# Planning

**08 January** =

* **Items =>**
  + General code for shooting a laser beam. (With option to make its max length exactly **1 square**.)
  + **Add new items I invented** (+tutorial, +test of course)
  + Add the “beam” option to other items that might need it?
* **Rewrite gate calculation code** (to be way cleaner and better => write single functions that show my *intent*)
  + Save more data inside the Edge object itself?
  + **IDEA:** when placing the new room, immediately open any edge with access to (half)open tile. Save this as “must\_open” or something, so it’s never closed.
    - Then, in “finish\_in\_hindsight”, we turn any other edge with access to (half)open into gates/borders, *except the must\_open*.
    - **THE PROBLEM IS,** that I just don’t have any certainty about what the bitmask/room layout is doing. Even the old edges, connecting to previous room, might change at some point.
  + Only create edges *where needed*, instead of doing the whole room? (Nah, might make the “soft open” harder to do.)
  + Also add **certainty of connection** code? Check the tiles on the *“new side”* of the room, make sure there’s an opening?
  + Just don’t give teleporters a border?
* **General stuff**
  + Also take probability into account when randomly *selecting* stuff.
    - While drawing the original list, save a final probability in “final\_prob”? Or is that a bad idea?
    - Meh, instead repurpose the “draw\_list\_weighted” for the other random draws as well
  + **Slightly stronger gravity, and weaker jump.** (Would lead to better friction with the floor, and less bumping your head against the ceiling. Check if rolling force needs an adjustment as well then.)
  + “IS EVERYTHING TAUGHT?!” Not working, so it seems.
    - Write an easy way to test *extremely short levels*.
  + **“BUTTON LOCK” =>** count is sometimes off (sometimes *way off*). Step 1: forbid erasing buttons, even if they float. Step 2: figure out if something else is wrong during button spawning. (Maybe they just assume placement was fine, even though it might not be?)
    - Also, *pressing* the button isn’t as precise as I’d like …
    - **IDEA: Simply set the “num\_buttons\_to\_place” to the size of special\_elements array?** (So update any time one is removed? AAAAARGH NO WON’T WORK, as then it would also do that for real removals
  + Teleporters => make less likely on tutorial course + more robust implementation?
  + Test on controller.

**09 January** =

* **Soundtrack**
* **BG music trailer**
* **Marketing items (favicon, header, etc.)**
* Go through all TO DO/DEBUGGING and fix them, create proper release build for project.
* Write all other ideas/bugs/whatever as a future to do

**?? =>** playtest, cut to trailer/GIFs/screenshots, completely finish

# To Do

## Special Tiles

**Multiple bodies?**

* A tile that gives you as many coins as you have bodies.
* A tile that freezes bodies nearby *if they aren’t your worst body*.
  + This might seem an obstacle, but it’s actually a helpful thing.
  + It locks a body, so you can safely move your others without worrying about it.
* A tile that changes *all your bodies* to a triangle.
  + The one tile that **does** have its effect on **all** your bodies

**Coins?** (So these act as a sort of “shop”.)

* Pay X coins to destroy all your other bodies.
* Pay X coins to *blast away* or *slice* all nearby bodies (excluding your own, of course) (solo\_unpickable)
* Pay X coins for a huge time bonus (solo\_unpickable)
* Pay X coins to *shrink* (+ *make triangle)* everyone around you (solo\_unpickable)
* Pay X coins to slow down the thing chasing you (multi\_unpickable, high prob?)

**Growing/Shrinking**

* A tile that grows you to max size
* A tile that shrinks you to min size

The *shrinking* one should be way more likely in Solo Mode.

**Platforming?**

* A magnet that attracts/repels all bodies in a radius (ignoring walls)
* A platform placed somewhere in the room that simply turns on/off on a timer. (Acting like a door, or gate, or moving floor, depends on situation.)
* A similar platform that *moves* or has a *hole* in it.
* An item that just has a *slight* slope on it, causing you to roll/fly off, without blocking too much.
* ?? Freeze Beam: everything inside just slows down a lot (area physics override gravity and damping?)

### Bugs

**BUG:** Still too many teleporters placed, at least in the tutorial course.

* Also a crash (infinite loop?) when using teleporter, with multiple players, and multiple bodies being left out.

**REMOVE:** The spiderman item => really adds nothing useful.

**REMOVE/MODIFY:** The invincibility item? What does *that* do?

=> Maybe these items should *all* cast a beam, like the laser. (If not a beam, then at least exactly one *rectangle* above it.)

=> This can then be re-used in *many* items, if I make it general

## Fix Outline/Gates

**BUG:** Still not great when calculating which sides must be *open* (so players can come in), without accidentally opening *gates/other side* as well.

**IN FACT,** completely rewrite the code for outlines, as it’s just a mess now. (I think “open\_connection\_to\_previous\_room” does nothing.)

* Determine outline (the literal) edges at the start, like now, but nothing else.
* Create\_border\_around\_us should do just that: create all edges. *Later* we can chip away.
* Only once placement is completely done, read the area to see what we want to chip away.
  + Get position at other side. Is it inside the *previous room*? Open. Is it inside the *next room?* Keep closed.

## Re-Test

* Glueing
* Reverse Controls
* No-Holes

## General Gameplay

* Don’t forget to turn on the **fog** again + test if it’s even an addition anymore

# Future To Do

**Simple Generation mode (used in tutorial course):** Add more control over first few rooms. (Keep flat, ensure a jump is needed, etcetera.)

**Better labels** => somehow, ensure they are placed *within the room* and without *overlapping important stuff*.

**Magnet particle/visual =>** create white circle moving inward?

**Shape change particle/visual?**

~~When you are a non-standard shape, make the timers (for rounding/malforming) a bit~~ *~~longer~~*~~, so you stay in that form for a while.~~ => probably a bad idea, as it’s hard enough as it is, and it would just frustrate players with no way out

**BUG/IMPROVEMENT:** Sometimes there are still really tiny gaps to move through, otherwise you just can’t progress => It’d be best to eliminate them *for sure*

More items specifically useful in *solo mode*?

**BUG:** That bug with the entities array containing a “previously freed instance”. Theories:

* Happens after “stood still too long” teleport?
* Happens when sliced at some odd moment?

## Map Improvements

**FILL ROOM Algorithm:** Add a variation where we’re allowed to place tiles *against the walls*, but *not in the center*. (By default, we only place away from walls, in the center.)

**Problem?** Should find a way to ensure that connections to other rooms stay open.

(Before placing, check if this tile connects to a different room. If so, don’t allow it.)

(When placing a new room, also recheck the tiles on the previous room?)

**3-way-open-tiles removal:** Create option to *replace* them with something instead? (Now there’s a chance of breaking walls and ruining the generation.)

**Tutorial generation:**

* Randomly remove slopes at the *top* (or bottom, for that matter) to allow for bigger and cleaner openings => especially in **simple generation**/tutorial

Find **“annoying patterns”** and fix them in varied ways.

**Special Tiles/Items:**

Think of more fun stuff that plays with **blocking players doing really well + physics variation**

* Create platforming sections (with ramps, moving platforms, etcetera)
* Create other inner gates. (Like the laser/cannon.)
* **IDEA:** Some beam/magnet that attracts *across the whole column*. (Shoots across the room like the laser.)
* **IDEA (related):** A fan that blows ( = repels) *across the whole column*.

**Lock:** **Shop**. (If I figure out a good way to do it.)

**Add *special tiles* that do something with coins!** (And once we’ve done that, add an area to bodies, which lights up the coin interface when such a “coin\_related” tile is in range.)

**INPUT:** Add different control scheme for controllers: joystick to roll left/right, any button to jump/float.

* (Make this default? Or can players configure it themselves?)

**GENERATION**

* Play with generation parameters => I feel big rooms should be *slightly* less filled (or have more varied filling), maps should *flow* a bit more (with slopes, rooms that are not *too* different in size/displacement)
  + “Preferred” displacement would be something that does NOT create a bump in the line. So either it stays flat at the ceiling, or it stays flat/falls down on the ground.

Don’t put a **paint lock** on a *huge ass room*.

# Trailer

Parody of “Rolling in the Deep” (by Adele).

## Footage

**Record footage from playtest.** Accompany background with footage that complements what’s being sung.

* Under 90 seconds
* Start with one attention-grabbing, supercool intro. (During intro of song, with just the drum. Or maybe even before the song starts.)
  + Start with your best joke, end with your strongest material.
* Then go slower, and build it up again, until the final climax.
* Use (animated) text and titles that fit the game.
* **Make sure they can easily be cut into cool GIFs => spread those as much as possible.**
* **Record footage *without* BG, so sound FX come in the trailer.**
* Make first 2 lines of YouTube description the best => those will appear in search results

Game trailer templates

* **Tell, Show, Repeat:** use title cards/narration to tell a thing … then show the thing. Repeat. It’s best to call out what you *do* in the game or what your *goals* are. (Don’t call out *raw features*.)
* **Music Video Montage:** Take fun gameplay and cut it to the beat of the music. Best used for games which are *very simple to understand from just watching*. Can also sequence gameplay so the *simplest shots* are at the beginning and you slowly build.
* **Chronological Order:** record footage of gameplay, then simply keep them in that order but cut them in some exciting way. Many games are structured like that: introduce ideas, then explore some twists on the idea, then test player’s ability to understand them. So keep that sequence to entice the player.
* **Just Explain The Game:** some games are really hard to understand at a glance. So just use text/title cards/narration to explain the game in a linear fashion. (Then use good music and editing to make that exciting.) Mostly works for games with visuals that are hard to parse/understand in a fast-cut trailer.
* **In A World …:** story trailer. Say in this order: 1) this is the world/premise; 2) this is the *person* in that world; 3) this is the *problem* that person faces; 4) this is how they *confront* the problem; 5) these are the *obstacles*

Make sure footage is **clear:** no HUD or UI, or just compose shots to have a clear focal point (or limited number of things on-screen).

Retention of knowledge/information is far more important than quantity of content.

## Lyrics

### Verse

There’s a strange folk

They are called the Sheepe

Weird shapes and biting wolves

They roll and jump and leap

There’s a game

For 1-6 players

Roll to the finish first

Before your sheep gets hurt

### Pre-Chorus

The world is always

Different every game

To keep you thinking:

Boy, when have we seen it all?

You only need to know

Two buttons total

No time for blinking

### Chorus

You could just win it aaaaaaaallll

Rolling in the Sheeeeeepe

Enjoy the game with aaaaalll of your heart

Now go play, go play it, it is cheap

**Not mentioned in lyrics =** local multiplayer

## Discarded Lyrics

There’s a game

Two buttons to learn

Roll left and roll right

But don’t take the wrong turn

# Done

## Annoyances

**ANNOYANCE:** When you jump with your head against the ceiling, your *rotating* movement actually pushes you in the wrong direction. Which is just … annoying? (Yes, you can learn it, and use it for stuff, but … not great.)

* Solution #0: Make ceilings frictionless => can’t do it, as they’re part of the tilemap, which has *one* physics material.
* Solution #1: Always cling to ceilings => possible (check if cling vector is opposite to gravity vector)
* Solution #2: Make jumping less powerful
* Solution #3: *Hold* both buttons to *float* or *steady yourself*. (So when you hold both, your Y-velocity becomes 0. But your X-velocity continues.)

## Basic Bodies

**Step 1:** Generate a random polygon

* <https://stackoverflow.com/questions/8997099/algorithm-to-generate-random-2d-polygon> => basically, create a circle, but allow each point to vary in radius/angle
* <https://stackoverflow.com/questions/59287928/algorithm-to-create-a-polygon-from-points> => draw a point cloud first, order by angle, then draw through it

**Step 2:** Calculate its centroid. Place a smiley face there. Then center the polygon around it.

**Step 3:** Turn it into a physics body + draw it each frame.

**Step 4:** When given input, roll in a certain direction. (Check if this actually works for movement.)

## Body slicing

**Step 1:** Write the slicing algorithm I scribbled on paper.

* <https://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-intersect> => detect intersection point of two lines
* The rest of the algorithm is just:
  + Loop through shape.
  + Detect first intersection point. Add it to the shape. (Between the start/end vertices of the edge it intersects.)
  + Continue until second intersection point. Add it to the shape.
  + Now *extract* the part between the two points: shape 2. *Remove* the part you extracted from the original shape: shape 1.
  + Now recreate the *bodies* + *draw/move scripts* for each.

**Step 2:** Allow testing by drawing with the mouse. (Or clicking twice. Or pressing a key and testing a predefined line.)

**Step 3:** If successful, allow applying dynamically.

# Discarded

The old idea with “placing precreated rooms”

## Rooms & Routes

**Issue 1:** How do we allow *rotating* rooms?

* Translate everything to anchor center
* Rotate the thing
* Translate everything back => DOESN’T WORK, because the “position” property is still local, so translating back would just *follow the new orientation*
* Now recalculate opening values

**Issue 2:** What if a single side has *multiple* openings?

* We should be able to match any of them
* But *not* necessarily close the others when filling gaps

**Issue 3:** Now we have ugly *double walls* between rooms.