

**GAZI UNIVERSITY**  
**FACULTY OF ENGINEERING**  
**COMPUTER ENGINEERING DEPARTMENT**



**Burak ÇETİN – 22118080032 – brakcetin660@gmail.com**  
**İsmail Görkem ULAŞ – 21118080007 – gorkem17ulas@gmail.com**

**MEDIEVAL CASTLE SCENE**  
**PROJECT PROPOSAL**

**CENG376 – COMPUTER GRAPHICS**

**MAY 2025**

# TABLE OF CONTENTS

TABLE OF CONTENTS .....	i
MEDIEVAL CASTLE SCENE .....	1
<b>Introduction</b> .....	1
<b>Description</b> .....	1
REFERENCES.....	2

# MEDIEVAL CASTLE SCENE

## Introduction

This project presents an interactive 3D medieval castle environment developed using WebGL. The user can move and rotate the camera, interact with objects like a catapult and torches, switch between day and night modes, and earn points by performing various actions. The project features dynamic lighting, textured models, and a responsive interface. Technologies such as Three.js, GLTFLoader, and dat.GUI will be used to deliver the requests of the assignment.

## Description

In this project, a medieval castle environment will be created using WebGL. The objective is to provide user with an interactive 3D scene that responds to real-time inputs.

The castle scene includes a fortress structure, a catapult, decorative and functional torches, and stones that can be interacted with. The user can explore the area by moving the camera freely along the x, y, and z axes and rotating it in all directions.

The scene will feature objects with at least three distinct morphologies:

- **Castle walls and towers** – Made from textured rectangular stone blocks
- **Catapult** – A mechanical object with complex geometry
- **Stone projectiles** – Rounded or cylindrical in shape
- **Torches** – With cylindrical handles and animated flame textures

These models will use different textures. Interactive elements include picking up and launching stones using the catapult, as well as carrying a torch to illuminate dark areas. Fixed torches on the castle walls can be toggled on or off through user interaction. Torch brightness can be adjusted using a simple UI control.

A **day/night toggle** is included as one of the required features. During **day mode**, a sun model serves as the primary light source, which can be repositioned and adjusted in brightness. During **night mode**, torches become the primary light source, casting warm lighting throughout the scene.

The project also incorporates a **point-based scoring system**. User earns points for interacting with specific elements like lighting torches or successfully using the catapult. Some actions (such as missing a shot) may reduce points. The score will be displayed on a panel in the interface.

This project meets all minimum assignment requirements, including WebGL rendering, 3D scene construction, camera movement and rotation in three axes, object interactivity, multiple textured object types, and a controllable light source. The final product aims to combine educational value and technical achievement with a fun, game-like environment.

## Technologies to Be Used:

- **Three.js** – For rendering the 3D scene and managing camera, lighting, and object interactions
- **GLTFLoader** – To import and display GLTF/GLB format 3D models
- **OrbitControls / PointerLockControls** – To allow full camera movement and orientation
- **dat.GUI (optional)** – For modifying scene parameters like light brightness in real-time
- **HTML / CSS / JavaScript** – For UI controls such as toggles, buttons, and the score display
- **VS Code + Live Server** – For project development and local testing

## REFERENCES

[1] *three.js docs*. (n.d.). <https://threejs.org/docs/>.

[2] Dataarts. (n.d.). *GitHub - dataarts/dat.gui: Lightweight controller library for JavaScript*. GitHub. <https://github.com/dataarts/dat.gui>.