

# Documentation

Group/Game Name: Bake the Cake!

Brief description of implementation:

Players control Red Riding Hood using WASD for movement and Spacebar to jump, navigating a small forest level containing trees, mushrooms, and collectible fruits. Approaching fruits automatically collects them. A timer tracks gameplay duration, and clearly defined win/lose conditions are included. An intuitive third-person camera smoothly follows Red Riding Hood. The cursor is hidden during gameplay but becomes visible when accessing the menu.

Additional libraries:

- GLFW 3.3.x – Window & Input (<https://www.glfw.org>)
- GLEW 2.2.x – OpenGL-Function Pointer management (<https://github.com/nigels-com/glew>)
- GLM 0.9.9.x – Maths (<https://github.com/g-truc/glm>)
- stb\_image.h – Texture Loading (<https://github.com/nothings/stb>)
- Bullet Physics 3.24 – Physics & Collision Detection (<https://github.com/bulletphysics/bullet3>)
- INIReader 1.5 – Configuration Parser (<https://github.com/jtilly/inih>)
- Assimp 5.4.3 – Asset Importer Library (<https://github.com/assimp/assimp>)

Gameplay:

Mandatory:

- 3D Geometry: 3D Geometry (6 points): External OBJ models loaded via Assimp instead of primitives
- Playable: Fully playable with movement, timer, and collection mechanics
- Advanced Gameplay: Gameplay complexity increased through timer, collectible mechanics, and structured level design
- Min 60 FPS and Framerate Independence: Movement adapted to deltaTime
- Win/Lose Condition: Time limit and collected fruits define win/lose scenario
- Intuitive controls: Movement with WASD, jump with Spacebar, continuous input handling, and GLFW callbacks
- Intuitive Camera: Smooth third-person camera with orbit and follow functionality
- Illumination model: Implemented lighting model with light sources, materials, and normals
- Textures: Textured assets throughout the level
- Moving Objects: Red Riding Hood moves around the level
- Documentation:
- Adjustable Parameters: Screen resolution and fullscreen mode adjustable

Optional:

- Collision Detection (Basic Physics): Red Riding Hood collides with trees; fruit collisions are correctly detected
- Advanced Physics: Collision trigger reduces berry count and removes collected fruits

- Scripting Language Integration: x
- View-Frustum Culling: x
- Heads-up Display: HUD overlay displays relevant information

#### Effects:

##### Lighting:

- Lightmap using Separate Textures: x
- Lightmap using In-Game Calculation: x
- Shadow Map with PCF: x
- Shadow Volumes: x

##### Advanced Modelling:

- CPU Particle System: x
- GPU Particle System using Transform Feedback: x
- GPU Particle System using Compute Shader: x
- L-System: Used to generate and render bushes procedurally
- Blobby Object Using Marching Cubes: x
- Subdivision Surface: x

##### Terrain:

- Tessellation from Height Map: x
- Voxel Terrain using an Octree: x

##### Animation:

- Hierarchical Animation: x
- Vertex Shader Animation: Vertex shader animates water surface with wave simulation
- GPU Vertex Skinning: x

##### Texturing:

- Procedural Texture: x
- Video Texture: x
- Specular Map: x
- Environment Map: Reflection simulation on water surface using cube map

##### Shading:

- Simple Normal Mapping: x
- Cel Shading: x
- Style Transfer: x
- Brush Strokes: x
- Physically Based Shading: x

##### Advanced Data Structures:

- BSP Tree: x
- kd-Tree: x
- LOD using an octree: x

##### Post Processing:

- Bloom/Glow: Fruits exhibit glowing bloom effect
- Lens Flares: x
- Contours via Backfaces: x

- Contours via Edge Detection: x

Other special features:

Keybindings:

- W/A/S/D = Forward / Left / Backward / Right
- Space = Jump
- ESC = Exit
- F1/F2/N/T = Debug modes
- F = Toggle fullscreen

Walk-through:

- Navigate to bin/ and start the game with ./GCGProject\_GL.
- The game launches in 1280×768 resolution at  $\geq 60$  FPS.
- Red Riding Hood starts at position (0, 0, 0) on the ground plane, camera at (0, 2, 10).
- Move using WASD and jump with Space.
- Collect fruits in the forest and keep an eye on the timer.