Basic Level Builder

# Introduction

The **Basic Level Builder** (BLB) is a level creation program for use in teaching the core game design concepts that are involved in level creation, such as intensity curves and guidance. The BLB consists of a simple tile-based video game and a carefully chosen set of tools that may be used to create levels for it.

# Goals

Students can use the BLB in learning and practicing *intensity curves*; *signifiers*, *feedback*, and *affordances*; *guidance*; *difficulty* and *fairness*; *playtesting* and *iteration*; and more. Any of these skills can be taught and practiced using other tools, such as Unity or the Zero Engine, but the restrictions of the BLB help to keep the student focused on the parameters of whatever their current assignment is, without being distracted by the boundless possibilities that a larger game editor has to offer.

# Details

The BLB is broken into two parts: the game and the editor.

## The Game

The game within the BLB is a very basic tile-based platformer. It has a small set of mechanics: the game spawns the hero at their starting point, and they must reach a goal object. They can move side to side, and they can jump. Gravity pulls them down while they are airborne. The hero interacts with some objects by contact, such as coins and checkpoints.

## The Editor

The editor is divided into the main **Level Window**, where the designer places the tiles that make up the world; the **Tool Palette**, where the designer selects which tool to use in editing the level; and the **Tile Palette**, where the designer chooses the tiles they will place in the world.

### Level Window

The Level Window is the student’s interface to the game’s world. The world of the game is logically divided into a tile grid, which is empty by default. The designer constructs the world by placing tiles into the grid’s positions. The window can be scrolled arbitrarily far in any direction, but there is a *kill height* that can be set by the designer — in the game, if the hero falls to this height, they “die”.

### Tool Palette

The tools of the Tool Palette are the means for the designer to edit and manipulate the level they are creating in the Level Window. The tools available to the designer are the **Brush**, the **Box**, the **Eyedropper**, and the **Selector**. Each tool may be selected either from the Tool Palette or via a unique keystroke.

#### Brush

The Brush is used to paint copies of the primary and secondary tiles into the world.

#### Box

The Box tool is used to create rectangles using the selected tiles. Dragging with the primary mouse button creates a rectangle from the primary tile, and dragging with the secondary button uses the secondary tile.

#### Eyedropper

The Eyedropper sets the primary or secondary tile by choosing it from the Level Window.

#### Selector

The Selector tool selects regions of tiles in the world. Selected regions may then be cut, copied, and pasted.

### Tile Palette

The tiles available to the designer to place in the world may be selected from the Tile Palette. There are always two actively chosen tiles at a time, the *primary tile* and the *secondary tile*, which may be placed in the world using the tools. Among the tiles is the “empty” tile, which is actually no tile at all, but it acts as an eraser: “placing” it removes whatever was previously in that position.

The tiles are:

* **Empty** — Empty space that the hero may freely move through.
* **Solid** — Solid tile. The hero can stand on it, land on it, bump into it, etc.
* **Slope Left** — Solid tile with a 45-degree slope going up and to the left.
* **Slope Right** — Solid tile with a 45-degree slope going up and to the right.
* **False Solid** — Empty space with the same outward appearance of a solid tile.
* **Invisible Solid** — Invisible solid tile. Behaves just like a solid tile, except that it cannot be seen.
* **Start** — Where the hero spawns when the game starts.
* **Goal** — What the hero must find in the level. Multiple goals may be placed in the same level.
* **Deadly** — Kills the hero on contact.
* **Checkpoint** — The hero returns to the most recently touched checkpoint when they die.
* **Coin** — Increases an arbitrary counter. May be used to psychologically manipulate the player.
* **Teleporter** — Coded. When touched, the hero moves to the next matching teleporter.
* **Booster** — When touched, gives the hero a sudden speed boost in the chosen direction.
* **Door** — Coded. Acts as a solid until it is opened.
* **Switch** — Coded. When touched, opens all matching doors.
* **Key** — Coded. When touched, enables the hero to open, by touch, any matching door.

#### Coded Tiles

Some tiles, such as the teleporter, are **coded**. Just after the designer places a coded tile, they are prompted to assign it a letter or number, which becomes that tile’s *code*. Tile codes are used to link tiles in logical groupings according to their type: teleporters warp the hero to other teleporters of the same code, while keys and switches open doors of the same code.

#### Tile Colors

For visual clarity, coded tiles are also given a color. Just after the designer assigns a coded tile its code, they are prompted to select a color for that tile. **Selecting a color for a coded tile automatically updates all other tiles of the same code to have that color.**

#### Boosters

A booster gives the hero a sudden speed boost in the chosen direction. Just after the designer places a booster, they are prompted to choose its boost direction.

### Other Editor Functions

The Editor allows the designer to **save** their level and **load** a level that has previously been saved, both manually and also by an **auto-save** feature. It also has a basic standard **undo**/**redo** system.

#### Saving and Loading

The BLB allows the designer to save and load their work. A level can be saved in place, or to a specified path. Additionally, the editor automatically saves the current level after every so many actions. The auto-saves are each saved with time-stamped file names, and they are all listed, together with the designer’s manual saves, in a separate pane in the Level Window. This list allows the designer to quickly view multiple versions of the current level, Any of these versions may be loaded at any time, but the designer may also load a specified level that has been saved elsewhere on disk.

#### Undoing and Redoing

The undo system in the BLB is very basic: the designer may undo any action, which may then be redone. Of course, a more advanced undo queue is ideal for a tool like the BLB, but with its robust history system, it isn’t necessary.