

## Java Project - V1.0

This project has two steps. The first step will be due Friday, Nov. 7 and the second step will be due Dec. 12, the Friday of dead week. There will probably be points given for an early turn-in and there will be a (smaller) C++ project. You should begin working on the project immediately, planning it out. At the end of the two steps you will have created a “platform game” similar, in principle if not complexity, to the basic Nintendo™ Mario Brothers™ game.

The first step is a very simple platform game that is played on a flat surface. You will have a character, let's call him Char, whose goal is to get from the start of the game to the end. The screen should scroll with Char, e.g., if Char is moving right then things to the left should scroll off the screen and things to the right should scroll on. Greyed out items will be due for the second step.

### Char has the following abilities:

**Movement:** Char can move left or right. This should be controlled by the left and right arrow navigation keys on the keyboard. Char can also move up and down with the up and down arrow keys. If both the left or right key and the up key are pressed simultaneously, Char will jump forward or backwards (respectively). The rate of movement is up to you but it should be fast enough not to bore you and slow enough to be controllable. The speed of movement should be the same regardless of the machine you are running on, i.e. movement should be based on wallclock-time and not CPU speed.

**Shooting:** Char can shoot at other characters in the game. Char will shoot when the “s” key is pressed and the shots will travel N number of units, where you can pick N. If the “s” key is held down for some period of time (you should pick this interval to make the game playable) Char will shoot in automatic mode, i.e. shots will be fired continuously. Again, you can pick the rate at which shots will be fired but it should be fast enough to be effective and slow enough that the grader can tell that separate shots are being fired. After 10 shots are fired *continuously* in automatic mode the Char cannot fire again for one second. Every shot fired should be accompanied by a sound. Every time Char hits a creature there should be a different sound. Every time Char kills a creature there should be a different sound.

**Health:** Char will start out with a health rating of 20 and can have a maximum health rating of 40. Every unit of movement will increase Char's health by 1. For every 1 second that Char is motionless Char's health will increase by 5. Char's health will increase by +10 for every creature killed. Char's health will decrease by -5 every time Char is hit.

**Interaction with the environment:** Char will interact with other creatures and magic items in the game environment at outlined below.

### **Magic Items:**

**Mushroom:** Bumping into a mushroom raises Char's health rating by 5 and the mushroom disappears. A sound unique to mushrooms should play when it is bumped into.

**Apple:** Bumping into an apple lowers Char's health rating by 5 and the mushroom disappears. A sound unique to mushrooms should play when it is bumped into.

**Star:** Bumping into a Star makes Char invincible for either 1 second or 10 units of measurement and the star disappears..

**Creatures:** There is a single kind of creature (called Creature) and it doesn't like Char. If Char enters the same screen as a Creature it will wait for 1/2 second or until Char moves two units and then it will begin shooting at Char. A Creature will shoot at 1/2 the rate that Char does when the "s" key is held down. It will shoot straight ahead from whatever height you make Char. It's height should be less than or equal to Char's so that it can hit Char. The Creature will advance on Char at whatever speed you desire after it begins shooting. If Char hits a creature, either because Char ran into the creature or the creature ran into Char, Char dies.

**Blocks:** There are three kinds of blocks, which are described now. Each block should have a unique color or fill to make it easily identifiable to the player. Each block traversed is one unit of movement. The layout of blocks should be read from a file specified in the

**Exploding:** If Char hits one of these Char's health is decreased by -10. Once a block has exploded it will not explode again. Exploding blocks should not be placed where Char needs to jump on it to move forward.

**Gas:** If Char hits a gas block he cannot shoot for 1 second or 10 movement units. Gas blocks should not be placed where Char needs to jump on it to move forward.

**Normal:** Char will walk on these. If a block is higher than the one next to it Char will need to jump to get on it.

**Scoring:** In the upper part of the screen Char's health should be displayed and updated as actions occur.

**Game setup:** When the game is started, the following initialization should occur.

The block file is read from a filename that is given on the command line. You can have a default file, as well. The “map” for the game is initialized using this information. If no block information is used the map will initialize to a flat surface that plays the same as all normal blocks laid out on a level playing surface.

Data to display Char, etc. are read. You can either use graphics files for these or simply letters or other characters. At least one Character, creature or magic item should be a graphic.

Creatures placement should be done as part of the game setup. This will allow games to be repeatable and aid in debugging.

The score and health information is initialized to the starting values.

Char is placed at the leftmost part of the screen.

Any other initialization you need to perform.

The game begins.