

Graphics Requirements

General Graphics

1. The game needs a background. This can literally be anything, be as creative as you like. It also doesn't have to be a static image. If you want laser snakes to be flying back and forth in the background, we can make that work.

The aspect ratio of the game is 1.5 to 1, and the default resolution is 960x640, so consider making the background about 1.5 times that size (we can always scale it down in game).

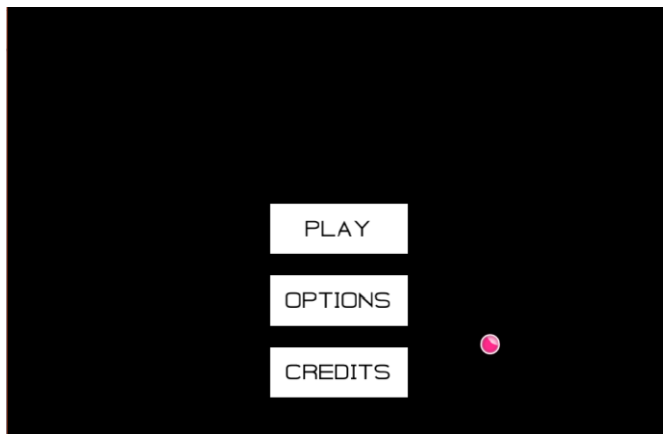
2. Every menu screen in the game uses the same button sprite, so you should create a generic design that all buttons can use. The button sprite is currently 200x70 at default resolution (960x640). The buttons must also have three states: unpressed, hovering (when the mouse is hovering over them), and pressed. If you want to make animations going from unpressed to hovering, hovering to pressed, and then pressed back to unpressed, go ahead.

Each button will have text displayed on it. You can either choose a font for the text to be displayed as (currently it's "Fairview"), or you can embed the words in the buttons.

3. There needs to be a mouse cursor sprite. Size doesn't really matter, though it should be maximum 60x60 at default resolution. Be creative, but pragmatic.

Note: The graphics I say you have to create are just the required graphics. If you have a design for the main menu that uses more graphics than what I describe as required, add them in. The more the merrier.

Main Menu

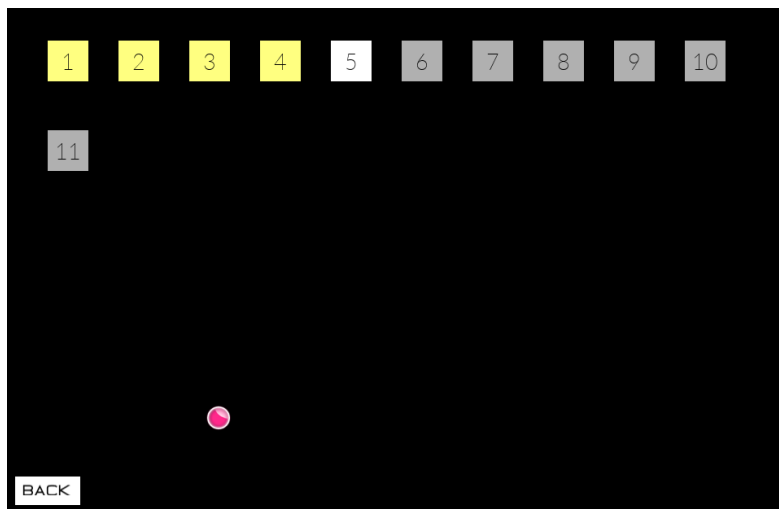


(The current main menu screen)

This is pretty open. It needs a background (the background from the General Graphics section), and the buttons from above. At one point or another a title thing should be added, but the game doesn't have a name right now, so that area should be left empty for now. The only requirements of the main menu are that it has the three buttons, is easy to use/navigate, and not cluttered. Be creative.

This task requires you to design the menu screen such that it satisfies the requirements, and then create the graphics so that I can recreate the design in the code. Don't just make a single image of what the main menu should look like, make the individual graphical components.

Level Select



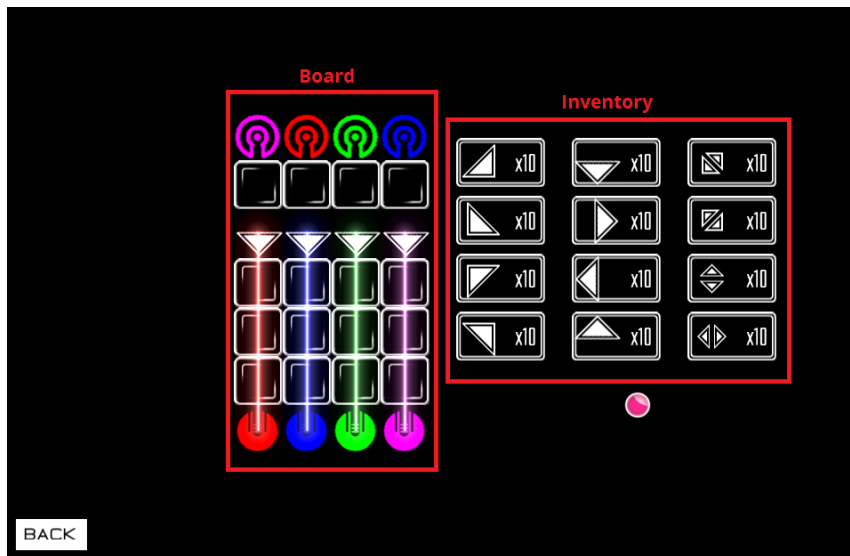
(The current level select screen).

There should be three different level select buttons; completed levels, unlocked levels, and locked levels. Right now, levels 1 to 4 are completed, 5 is unlocked, and 6 to 11 are locked. The buttons should be 50x50 maximum at the default resolution. Also, they should have a “hovering” and “pressed” state just like the general buttons.

There must also be a back button at the bottom of the screen somewhere, which can just be one of the general buttons resized. Also, when there are more levels than can fit on screen, a scroll bar appears. The scroll bar shaft should be 12 pixels in width and variable height (i.e. it should be tile-able. If you need clarification, ask me).

You can do cool stuff with this as well. If you want the buttons to randomly light up and die back down at different intervals, that can be arranged.

Level Screen



(The current level screen with the “board” and the “inventory” labelled.)

This is the big one. First, you need to make the empty tile sprites (the squares where the user can place refractors). These must be 55x55. Then, you have to make the inventory buttons, which should be about 100x55. If you have a good design with different dimensions, run them by me.

Though it isn’t displayed right now, you have to create a graphic for an “inventory cursor” of sorts, that lets the user know what item they have currently selected. This could be, for example, brackets that move back and forth enclosing the selected tile.

The most important components however are the tiles themselves (the refractors, the “outputters” at the bottom, and the “receivers” at the top). Every tile must be able to fit inside both an empty tile sprite and an inventory button sprite (as is shown in the image), meaning they should probably be about 50x50 or smaller.

The sprites for the refractors should represent their functions. As it stands, they look very unintuitive (as you probably remember from your test plays). For example, the refractor in the top left corner of the inventory (the triangle whose right angle is on the bottom-right corner) takes light from the bottom or the left side and outputs light moving diagonally up-left. It’s sprite should represent that somehow. I’ll leave it in your hands to do this in a creative way. If you need clarification on what each tile does, ask me.

The “outputters” are tiles that output a laser. The sprite should indicate both the direction from which the laser is outputted, and the colour of the laser (again, how you do that is up to you). I suggest making a single outputter sprite pointing upwards, as the other directions are simply rotations of the upwards sprite, which are done in-game.

The “receivers” simply receive laser input from a certain direction. They require a specific colour of laser to enter them. When the right colour laser enters from the correct direction, it should light up (you don’t have to create a “lighting up” animation, I’ll do it in code). The sprite should represent these functionalities. As with the outputter sprite, just make a single one pointing upwards.

Synopsis

- Background (approx. 1440x 960)
- Menu Button (unpressed, hovering, and pressed states) (approx. 200x70)
- Cursor (max. 60x60)
- Optional Main Menu Graphics
- Level Select Button (unpressed, hovering, and pressed states) (approx. 50x50)
- Scroll Bar and Scroll Bar Thumb (approx. 12 wide and variable height)
 - (the thumb is the part you grab to scroll).
- Optional Level Select Graphics
- Empty Tile (approx. 55x55)
- Inventory Button (approx. 55x100)
- Inventory Button “Cursor”
- Refractors
 - UxL_UL, UxR_UR, DxL_DL, DxR_DR
 - ULxUR_U, DLxDR_D, ULxDL_L, URxDR_R
 - pass_UL, pass_UR, pass_DL, pass_DR
- Outputter
- Receiver
- Optional Level Graphics

Refractor Notation

The UxL_UL stuff above is shorthand for the different refractors. UxL_UL is the refractor that takes Up and Left as input and outputs Up-Left. Similarly, URxDR_R is the refractor that takes Up-Right and Down-Right as input and outputs Right.

The pass_UL refers to the refractors that allow lasers to pass through them at specific directions. They’re the refractors you see in the right-most column of the inventory. For example, pass_UL is the refractor that allows a laser going Up-Left to pass through it (the refractor in the top right corner of the inventory).

If you need clarification, just ask.