

The Block Game from High School Math Class

Summary

We plan on rewriting the “Block Dude” game found on many TI-84’s in the early 2000’s as a JavaFX application. This popular game is fairly simple; it involves a player, or ‘dude’ who can move a square, or ‘block’ back and forth along the screen. The screen is made up of blocks creating stairs. This dude can step on any block which is no taller than he is, and can pick up any block so long as it is on his level. He can carry a block and set it down any level below him, equal to him or one higher than him. The objective is to alter the environment around the dude so as to facilitate entry into an otherwise inaccessible door.

Design Patterns

Conceptually, BlockDude can be split into three Model, Viewer, and Controller components. This separation is quite sensible, as changes in the Model are dependant only on user input, and the view is dependant only on the current state of the model. This will allow us to store all complicated logic within the Controller component, which in turn will allow us to quarantine software bugs in their respective component.

Pieces

- Model: These pieces describe the state of the game at any one point
 - Character Class - The Character Class describes the state of the player’s sprite, the ‘dude’ in the original TI-84 game. Attributes of the class will include the sprite’s position and whether or not he has a block. Methods of the class will allow our controller to modify the state of the sprite, such as it’s x,y position and the boolean which indicates whether or not the sprite is carrying a block.
 - Block Class - Levels are composed of block’s, however, some of those blocks are movable. The block class will describe the position of movable blocks.
 - Level Class - This includes the layout of each level the user encounters. Attributes will include coordinates (x,y) within the level, with some being valid points to move to for the character and others not. The class will also contain a finish coordinate for where the character must reach to complete the level.
- Controller Interface: This interface will provides methods for reading in users input and manipulating the state of the data. We may implement several different versions of the controller interface to allow for various user experiences. For instance advanced players may prefer to solely use the keyboard, while novices might find it easier to click on buttons with their mouse.
- Viewer: This class renders the Model. Additionally the Viewer will re-render the state upon changes in the position of the sprite and the position of the movable blocks.

Sketches

See attachment.

Work Distribution

Dan will work on coding up the classes for the actual game (character class, level class, etc). Phineas will work on implementing the controller interface (the classes and methods in which the user interacts with to control a game). West will work on graphics and the Viewer class.