CMPM 120

Text

Objectives

By the end of today you should be able to...

1. Text & Fonts
    a. Demonstrate how Phaser displays displays text
    b. Demonstrate how to include custom fonts
    c. Practice using git
        i. Particularly, practice making a pull request

Pair up

Fork the repository

Why GitHub? Enterprise Explore Marketplace Pricing Search / Sign in Sign up

ikarth / phasertext

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights

Join GitHub today

GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.

Sign up

No description, website, or topics provided.

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or dow

ikarth example javascript for fonts Latest commit 90eef17 7 minutes ago

assets initial commit 1 hour ago

Download the repository

git clone http://github.com/ikarth/phasertext

cd phasertext

git branch *yourname*

git checkout *yourname*

Why talk about text and fonts?

Most games have some form of text, even just for the score

Good use of text and fonts makes a huge difference to the look and feel of your game

Therefore, this will be useful in nearly every game you make, including the ones in this class

Text Style Properties

font string <optional> 'bold 20pt Arial'

The style and size of the font.

fontStyle string <optional> (from font)

The style of the font (eg. 'italic'): overrides the value in style.font.

fontVariant string <optional> (from font)

The variant of the font (eg. 'small-caps'): overrides the value in style.font.

fontWeight string <optional> (from font)

The weight of the font (eg. 'bold'): overrides the value in style.font.

fontSize string |number <optional> (from font)

The size of the font (eg. 32 or '32px'): overrides the value in style.font.

backgroundColor string <optional> null

Text Style Properties (continued)

backgroundColor string <optional> null

A canvas fillstyle that will be used as the background for the whole Text object. Set to null to disable.

fill string <optional> 'black'

A canvas fillstyle that will be used on the text eg 'red', '#00FF00'.

align string <optional> 'left'

Horizontal alignment of each line in multiline text. Can be: 'left', 'center' or 'right'. Does not affect single lines of text (see textBounds and boundsAlignH for that).

boundsAlignH string <optional> 'left'

Horizontal alignment of the text within the textBounds. Can be: 'left', 'center' or 'right'.

boundsAlignV string <optional> 'top'

Text Style Properties (continued)

boundsAlignV string <optional> 'top'

Vertical alignment of the text within the textBounds. Can be: 'top', 'middle' or 'bottom'.

stroke string <optional> 'black'

A canvas stroke style that will be used on the text stroke eg 'blue', '#FCFF00'.

strokeThickness number <optional> 0

A number that represents the thickness of the stroke. Default is 0 (no stroke).

wordWrap boolean <optional> false

Indicates if word wrap should be used.

wordWrapWidth number <optional> 100

The width in pixels at which text will wrap.

maxLines number <optional> 0

Text Style Properties (continued)

wordWrapWidth number <optional> 100

The width in pixels at which text will wrap.

maxLines number <optional> 0

The maximum number of lines to be shown for wrapped text.

tabs number <optional> 0

The size (in pixels) of the tabs, for when text includes tab characters. 0 disables. Can be an array of varying tab sizes, one per tab stop.

Text Style Properties

font	string	'bold 20pt Arial'
fontStyle	string	(from font)
fontVariant	string	(from font)
fontWeight	string	(from font)
fontSize	string | number	(from font)
backgroundColor	string	null
fill	string	'black'
align	string	'left'
boundsAlignH	string	'left'
boundsAlignV	string	'top'

stroke	string	'black'
strokeThickness	number	0
wordWrap	boolean	false
wordWrapWidth	number	100
maxLines	number	0
tabs	number	0

https://photonstorm.github.io/phaser-ce/Phaser.Text.html

Phaser's Default Style is Cliché and Passé

"bold 20pt Arial"

Both boring *and* everyone has seen it before.

You can do better!

Phaser.Text extends Phaser.Sprite

What does this mean for us?

Phaser Text as Extended Sprites

Since Text inherits from the Sprite class, we can read and apply sprite-like properties and methods (even weird stuff like damage and health).

That means we can do interesting interactive stuff like grant input control over text or apply physics bodies to text objects.

Custom Fonts

Bitmap Fonts

"BitmapText objects work by taking a texture file and an XML or JSON file that describes the font structure. It then generates a new Sprite object for each letter of the text, proportionally spaced out and aligned to match the font structure."
Phaser API

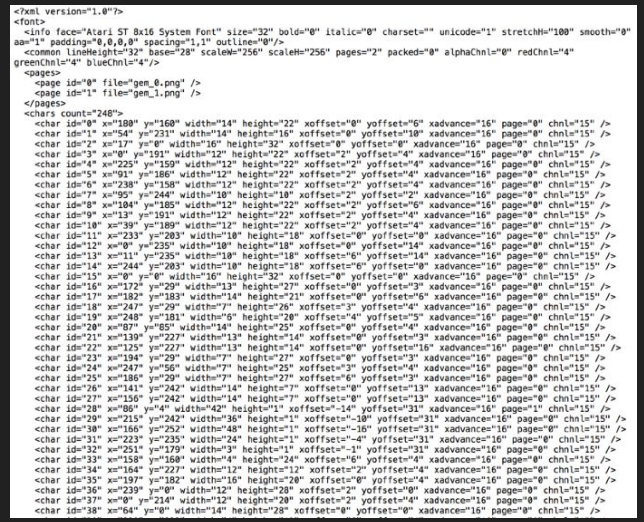
Translation: bitmap text is a font that has been laid out in a grid (like a sprite sheet). Bitmap text is less flexible than a text object, but renders much faster.

Atari ST 8x16 System Font

I got this bitmap font from the Phaser examples

gem.png

+



```
<?xml version="1.0"?>
<font>
  <info face="Atari ST 8x16 System Font" size="32" bold="0" italic="0" charset="" unicode="1" stretchH="100" smooth="0"
aa="1" padding="0,0,0,0" spacing="1,1" outline="0"/>
  <common lineHeight="32" base="28" scaleW="256" scaleH="256" pages="2" packed="0" alphaChnl="0" redChnl="4"
greenChnl="4" blueChnl="4"/>
  <pages>
    <page id="0" file="gem_0.png" />
    <page id="1" file="gem_1.png" />
  </pages>
  <chars count="240">
    <char id="0" x="180" y="160" width="14" height="22" xoffset="0" yoffset="6" xadvance="16" page="0" chnl="15" />
    <char id="1" x="54" y="231" width="14" height="16" xoffset="0" yoffset="10" xadvance="16" page="0" chnl="15" />
    <char id="2" x="17" y="0" width="16" height="32" xoffset="0" yoffset="0" xadvance="16" page="0" chnl="15" />
    <char id="3" x="0" y="191" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="4" x="225" y="159" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="5" x="91" y="186" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="6" x="238" y="158" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="7" x="95" y="244" width="10" height="10" xoffset="2" yoffset="2" xadvance="16" page="0" chnl="15" />
    <char id="8" x="104" y="185" width="12" height="22" xoffset="2" yoffset="6" xadvance="16" page="0" chnl="15" />
    <char id="9" x="13" y="191" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="10" x="39" y="189" width="12" height="22" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="11" x="233" y="203" width="10" height="18" xoffset="0" yoffset="0" xadvance="16" page="0" chnl="15" />
    <char id="12" x="0" y="235" width="10" height="18" xoffset="0" yoffset="14" xadvance="16" page="0" chnl="15" />
    <char id="13" x="11" y="235" width="10" height="18" xoffset="0" yoffset="14" xadvance="16" page="0" chnl="15" />
    <char id="14" x="244" y="203" width="10" height="18" xoffset="0" yoffset="0" xadvance="16" page="0" chnl="15" />
    <char id="15" x="0" y="0" width="16" height="32" xoffset="0" yoffset="0" xadvance="16" page="0" chnl="15" />
    <char id="16" x="172" y="29" width="13" height="27" xoffset="0" yoffset="3" xadvance="16" page="0" chnl="15" />
    <char id="17" x="182" y="183" width="14" height="21" xoffset="0" yoffset="6" xadvance="16" page="0" chnl="15" />
    <char id="18" x="247" y="29" width="7" height="26" xoffset="3" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="19" x="248" y="181" width="6" height="20" xoffset="4" yoffset="5" xadvance="16" page="0" chnl="15" />
    <char id="20" x="87" y="85" width="14" height="25" xoffset="0" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="21" x="139" y="227" width="13" height="14" xoffset="0" yoffset="3" xadvance="16" page="0" chnl="15" />
    <char id="22" x="125" y="227" width="13" height="14" xoffset="0" yoffset="16" xadvance="16" page="0" chnl="15" />
    <char id="23" x="194" y="29" width="7" height="27" xoffset="0" yoffset="3" xadvance="16" page="0" chnl="15" />
    <char id="24" x="247" y="56" width="7" height="25" xoffset="3" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="25" x="186" y="29" width="7" height="27" xoffset="6" yoffset="3" xadvance="16" page="0" chnl="15" />
    <char id="26" x="141" y="242" width="14" height="7" xoffset="0" yoffset="13" xadvance="16" page="0" chnl="15" />
    <char id="27" x="156" y="242" width="14" height="7" xoffset="0" yoffset="13" xadvance="16" page="0" chnl="15" />
    <char id="28" x="86" y="4" width="42" height="1" xoffset="-14" yoffset="31" xadvance="16" page="1" chnl="15" />
    <char id="29" x="215" y="242" width="36" height="1" xoffset="-10" yoffset="31" xadvance="16" page="0" chnl="15" />
    <char id="30" x="166" y="252" width="48" height="1" xoffset="-16" yoffset="31" xadvance="16" page="0" chnl="15" />
    <char id="31" x="223" y="235" width="24" height="1" xoffset="-4" yoffset="31" xadvance="16" page="0" chnl="15" />
    <char id="32" x="251" y="179" width="3" height="1" xoffset="-1" yoffset="31" xadvance="16" page="0" chnl="15" />
    <char id="33" x="158" y="160" width="4" height="24" xoffset="6" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="34" x="164" y="227" width="12" height="12" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="35" x="197" y="182" width="16" height="20" xoffset="0" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="36" x="239" y="0" width="12" height="28" xoffset="2" yoffset="0" xadvance="16" page="0" chnl="15" />
    <char id="37" x="0" y="214" width="12" height="20" xoffset="2" yoffset="4" xadvance="16" page="0" chnl="15" />
    <char id="38" x="64" y="0" width="14" height="28" xoffset="0" yoffset="0" xadvance="16" page="0" chnl="15" />
```

gem.xml

```
// some global variables
var game;

// wait for browser window load, then start the party
window.onload = function() {
    game = new Phaser.Game(650, 650);
    game.state.add('Play', Play);
    game.state.start('Play');
}

var Play = function(game){};
Play.prototype = {
    preload: function() {
        game.load.path = '../assets/fonts/';
        game.load.bitmapFont('gem', 'gem.png', 'gem.xml');
    },
    create: function() {
        // bitmapText(x, y, font, text, size, group)
        text01 = game.add.bitmapText(32, 32, 'gem', 'Bitmap text, yeah!', 12);
        text02 = game.add.bitmapText(32, 64, 'gem', 'OK, that was too small...', 24);
        text03 = game.add.bitmapText(32, 128, 'gem', 'Here\'s the default size of 32 ;p');
        text04 = game.add.bitmapText(32, 256, 'gem', 'NOW I AM SHOUTING!', 64);
        text05 = game.add.bitmapText(32, 375, 'gem', 'BLUR', 256);
    },
    update: function() {

    }
};
```

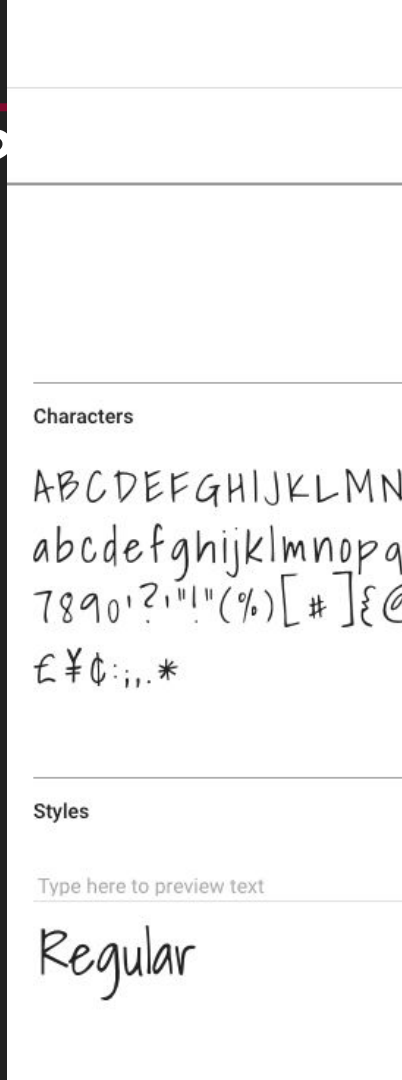
What about Google (or other web) fonts?

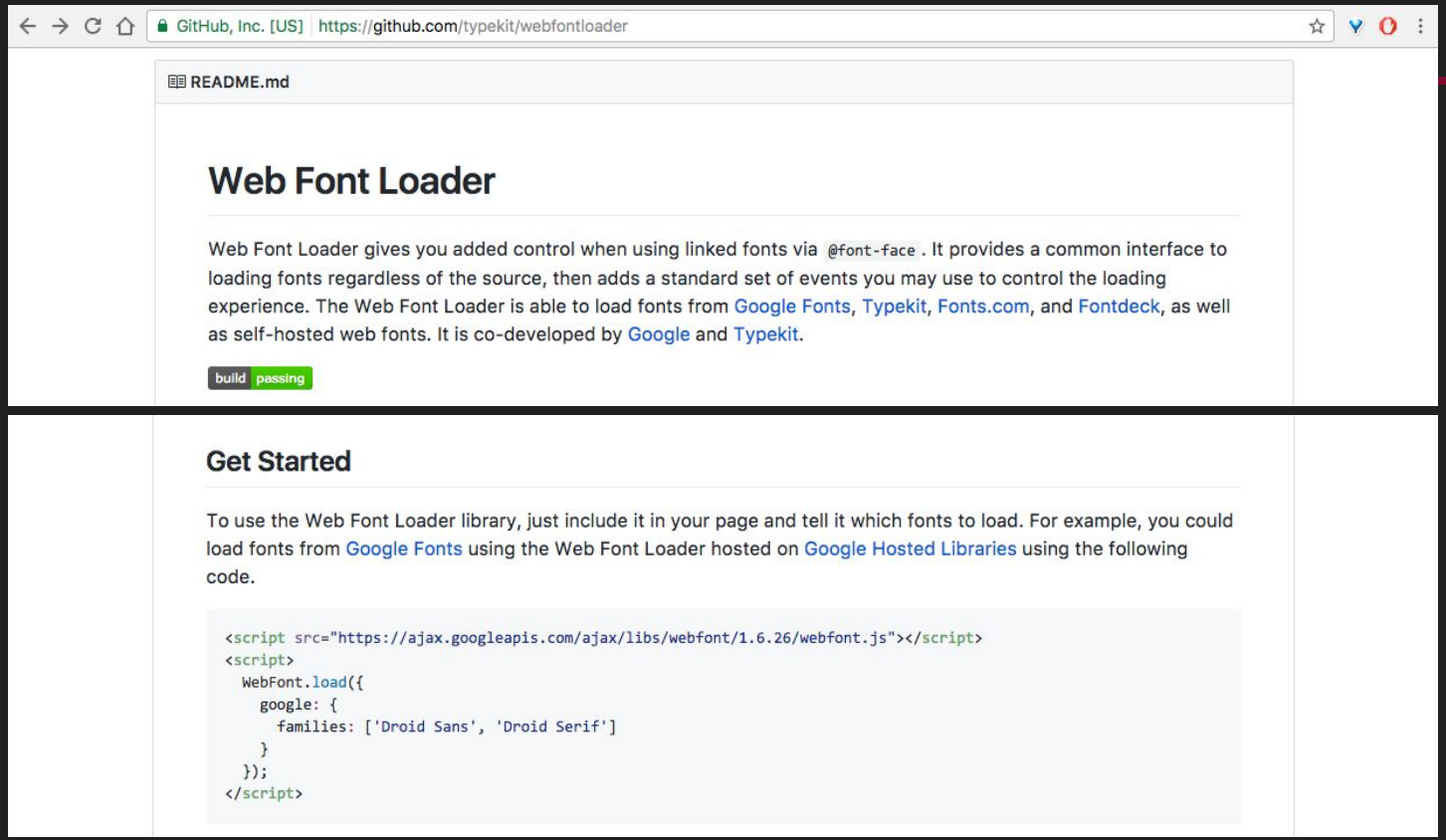
It's complicated.

The Phaser examples page has a complex method for using external fonts that involves creating a timer, loading a script file, then using the font in your game. It feels pretty hack-y due to how fonts are loaded online.

But...there's a (slightly, maybe?) easier way.

(Kudos to Arcanorum on html5gamedevs for this technique)





Web Font Loader!

https://github.com/typekit/webfontloader

Google Hosted Libraries

Search LANGUAGE ALL PRODUCTS

Web Font Loader

snippet>

```
<script src="https://ajax.googleapis.com/ajax/libs/webfont/1.6.26/webfont.js"></script>
```

site:

github.com/typekit/webfontloader

versions:

1.6.26, 1.6.16, 1.5.18, 1.5.10, 1.5.6, 1.5.3, 1.5.2, 1.5.0

Troubleshooting

Seeing an outdated version? Make sure you're not using the "automatic version" links, like `/jqueryui/1/...`, but instead use URLs referring to exact versions. Due to concerns over caching and lack of compatibility between even minor versions, we have deprecated and stopped updating the automatic version aliases some time ago, so they will forever refer to an old version (in order to not break existing sites that still use them).

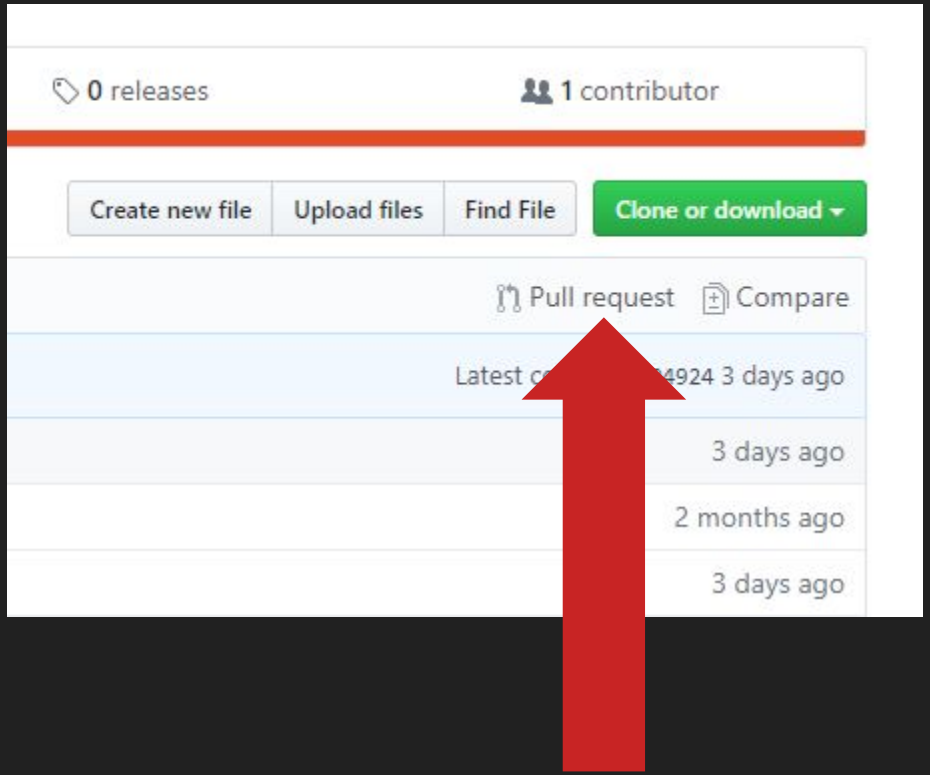
If you encounter problems:

- Look for typos. Remember that JavaScript is a case-sensitive language.
- Use a JavaScript debugger. In Chrome, use the Chrome DevTools. In Firefox, you can use the built-in Firefox DevTools. In IE, you can use the F12 developer tools.

Last updated May 13, 2019.

Contents

```
<!DOCTYPE html>
<html>
<head>
    <title>Phaser</title>
    <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Shadows+Into+Light" rel="stylesheet">
    <!-- Snag the latest Web Font Loader from Google Hosted Libraries -->
    <!-- https://developers.google.com/speed/libraries/#web-font-loader -->
    <script src="https://ajax.googleapis.com/ajax/libs/webfont/1.5.18/webfont.js"></script>
    <script>
        WebFont.load({
            google: {
                // Load fonts here
                families: ['Pacifico', 'Shadows Into Light']
            }
        });
    </script>
    <script type="text/javascript" src="../framework/phaser.min.js"></script>
    <script type="text/javascript" src="text03.js"></script>
</head>
<body>
</body>
</html>
```


0 releases
1 contributor
Create new file
Upload files
Find File
Clone or download
Pull request
Compare
Latest c
4924 3 days ago
3 days ago
2 months ago
3 days ago

Pull Requests

About: https://help.github.com/en/articles/about-pull-requests

Creating a Pull Request:

https://help.github.com/en/articles/creating-a-pull-request

https://help.github.com/en/articles/creating-a-pull-request-from-a-fork

Merging a Pull Request:

https://help.github.com/en/articles/merging-a-pull-request

Phaser Examples: Text

https://phaser.io/examples/v2/category/text

Dialog Systems?

What do we need in a dialog system?

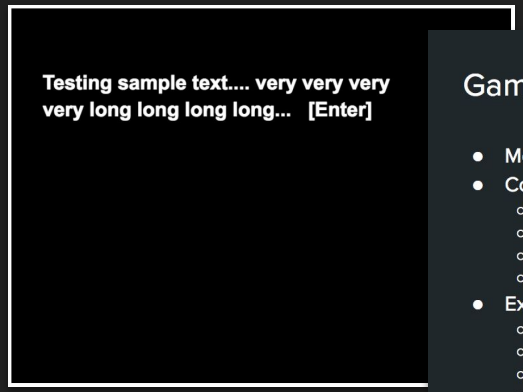
- ???

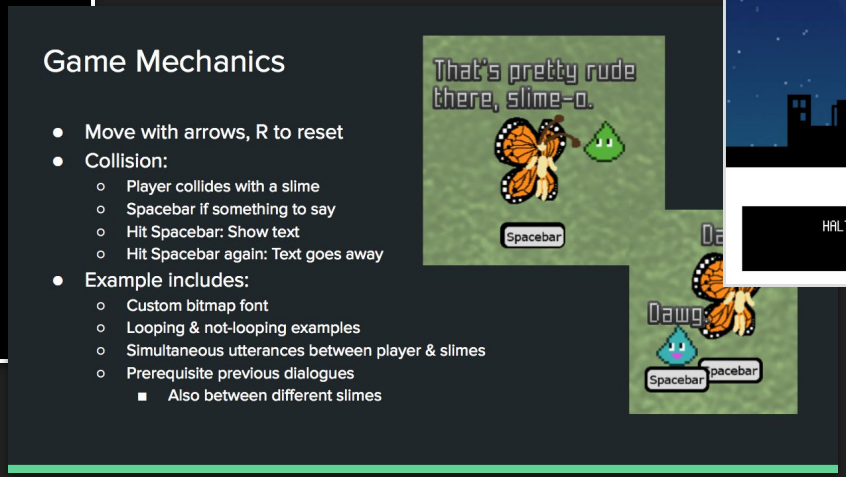
Nathan's Architecture

1. Create structured dialog data in JSON
2. Create and position dialog box sprite
3. Check to see if there are dialog lines remaining in current conversation
4. Check to see if there is a new speaker and tween them into view (and tween out previous speaker)
5. Construct dialog by adding speaker + line
6. Create a timer to “fire” dialog letter by letter
7. Lock input until all characters have printed
8. Increment; repeat

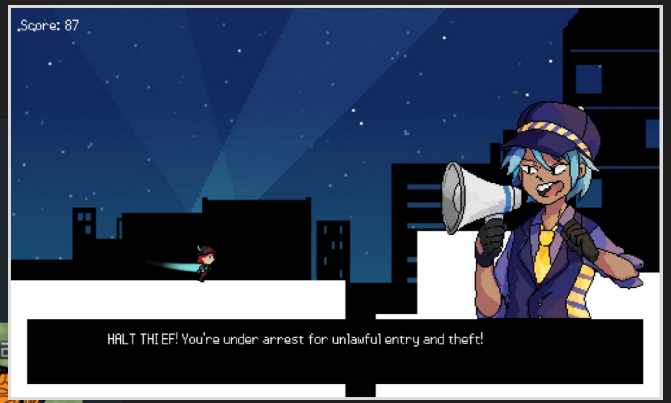
Some dialog systems

Yi's PhaserDialog
https://github.com/kthtes/PhaserDialog





April Grow's dialog system



120 student Tina Peng's timed dialog

Other Narrative Tools

Yarn: https://github.com/InfiniteAmmoInc/Yarn

Javascript port of Yarn: https://github.com/jhayley/bondage.js/

Ink (from Inkle): https://www.inklestudios.com/ink/

Javascript port of Ink: https://github.com/y-lohse/inkjs

More Debugging Tips

Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
    a. assert()
    b. Keep a debugging notebook
2. Make debug tools
    a. Quicker feedback is better
    b. Display values live if possible
3. Only make one change at a time and then test it
4. Just because you paused the game doesn't mean it's paused
    a. And stopping one update doesn't mean you stopped all of them
5. console.log() is slow
    a. Faster to print an array as a string than to individually print the contents

Useful random debugging advice

Walk through your code step by step, explaining to yourself what is supposed to happen

Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
    a. assert()
    b. Keep a debugging notebook
2. Make debug tools
    a. Quicker feedback is better
    b. Display values live if possible
3. Only make one change at a time and then test it
4. Just because you paused the game doesn't mean it's paused
    a. And stopping one update doesn't mean you stopped all of them
5. console.log() is slow
    a. Faster to print an array as a string than to individually print the contents

Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
    a. assert()
    b. Keep a debugging notebook
2. Make debug tools
    a. Quicker feedback is better
    b. Display values live if possible
3. Only make one change at a time and then test it
4. Just because you paused the game doesn't mean it's paused
    a. And stopping one update doesn't mean you stopped all of them
5. console.log() is slow
    a. Faster to print an array as a string than to individually print the contents

Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
    a. assert()
    b. Keep a debugging notebook
2. Make debug tools
    a. Quicker feedback is better
    b. Display values live if possible
3. Only make one change at a time and then test it
4. Just because you paused the game doesn't mean it's paused
    a. And stopping one update doesn't mean you stopped all of them
5. console.log() is slow
    a. Faster to print an array as a string than to individually print the contents

Useful random debugging advice

1. When you find a problem, change something so that same problem can't happen again
   a. assert()
   b. Keep a debugging notebook
2. Make debug tools
   a. Quicker feedback is better
   b. Display values live if possible
3. Only make one change at a time and then test it
4. Just because you paused the game doesn't mean it's paused
   a. And stopping one update doesn't mean you stopped all of them
5. console.log() is slow
   a. Faster to print an array as a string than to individually print the contents

AABB characters and slopes

An example of a real-world physics-and-debugging problem in a game with 2D physics like yours

https://twitter.com/eevee/status/1133248372624613376