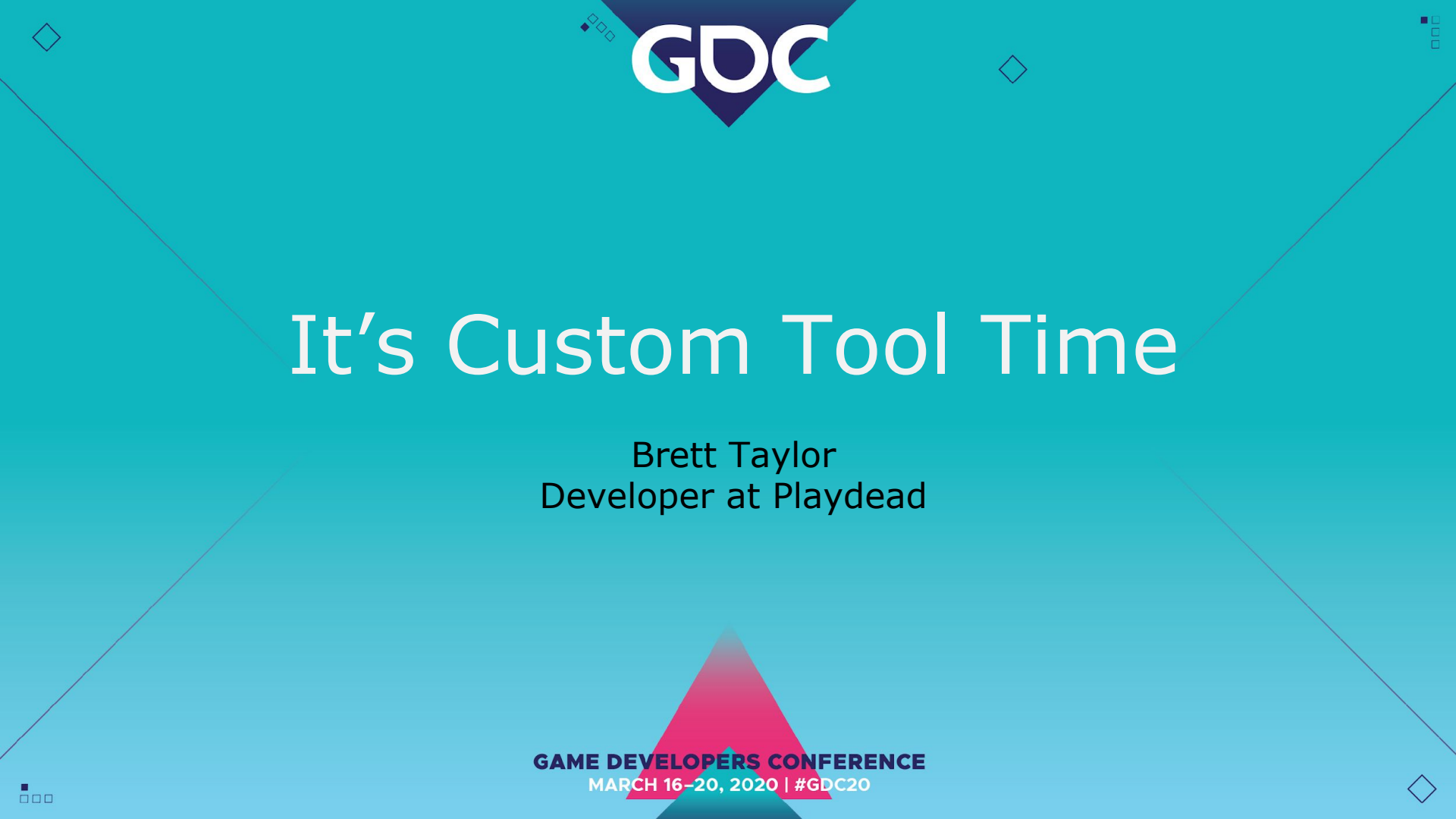

GDC
It’s Custom Tool Time
Brett Taylor
Developer at Playdead
GAME DEVELOPERS CONFERENCE
MARCH 16–20, 2020 | #GDC20

Me

- Brett Taylor
- Masters in Cognitive Studies from Columbia
- Developing games since 2008

Developer
2012 - 2015

Indie!
2015 - 2019

Designer/Developer
2019 - present

Overview

1. What is a Custom Tool?
2. Level Editors
3. When to Make a Tool
4. Unity Code Shortcuts

1. What is a Custom Tool?

What is a Custom Tool?

Code or system that saves time
for designer and/or programmer.

What is a Custom Tool?

- I use “tools” loosely
- This talk is really about saving time and energy
- This talk is about *design* tools (not production tools, like for bug-finding and profiling)

Staying “*In the Groove*”

- What about saving 20 seconds 12x a day? Worth it?
- Not just losing 4 minutes. Losing *groove*.
- Tools don’t just save time-- they keep *the groove*!
- Groove:
  - Creating, uninterrupted
  - Very productive
  - Easy to interrupt!

designer →

← obstacles that waste time

Bob Ross: A Time-Saving Role Model

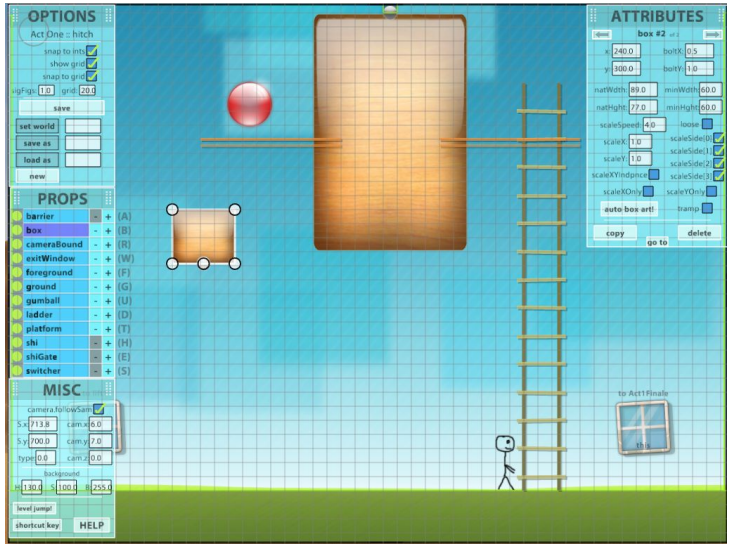
- Bob Ross LOVES shortcuts
- “This is the lazy man’s way of painting.” -Bob
- FALSE. This is the Smart Person way of painting.

small

BIG

```
MathUtils.RandomBool();
```

reusable code snippet



robust level editor

Designer Tools

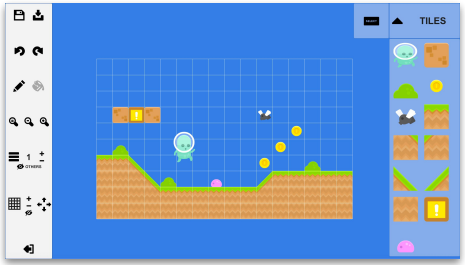
A system to add content

this talk

Programmer Tools

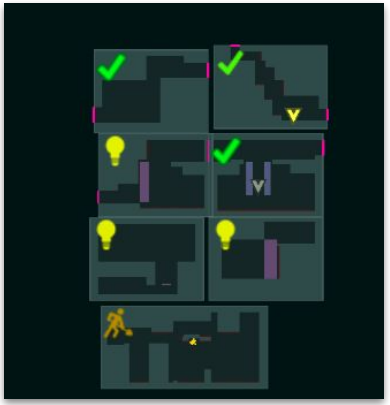
Reusable code snippet or library

Designer Tool Examples

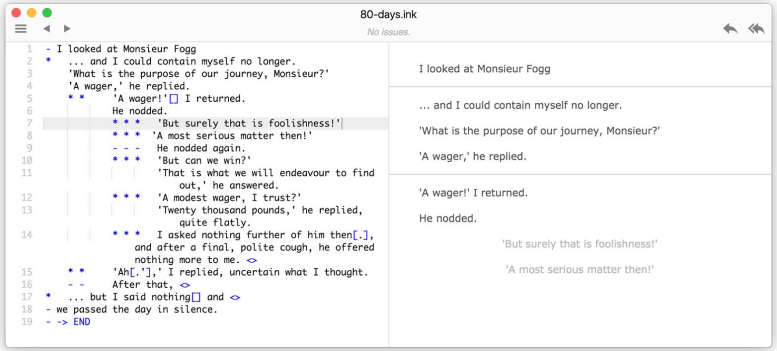


Level editor

Highlight content issues



Organizational visuals



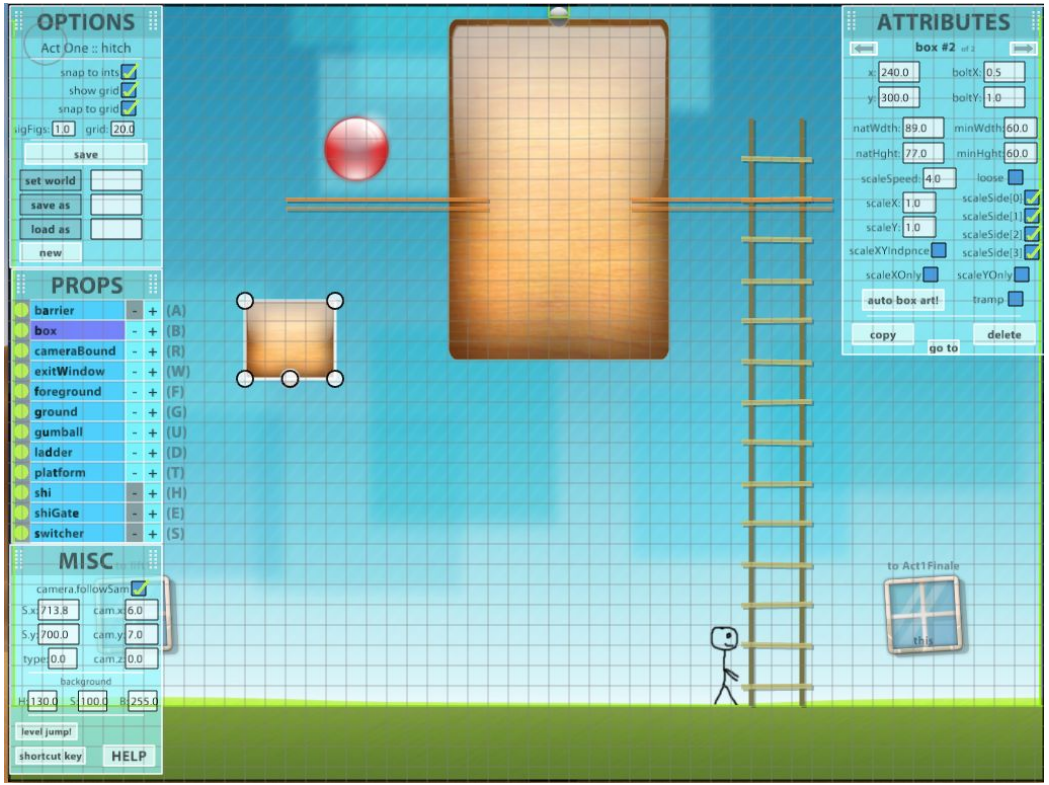
Text editor for branching narrative

2. Level Editor Examples

Scalea

- My first level editor!
- Made in college

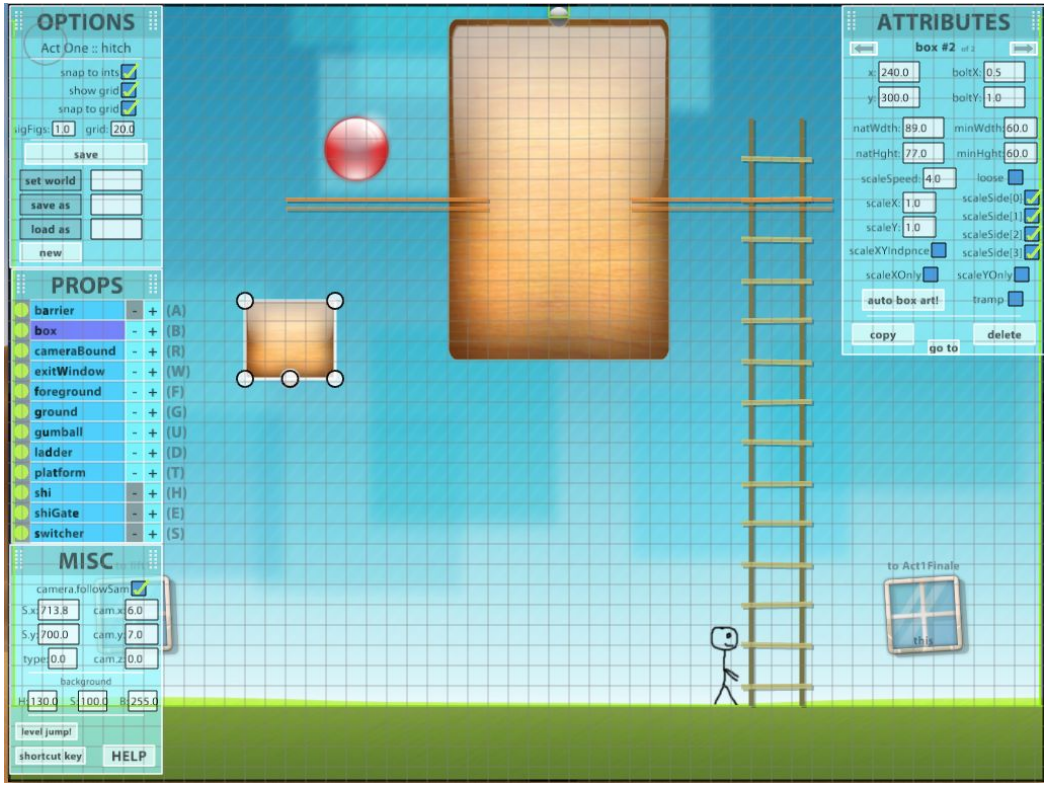
Braid's editor, my inspiration



My editor

Scalea

- WAY over the top (game was only 30 min long!)

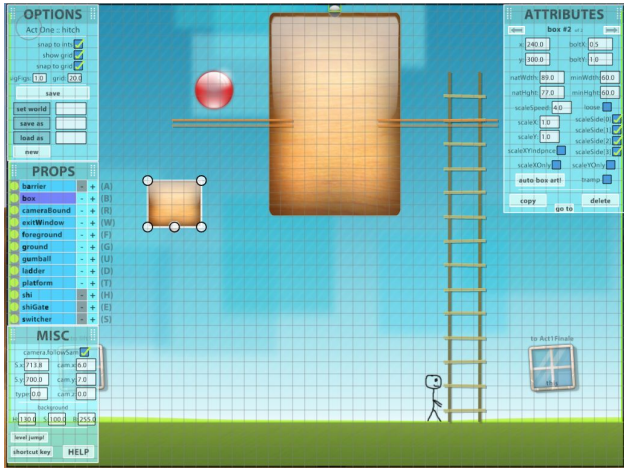


All the functionality. Too much!

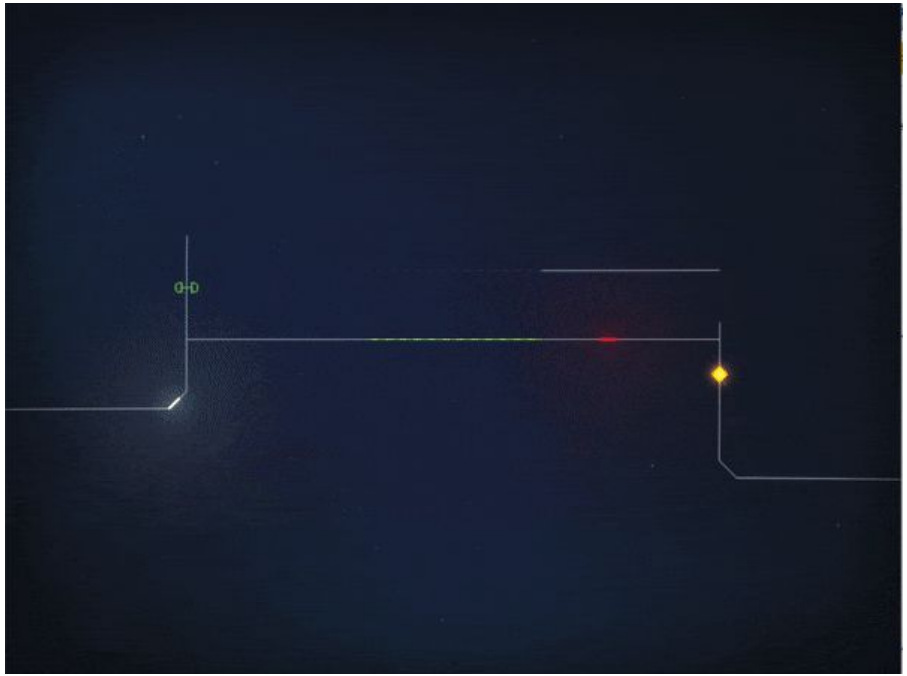
Scalea Editor: Outcome

COST	=	~3 months
PAYOFF	=	not much :/

though I did learn a LOT!



Linelight Level Editor

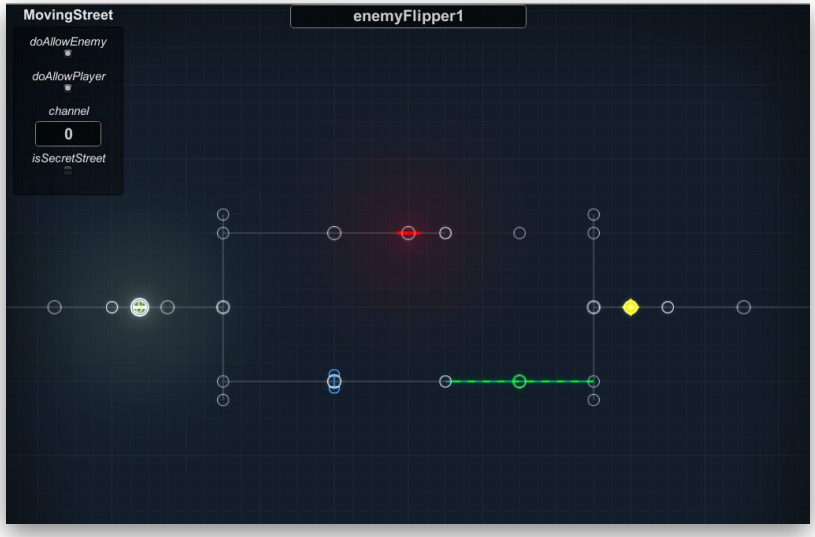


Linelight Level Editor

Made editor AFTER hardcoding enough levels to decide game was fun

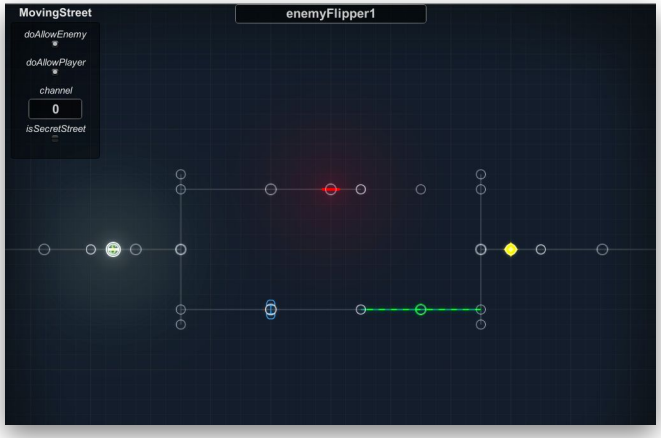
```
  // Intro to Moving Streets
  else if (levelKey == L_introMovingStreets) {
    float y = 70;
    l.addStreet(-400,y, 0,y);
    l.addStreet(-400,y, -200,y-200);
    l.addStreet(-200,y-200, -200,y);
    l.addMovingStreet(0,y-100, 200,y-100,  0,100,  0);
    l.addStreet(200,y, 400,y);

    l.addStreetFlipper(0,0.25, 0);
    l.addStar(4, 0.5);

    l.addStreet(-200,y, -200,300);
//    l.addStreet(400,0, -200,300);

    l.playerStartingStreetIndex = 0;
  }
```

eventually



Linelight Level Editor: Outcome

COST	=	**~1 month**
PAYOFF	=	**tremendous!**

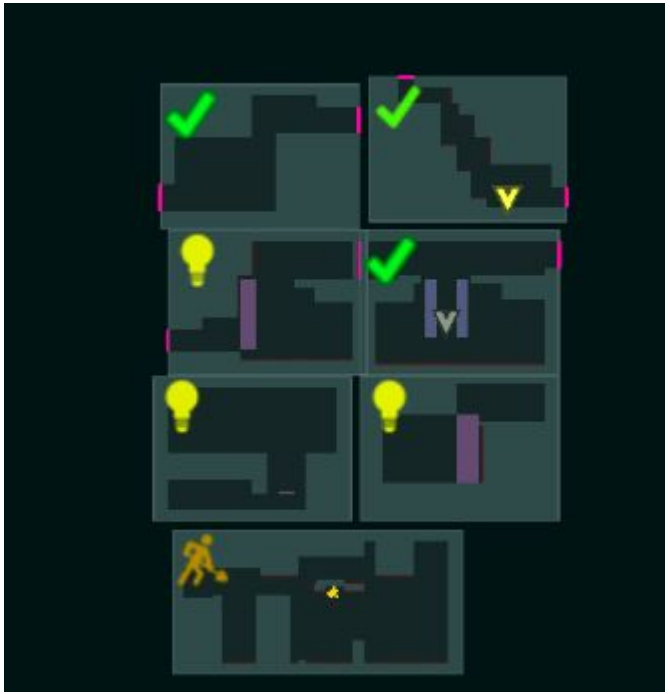


Linelight Map Editor

- Many, many features
- Added *gradually*

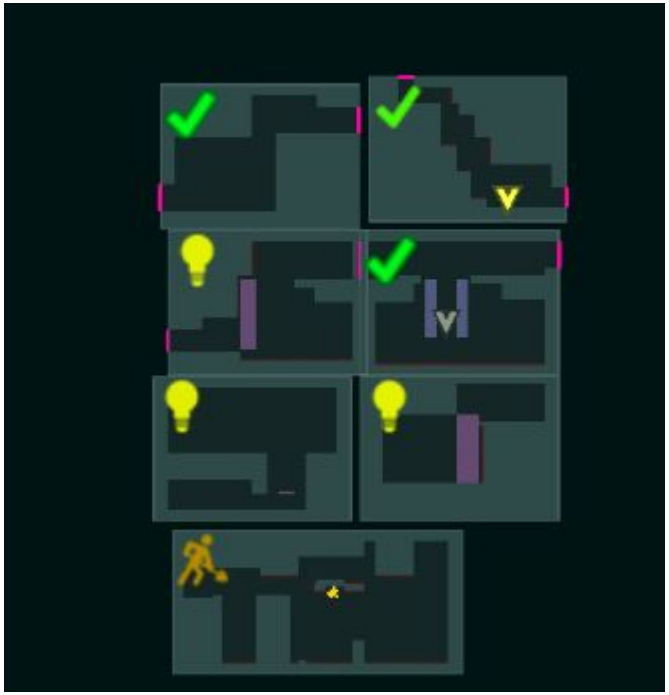
Designer Flags

- Each level has flag
- Tap to change
- Baked into editor (e.g. not in spreadsheet)
- I **love** this system
- Great for rapid editing, when levels in flux



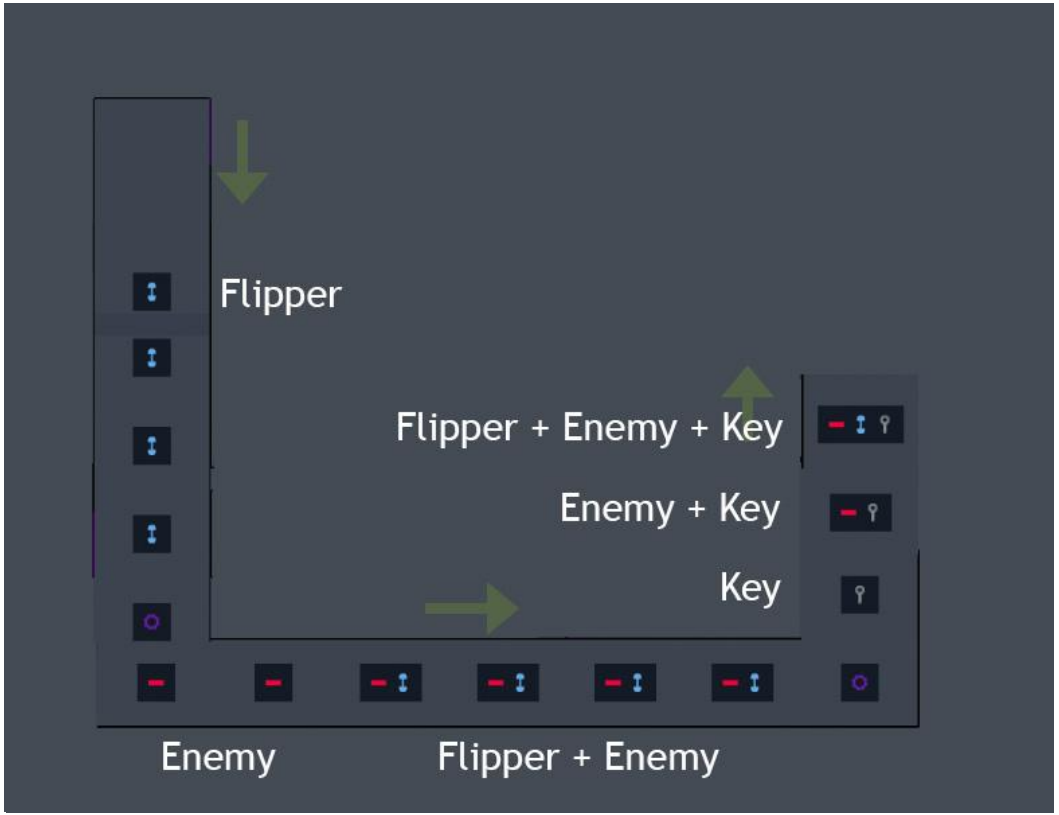
Designer Flags: Outcome

COST	=	**5 hours**
PAYOFF	=	**outstanding!!**



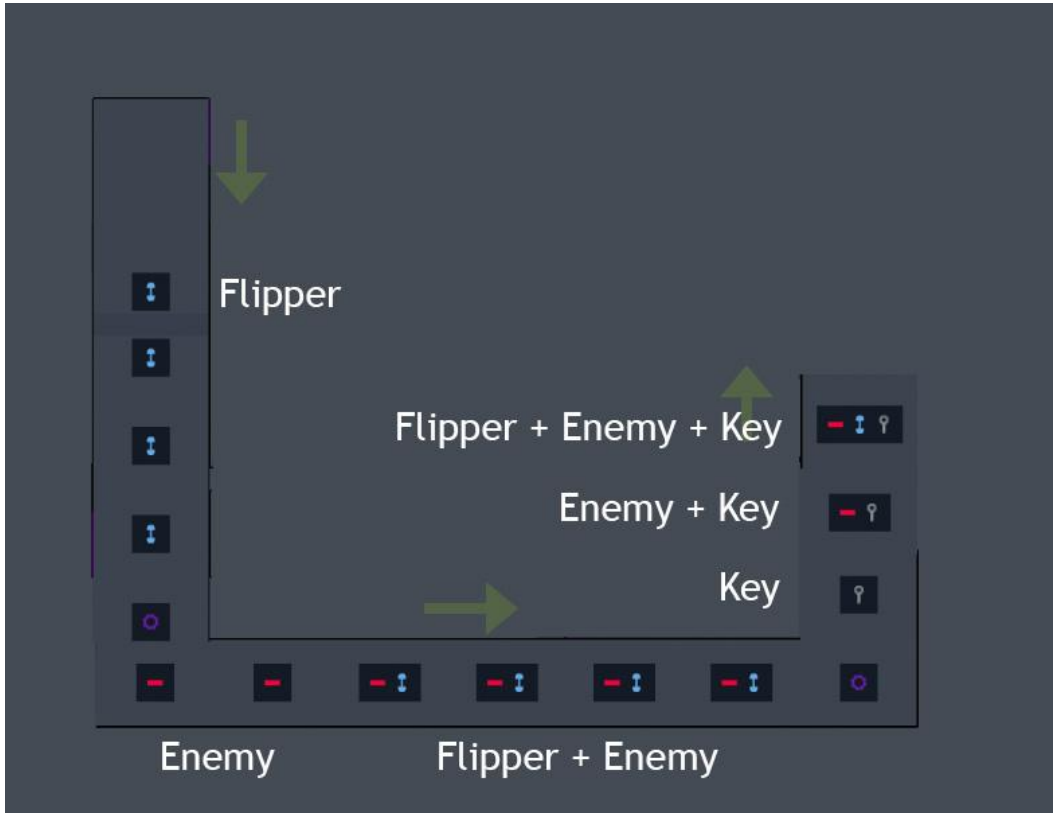
Prop Display

- Shows what's in what room
- For visualizing linear progression!



Prop Display

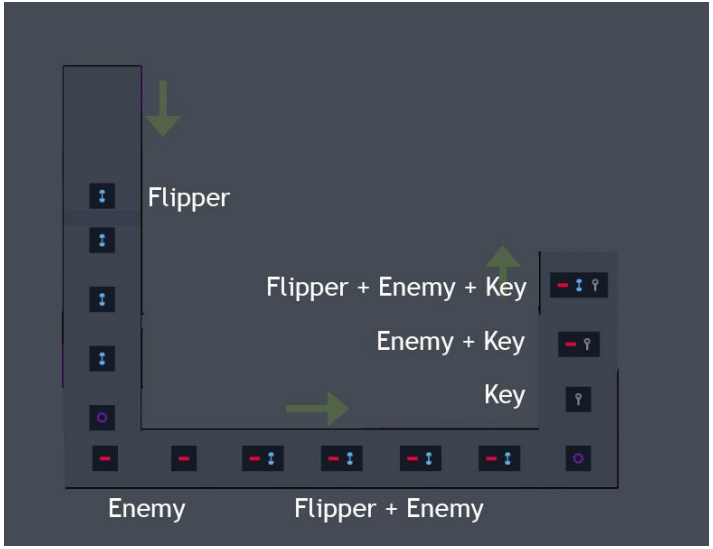
- Shows what's in what room
- For visualizing linear progression!
- Ultimately, I rarely used it.



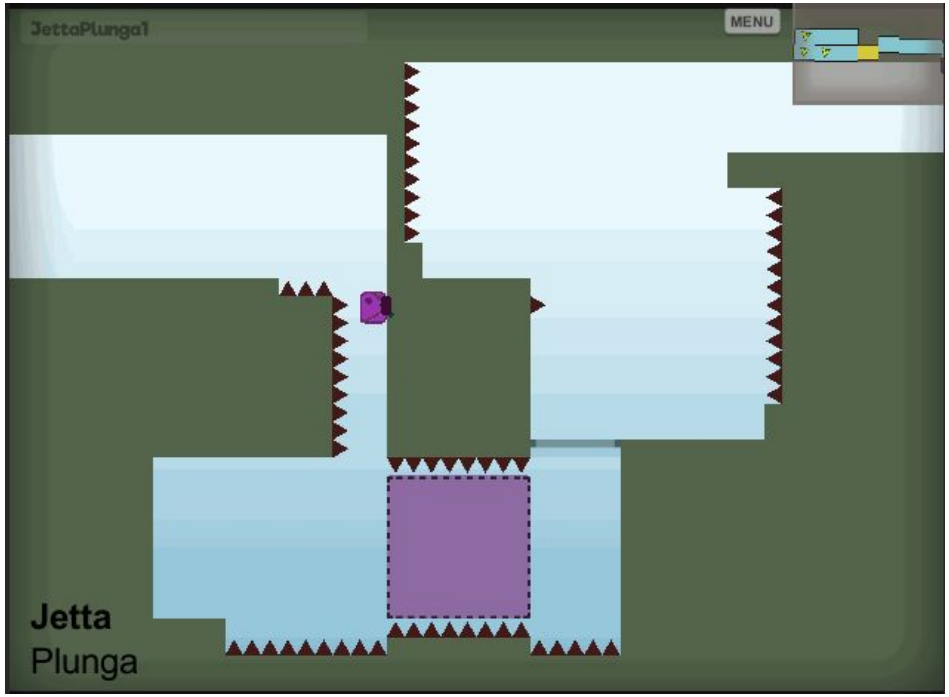
Prop Display: Outcome

COST = **3.5 hours**

PAYOFF = **none, really :P**

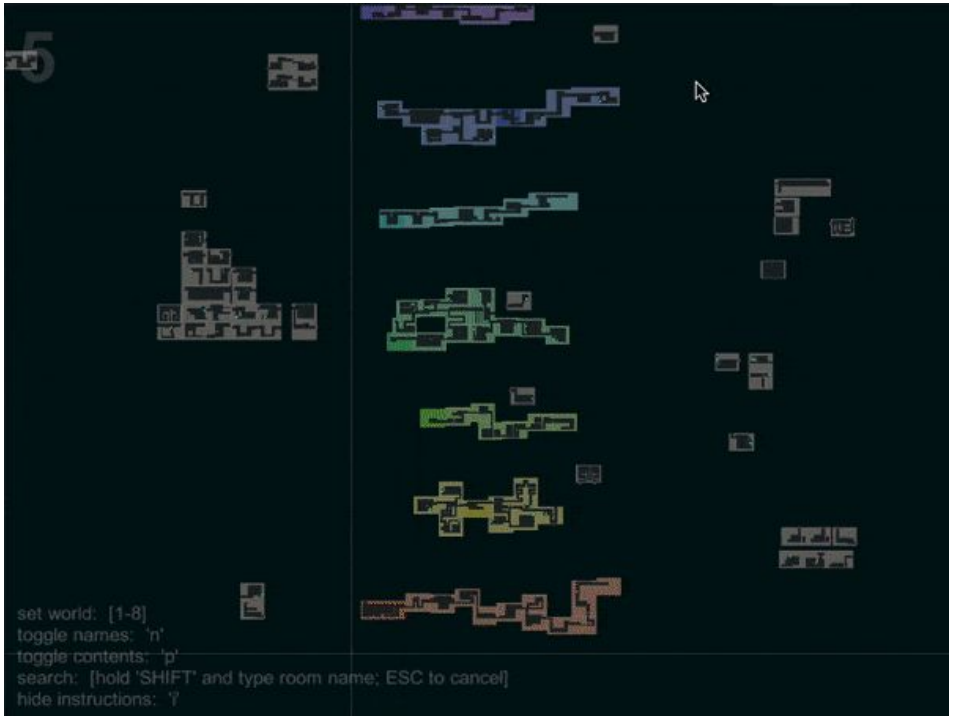


Bouncemeister



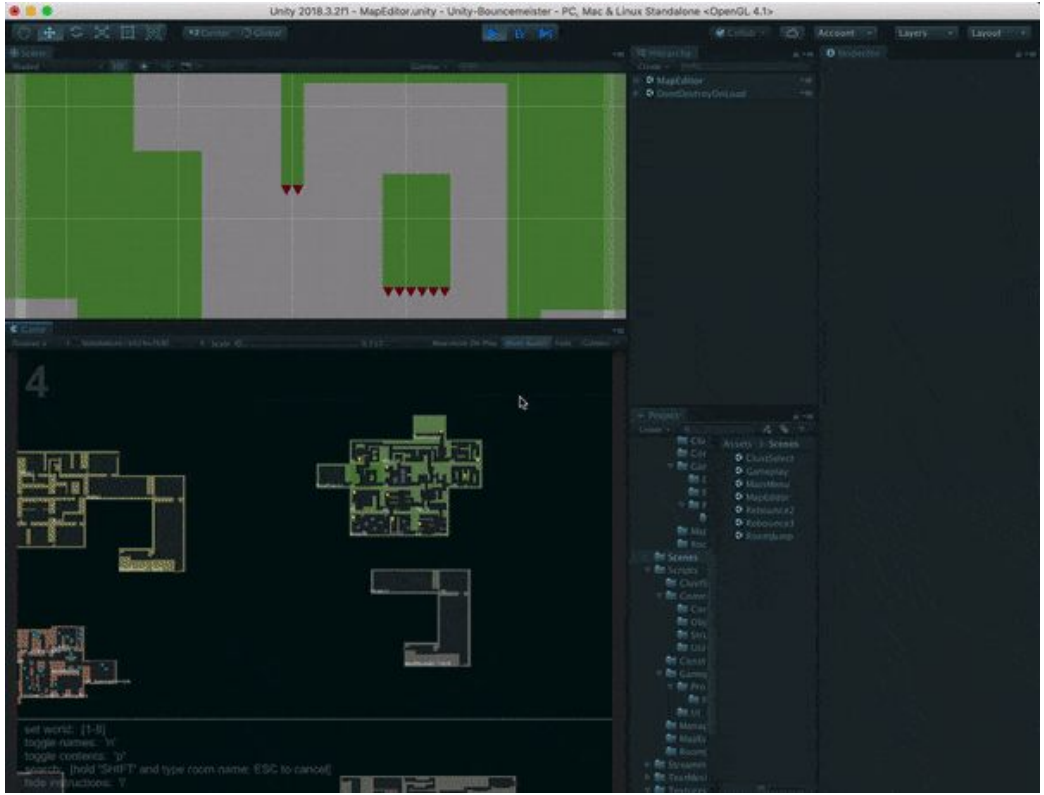
Bouncemeister Map Editor

- Rooms automagically connect
- Layout issues are highlighted
- Shift+[type] = search by name
- Shift+Click = select all connected Rooms
- *...and more!*



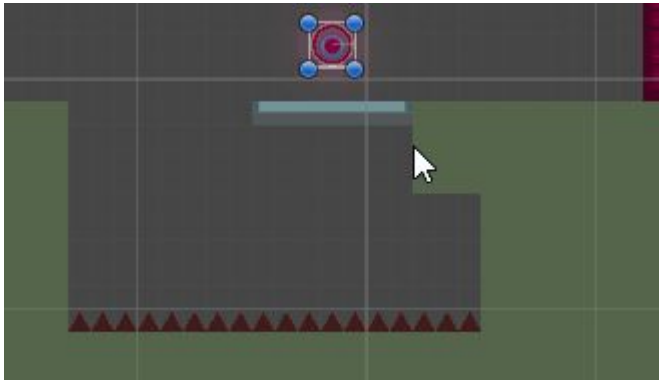
Bouncemeister Room Editor

- Uses Unity Editor
- Advantages
  - Already exists!
  - Undo/Redo
  - Multi-select/-move/-scale
- Disadvantages
  - Separate window to play vs. edit game
  - More windows = everything's smaller
  - Have to invent workarounds



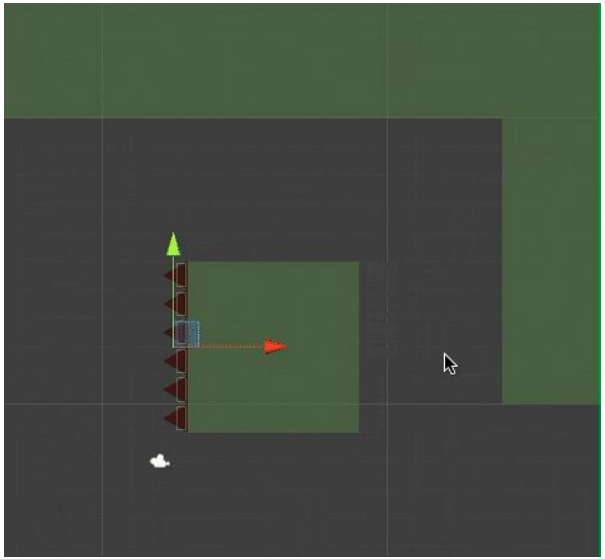
Bouncemeister Room Editor

- **Snap to grid**



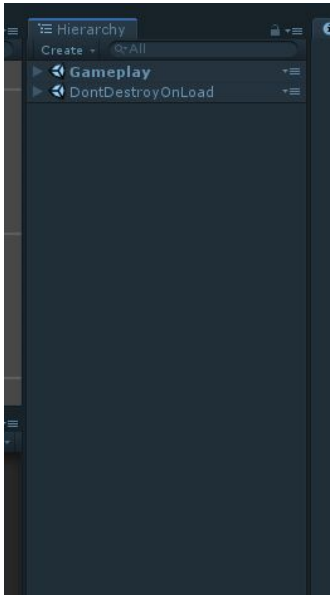
Bouncemeister Room Editor

- Snap to grid
- **Auto-rotate Spikes**



Bouncemeister Room Editor

- Snap to grid
- Auto-rotate Spikes
- **Auto-expand Hierarchy**

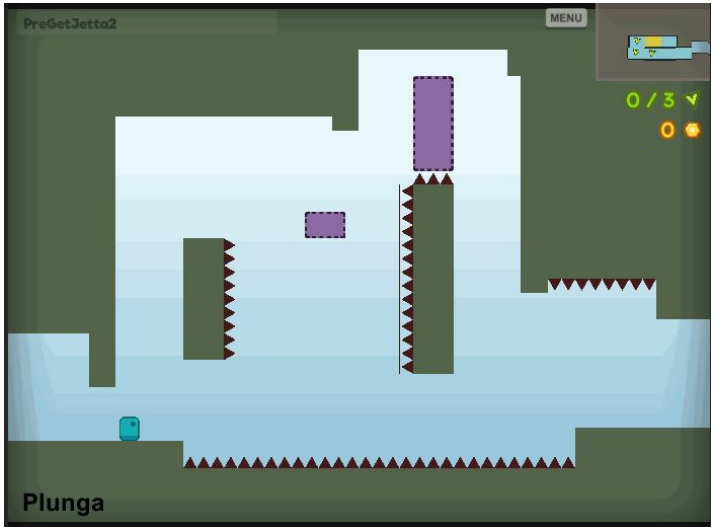


7 seconds
3 clicks

0 seconds!
0 clicks!

Bouncemeister Room Editor

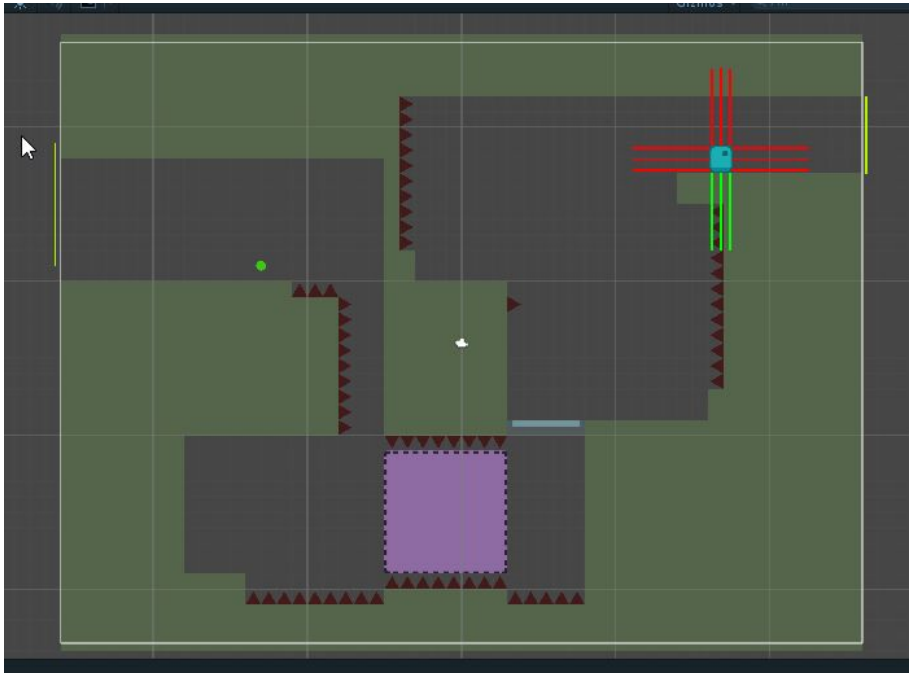
- Snap to grid
- Auto-rotate Spikes
- Auto-expand Hierarchy
- **Keyboard shortcuts**



room-jumping

Bouncemeister Room Editor

- Snap to grid
- Auto-rotate Spikes
- Auto-expand Hierarchy
- Keyboard shortcuts
- **Gizmos shows mismatched Room-connections**



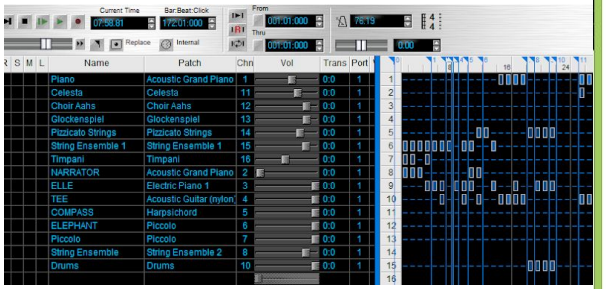
Music Story

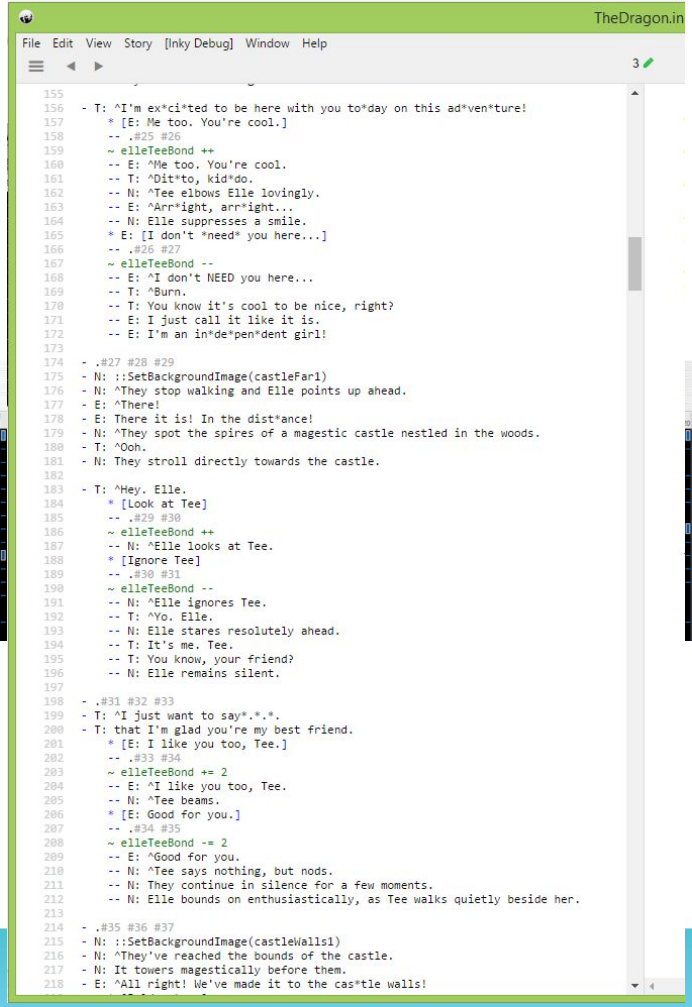
- Branching, musical narrative game
- All text set to music

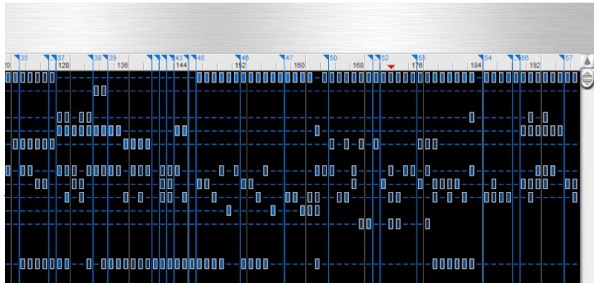
Music Story

- Wanted frictionless system to compose and connect branching story+music
- ^ Hard.

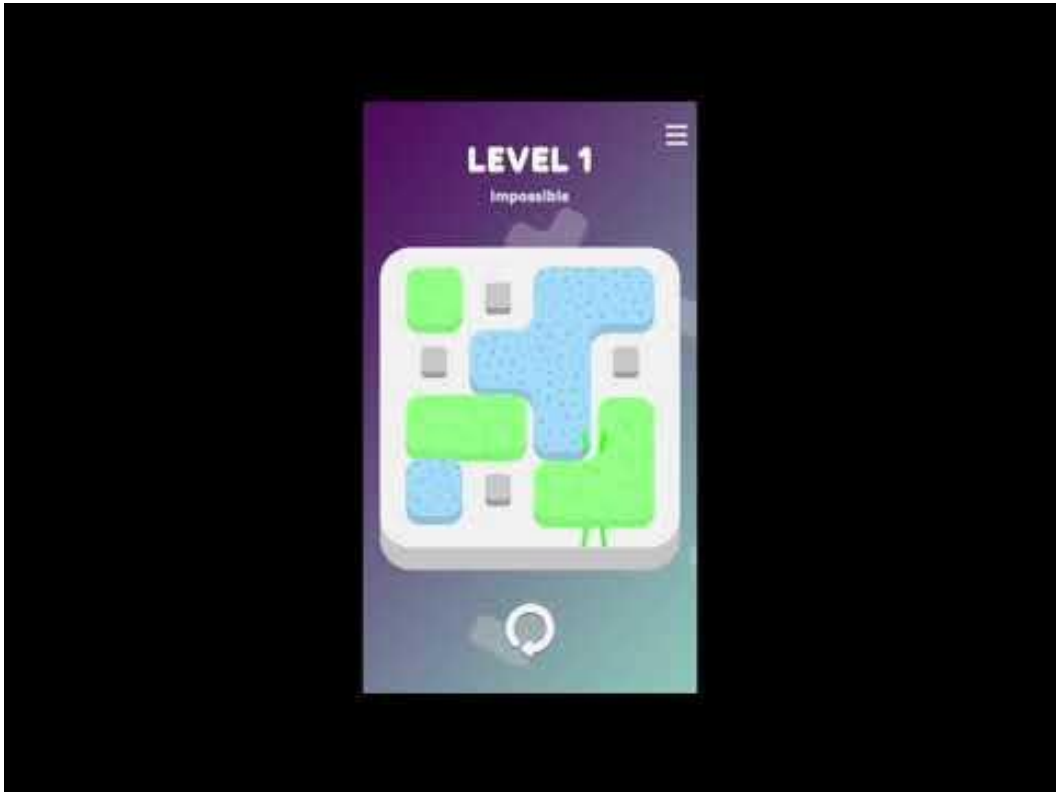
Music Story


Name
Patch
Chn
Vol
Trans
Port
Piano
Acoustic Grand Piano
Celesta
Choir Aahs
Glockenspiel
Pizzicato Strings
String Ensemble 1
Timpani
NARRATOR
ELLE
TEE
COMPASS
ELEPHANT
Piccolo
String Ensemble
Drums


TheDragon.in
File Edit View Story [Inky Debug] Window Help
- T: ^I'm ex*ci*ted to be here with you to*day on this ad*ven*ture!
* [E: Me too. You're cool.]
-- .#25 #26
~ elleTeeBond ++
-- E: ^Me too. You're cool.
-- T: ^Dit*to, kid*do.
-- N: ^Tee elbows Elle lovingly.
-- E: ^Arr*ight, arr*ight...
-- N: Elle suppresses a smile.
* E: [I don't *need* you here...]
-- .#26 #27
~ elleTeeBond --
-- E: ^I don't NEED you here...
-- T: ^Burn.
-- T: You know it's cool to be nice, right?
-- E: I just call it like it is.
-- E: I'm an in*de*pen*dent girl!
- .#27 #28 #29
- N: ::SetBackgroundImage(castleFar1)
- N: ^They stop walking and Elle points up ahead.
- E: ^There!
- E: There it is! In the dist*ance!
- N: ^They spot the spires of a magestic castle nestled in the woods.
- T: ^Ooh.
- N: They stroll directly towards the castle.
- T: ^Hey. Elle.
* [Look at Tee]
-- .#29 #30
~ elleTeeBond ++
-- N: ^Elle looks at Tee.
* [Ignore Tee]
-- .#30 #31
~ elleTeeBond --
-- N: ^Elle ignores Tee.
-- T: ^Yo. Elle.
-- N: Elle stares resolutely ahead.
-- T: It's me. Tee.
-- T: You know, your friend?
-- N: Elle remains silent.
- .#31 #32 #33
- T: ^I just want to say*.*.*.
- T: that I'm glad you're my best friend.
* [E: I like you too, Tee.]
-- .#33 #34
~ elleTeeBond += 2
-- E: ^I like you too, Tee.
-- N: ^Tee beams.
* [E: Good for you.]
-- .#34 #35
~ elleTeeBond -= 2
-- E: ^Good for you.
-- N: ^Tee says nothing, but nods.
-- N: They continue in silence for a few moments.
-- N: Elle bounds on enthusiastically, as Tee walks quietly beside her.
- .#35 #36 #37
- N: ::SetBackgroundImage(castleWalls1)
- N: ^They've reached the bounds of the castle.
- N: It towers magestically before them.
- E: ^All right! We've made it to the cas*tle walls!



Combi

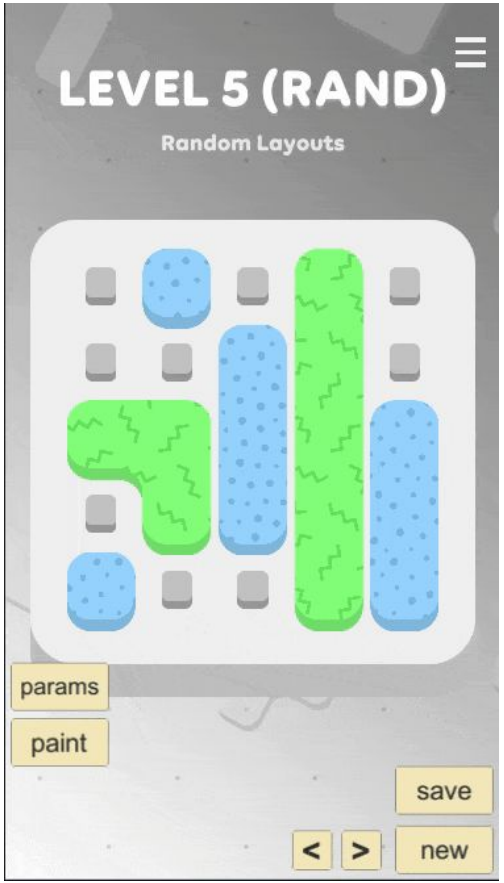


Combi

- Mobile casual game
- Wanted 1,000+ levels
- Game too complex for procedural level gen

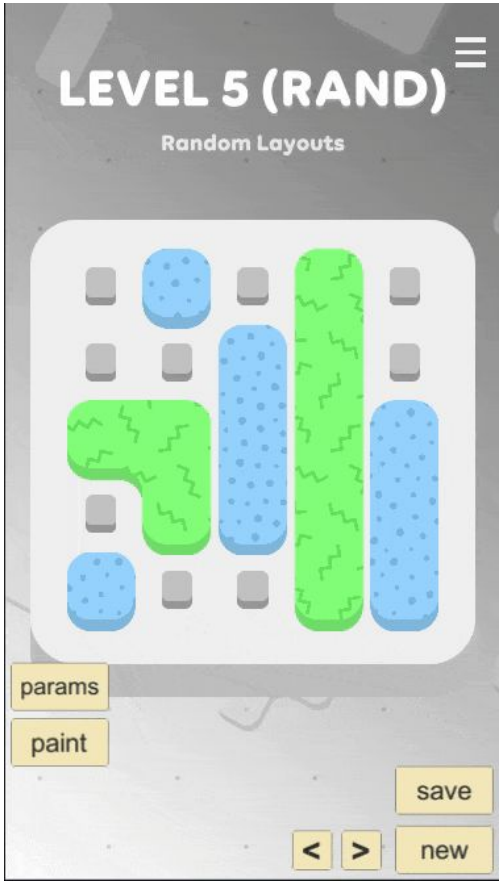
Combi Level Editor

- Solution: Made editor for my phone
  1. Generate rand level (with adjustable params)
  2. Play it
  3. Edit it manually (“paint” tools)
  4. Save/tag level
- Adding 1 level cost ~45 sec *total* :)



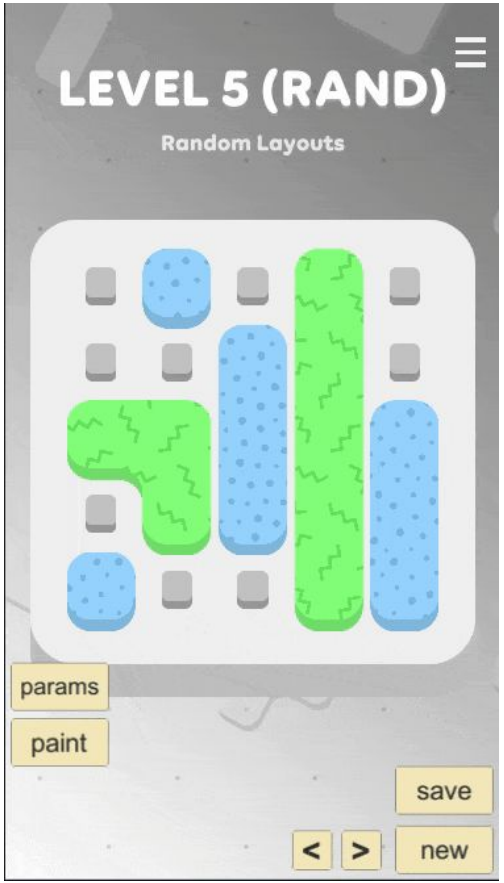
Combi Level Editor

- Made levels after waking up
- Made levels on subway
- Made levels before bed
- etc.



Combi Level Editor: Outcome

COST	=	**~3 days**
PAYOFF	=	**my sanity <3**



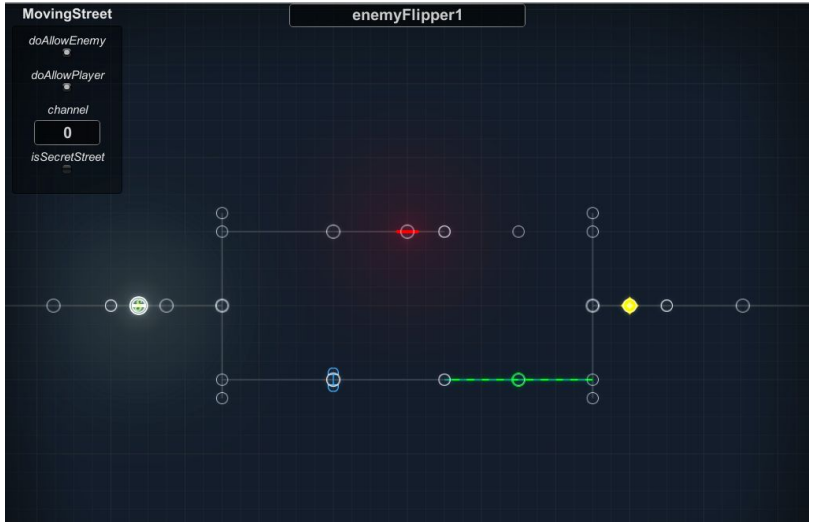
3. When to Make a Tool

When is Adding a Tool Right for Me?

- Hack in content *first*
- THEN you'll know what tools you do/don't need

```
// Intro to Moving Streets
else if (levelKey == L_introMovingStreets) {
  float y = 70;
  l.addStreet(-400,y, 0,y);
  l.addStreet(-400,y, -200,y-200);
  l.addStreet(-200,y-200, -200,y);
  l.addMovingStreet(0,y-100, 200,y-100,  0,100,  0);
  l.addStreet(200,y, 400,y);

  l.addStreetFlipper(0,0.25, 0);
  l.addStar(4, 0.5);

  l.addStreet(-200,y, -200,300);
//   l.addStreet(400,0, -200,300);

  l.playerStartingStreetIndex = 0;
}
```

eventually



When is Adding a Tool Right for Me?

You may be wasting more time than you think.

1. Identify your preferences.
2. Challenge them.

Easy-going?

Not bothered by repeating actions?

Efficiency-fueled?

Wanna build tool the moment you think of it?

Easy-going?

- Look at your process critically for **REPEATING ACTIONS** or **WAITING**
- Examples:
  - Renaming/editing files in bulk
  - Moving mouse/eyes far distances
  - Each code edit takes 15+ seconds to recompile
- Challenge your complacency!
- Saving time might be easier than you think

Efficiency-fueled?

- Try hacking content in first instead
- Design may change
- Very specific tool → Higher chance becoming obsolete
- Sunk-cost fallacy
  - Having tool for XYZ → Harder to cut/change XYZ

When is Adding a Tool Right for Me?

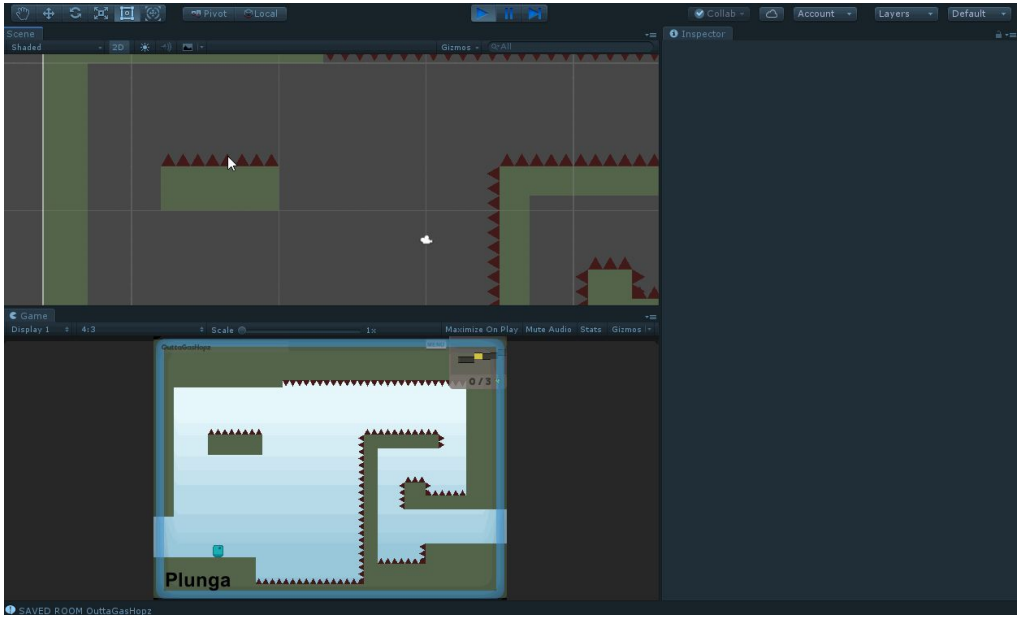
- My threshold for investing in tools → How annoying it was without one
- I'd try to live without it first
- Repeatedly think "This is annoying/inefficient"
- How long would a tool take? Would it save time?

Example: Spike Rotation

How to rotate a spike:

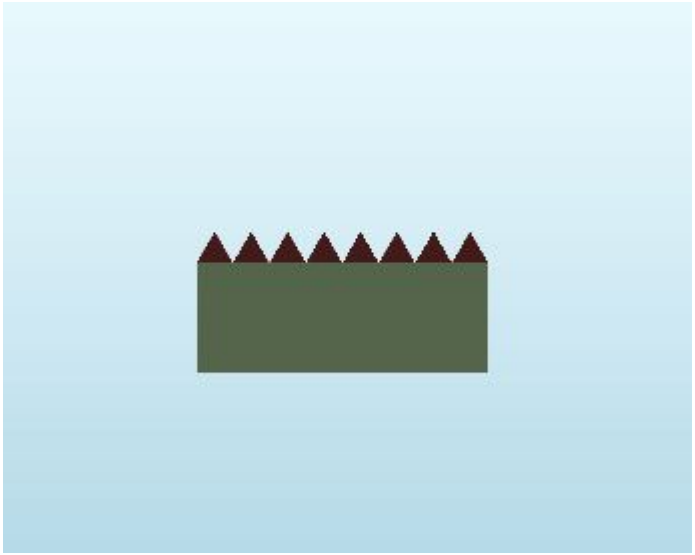
1. Select Spike, select Z rotation, type in “90”
2. Is 90 Left or is -90 Left?
3. Idk, try -90
4. Oops, no, Left *is* 90
5. Remove the “-”

I did this... for each spike



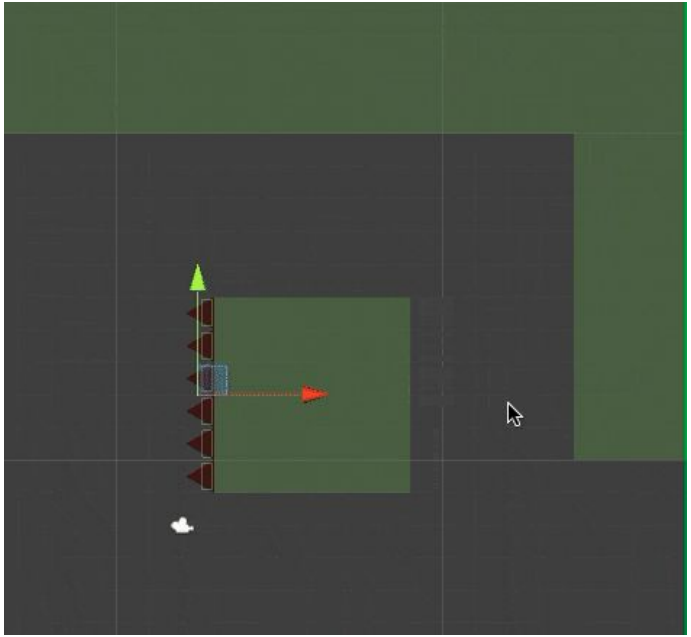
Example: Spike Rotation

- Annoying for weeks.
- Q: What's the EASIEST way to speed this up?
- A: CTRL + R to rotate 90 degrees
- Added CTRL + R in 15 minutes



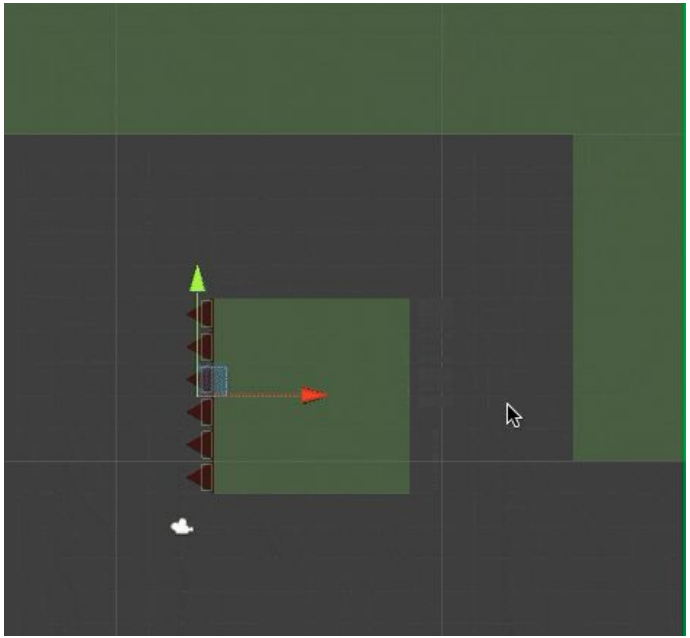
Example: Spike Rotation

- CTRL + R better... but still clunky
- Q: How long WOULD auto-rotating spikes take?
- A: ~2 hours
- Q: Do I *want* to program that?
- A: YES AT THIS POINT YES



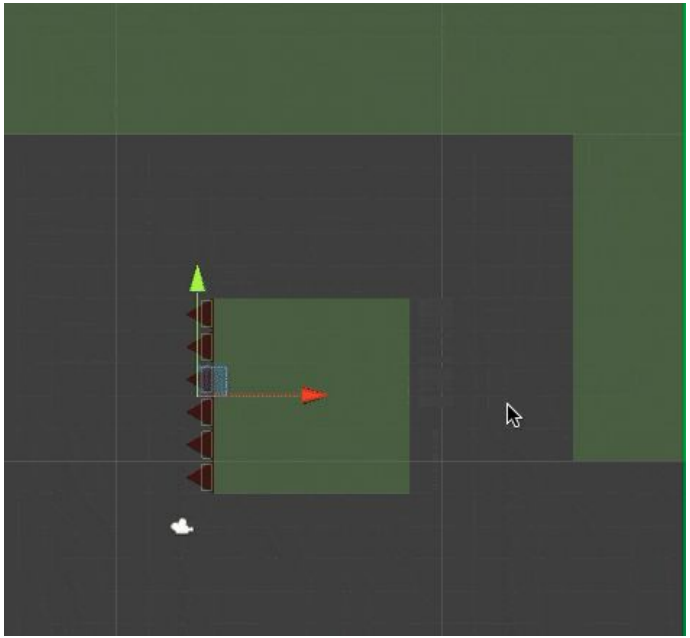
Example: Spike Rotation

- Added auto-rotate in 1.5 hours
- Way better! Saves SO much time!
- Manual rotation = ~10 sec + mental resources
- 50x / day = **3 hours/month** (+lost *groove*!)

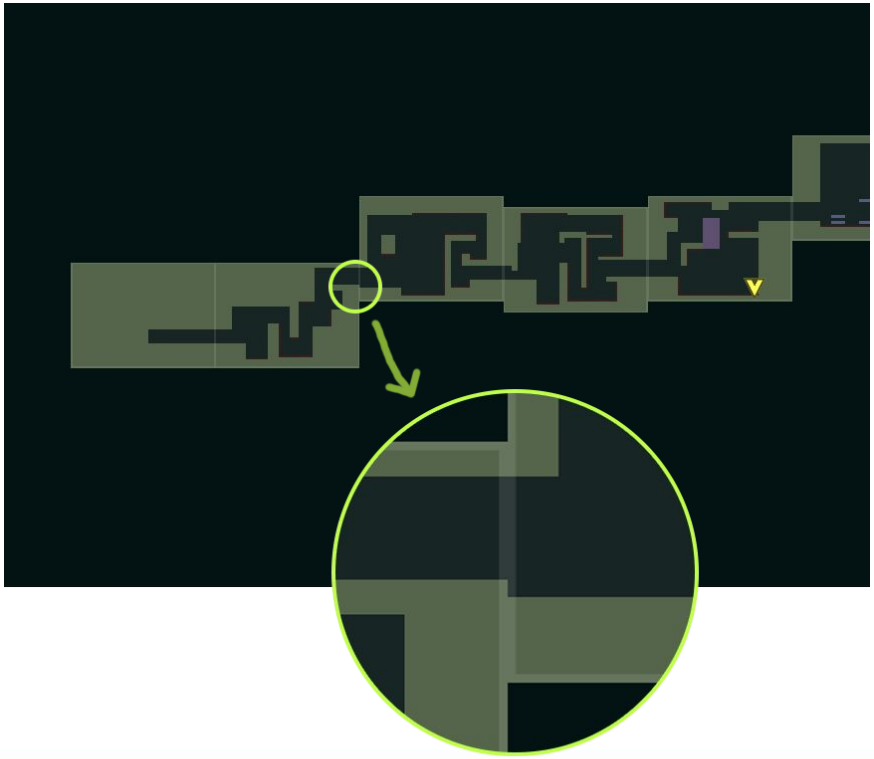


Spike Rotation: Outcome

COST	=	**1.5 hours**
PAYOFF	=	**awesome.**



Example: Misaligned Rooms



Example: Misaligned Rooms

- I wanted to add feature to highlight errors
- Q: Do I *want* to add this?
- A: Not really...
- → Didn't add it

When To Add a Tool

- Is there a right answer for either of these?
- No.
- Game development is messy!

Incomplete Tools

- Goal is to SAVE TIME, not make an amazing tool
- Linelight MapEditor had bugs
  - Had to rapidly click 2-10 times to open a level
  - Some level rect bounds were wrong
  - Connecting levels still required opening/jimmying each level manually
- Wasn’t necessary to fix every bug
- Copypastaed the code for Bouncemeister, fixed lots of stuff *then*

4. My Unity Shortcuts

My Shortcuts

- Unity-specific
- Lots of little time-savers I copy into new projects
- Use for inspiration! You can save time!!

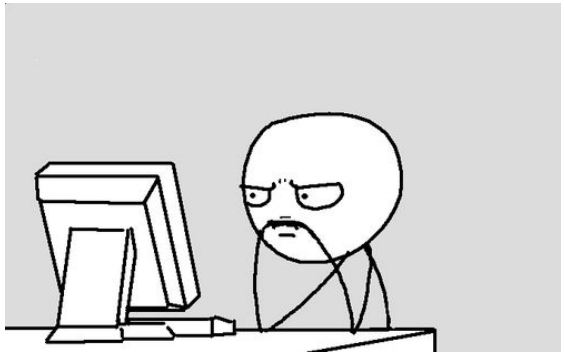
https://drive.google.com/drive/folders/**140fGiWSgVx6scMnMLtKYbEGQXG-0cqJU**

cutt.ly/**gdc**2020**tools**talk

LINK TO CODE

Unity Shortcut: Reload Scene on Script Reload

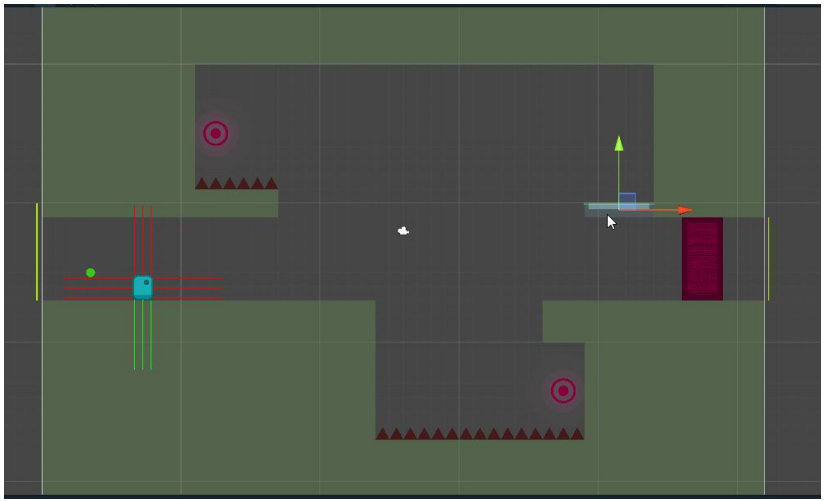
- Projects often break on recompile
- Don’t have to stop/run after every change
- Saves 10 seconds for every compile!! <3



```
#if UNITY_EDITOR
    [UnityEditor.Callbacks.DidReloadScripts]
    private static void OnScriptsReloaded() {
        if (UnityEditor.EditorApplication.isPlaying) {
            SceneHelper.ReloadScene();
        }
    }
#endif
```

Unity Shortcut: Snap to Grid in UnityEditor

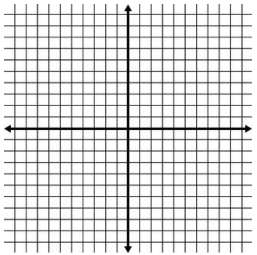
```
private void Update() {
     #if UNITY_EDITOR
     pos = new Vector3(
          Mathf.Round(pos.x/GridSize)*GridSize,
          Mathf.Round(pos.y/GridSize)*GridSize,
          pos.z);
     #endif
}
```



Unity Shortcut: Various Structs

- Color255.cs, ColorHSB.cs
  - `new Color255(120,255,18).ToColor();`
  - `new ColorHSB(0.42f, 0.95f, 0.95f).ToColor();`

- Vector2Int.cs
  - Vector2, but x/y are int instead of float
  - I use Vector2Int for every grid-based game



Code Shortcut: GameUtils.cs

- Static class in ALL my projects
- Common functions like...
  - Parent and reset a Transform
  - Size a SpriteRenderer by pixels
  - Set UIGraphic alpha
  - Change ParticleSystem attributes
  - Set Editor camera pos
  - etc.

Code Shortcut: MathUtils.cs

- Return random bool
- Merge two Rects
- Round to X decimal places
- Vector2 Abs, Max, Min, etc.
- etc.

Wrap-up

cutt.ly/**gdc**2020**tools**talk

Wrap-up

- Avoid generalizing. Make tools specific for your game!
- Tools will feel “unfinished.” That’s ok!
- Making mistakes is okay! It’s ok to learn!

cutt.ly/**gdc**2020**tools**talk

Wrap-up

- Good tools keep designers in flow state
- Look at your process critically to find wasted time
  - Repeating actions? Perfect opportunity for a tool!
  - Trigger-happy to add a tool? Try hacking it in next time instead.
- Leverage your programmer's enthusiasm/interest

- You are awesome <3

cutt.ly/**gdc**2020**tools**talk

Brett Taylor

brett@mydogzorro.com

batzerk