animated when moving, including the plant that flutters its leaves once in a while.
    - https://youtu.be/1U8hqmJMSOs
        - NOTE: since the servers weren't running, the AI NPC bee didn't send a knock back message to the rabbit, but this is later seen in the other two videos.
- Single Player POV, using the gamecube controller. The youtube video includes both the gameplay and the gamepad in action! Features are the same as the first one, but it shows how accurate the input handling is on both types of devices.
    - https://youtu.be/EE1yK9EfayI
- Multiplayer POV, it shows how your client can see the interactions of the ghost avatar. From being able to plant & collect the crops without any issues & tools being shown across scenes.
    - https://youtu.be/dU4VA2lTTHA