Documentation for my interpretation of Minecraft2d

Moritz Siefke

Contents

1 Introduction

This is a school project and was done in early 2023.
The Idea is to make a very easy version of a game I realy like called Minecraft.
The original game is an openworld survival/adventure game with authentic block graphics developed by Mojang and sadly bought by Microsoft.

2 Usage

when started turn up simulation speed to max!!!
The player is controled with w, a, s and d or arrow keys for movement.
1,2,3,4,5,6,7,8,9,0 are to select the blocks available for you.

3 How is the Program build

Please keep in mind that this is my first Project in the concept of oop and that I would build the Programm diffrently now.

3.1 There are to main types of Classes

3.1.1 the Player(), which obviously is the Player

The player is affected by move instructions and has collision actors to the blocks it is to be near to blocks when placed and destroyed, it also has the hotbar, which is there to show items or blocks selected.

3.1.2 and different types of Block{s}()

1. **Block** - static block

2. **Fluid** - no collision, spreads

3. **Ladder** - no collision, doesn't spread

4. **Falling Blocks** - static block, but moves done every half second or so when possible

5. **Trapdoor** - static block, can open and close(toggle collision)

6. **TNT** - static block - rightclick on int calls explode function, which removes some blocks

7. **BedRock** - static indestructable block

8. **Grass** - static block, but is needed to check for tree spawning, since these are just allowed to spawn on grass

9. **HotbarSlot** - static indestructable is there to show selected item

10. **ItemPlaceholder** - is the same as block, but indestructable, is there to give a preview of block to place

3.2   And then there are Functions

there are functions,

3.2.1   when keys are pressed

1. **Keyboard** - check key pressend and then move or change active block

2. **Mouse Left** - destroy block

3. **Mouse Right** - place block or interact with placed block

3.2.2   for generating the world

generating the world is split into different functions

1. **Generating height table** - generate heights, wich are all jumpable and save them to list pass to:

2. **fill with blocks** - fill blocks to given heights, and:

3. **add layer of grass** - add a layer of grass if above water level, else sand if under or at water level

4. **finaly add trees, on grass blocks** - generate trees of different heights (stem height 2,3 or 4), max 3 trees

3.2.3   and actfunctions which are there each gametick

these are used to make blocks fall and spread the water, they are sadly called every time when a block is generated, otherwise water wouldn't spread so quickly, there would be a workaround thoug, making a bool, which checks first function call and just skips it, but that wouldn't be to nice eather since performance wise this statement needs to be checked each time afterwards.