Minecraft

Introduction

Minecraft is a 3D sandbox video game that has no required goals to accomplish, allowing players a large amount of freedom in choosing how to play the game. The game also features an optional achievement system. Gameplay is in the first-person perspective by default, but players have the option of a third-person perspective. The game world is composed of rough 3D objects—mainly cubes, referred to as blocks—representing various materials, such as dirt, stone, ores, tree trunks, water, and lava. The core gameplay revolves around picking up and placing these objects. These blocks are arranged in a 3D grid, while players can move freely around the world. Players can "mine" blocks and then place them elsewhere, enabling them to build things.

Figure 1: Gameplay

Figure 2: Gameplay

Features

Physics engine

The physics or projectile motions will be applied to mobs such as chickens, cow, zombies, player,... consists of velocity, accelerations, pulse force, etc.

Gameplay

Building : Player can move, choose blocks/ cubes and build a building base on their imaginations and this building will be saved to that world.

Mine : Player can destroy block using tools provided (pickaxe or just by hands). And get that block on their inventory. The map will be generate that there will be materials like iron or diamond under the ground.

Infine / Finite Map Generator : Infinity for Creative mode and Finite for Survival mode.

Boss Fight : implement some strong weapons as sword, axe, pickaxe,... and some (or a) boss to fight with in its arena to win the game.

Mobs

Each mobs have their own behavior that suit with the real life creatures. Some mobs will attack player and some will be help full.

Material

Some types of blocks :

- Dirt
- Stone
- TNT
- ...

Each of these has their own properties.

Implementation idea

Language

The game will be developed on C++, with the help of CMake for compilation.

OOP aplications

OOP will be applied to following objects:

- PLAYER
- States
- Resource holder
- GUI
- so on

External Libraries (expected)

- GLFW(Graphics Library Framework): a lightweight utility library for use with **OpenGL**. It provides programmers with the ability to create and manage windows and **OpenGL** contexts, as well as handle joystick, keyboard and mouse input