

# Documentation

**Group/Game Name:** Snack Attack

## **Brief description of implementation:**

Our game features a third-person camera system centered around the main character, Cooper the Corgi. The camera follows our main character Cooper. This setup ensures players maintain visual context and spatial awareness during gameplay.

The game world includes several 3D background elements for worldbuilding, such as one big floating island and other assets, which contribute to a visually engaging environment and function as obstacles. Cooper moves using the WASD keys and can jump with the SPACE key.

The game window can be resized using adjustable parameters configured in an external settings file. This allows users to tailor the resolution and window size to their needs.

The goal of the game is to help Cooper collect 10 food items in 2 minutes. The food items fall randomly from the sky, and he has 5 seconds to collect them before they disappear after hitting the ground.

**Note to Submission:** We handed in our last version of the project (with the currently not properly working particle system). We didn't want to merge the particle branch with our main branch to avoid destroying everything. However, the latest version, also with the code from the development branch, is available in the particle branch. Depending on the feedback, we will then work on it further till the game event. Thank you! 😊

## **Additional libraries:**

### [TinyObjLoader:](#)

Used to load .obj model files for the 3D environment and character assets.

### [FreeType:](#)

Used to render simple text elements (e.g., counter, timer, ...) on display. (Reference: <https://learnopengl.com/In-Practice/Text-Rendering>)

### [Bullet Physics:](#)

Used for basic physics and collision detection.

Gameplay:

Mandatory:

- **3D Geometry:**

- Static: a world with obstacles that make collecting food more challenging (rocks, trees, decorative elements, etc.)
- Dynamic:
  - Cooper the Corgi (3rd person main character)
  - Falling food from the sky
- **Playable:**
  - WASD: Movement of Cooper
  - Space: Jump
  - ESC: Quit
  - F1-F9: Visual effects
  - P: Restart game
- **Min 60 FPS and Framerate Independence**
- **Win/Lose Condition:** Cooper needs to collect 10 food items within 2 minutes to win the game
- **Intuitive controls:** see above at playable
- **Intuitive Camera:** see above at playable
- **Illumination model:** Sun/ Moon (directional light) that illuminates the game scene
- **Textures:** different textures for the objects (incl. procedural texturing)
- **Moving Objects:** falling food and the main character Cooper
- **Documentation:** -
- **Adjustable Parameters:** window resizing, number of items to collect, and timer

Optional:

- **Collision Detection (Basic Physics):**
  - collision detection, so Cooper cannot walk through obstacles
  - used to detect collisions between Cooper and food items

Effects:

#### **Advanced Modelling:**

- CPU Particle System: sparkling animation when collecting a food item for better user feedback

#### **Animation:**

- Hierarchical Animation: wobbling basket on Cooper's back when he is moving

#### **Texturing:**

- Procedural Texture: marbled texture for realistic rock generated by procedural shader

#### **Shading:**

- Physically Based Shading: trashcan in the middle of the island has a shinier surface than other objects

#### **Post Processing:**

- Bloom/Glow: highlights specific objects in the game, golden items bring 2 points instead of 1, and the moon for a cozy atmosphere (Reference: <https://learnopengl.com/Advanced-Lighting/Bloom>)

Other special features:

- When Cooper is falling from the island, his position is reset
- Text rendering to give user feedback (e.g., won or lost game, time, and items to catch left)

Walk-through: not much to explain here, start the game, move Cooper, and try to collect the items within the specific timeframe