



# *punching holes in the sky*

mutating the DOM via wrinkles in space



(what are slots?)

## Receptors for visual (template) content

- *receptors* are structures that receive content that may be integrated into componentized systems

- Default
- Named
- Scoped
- Teleporting





















*Em*

*Jeff*

A miniature house with a peach-colored frame and white roof. Inside, two Calico Critters dolls are seated in wooden chairs. On the left, a tan rabbit-like doll named Em sits in a chair with a yellow floral cushion and a pink floral dress. On the right, a white rabbit-like doll named Jeff sits in a chair with a black and white patterned cushion and a yellow shirt with a green vest. The house has several arched openings in the back wall.

*Em*

*Jeff*



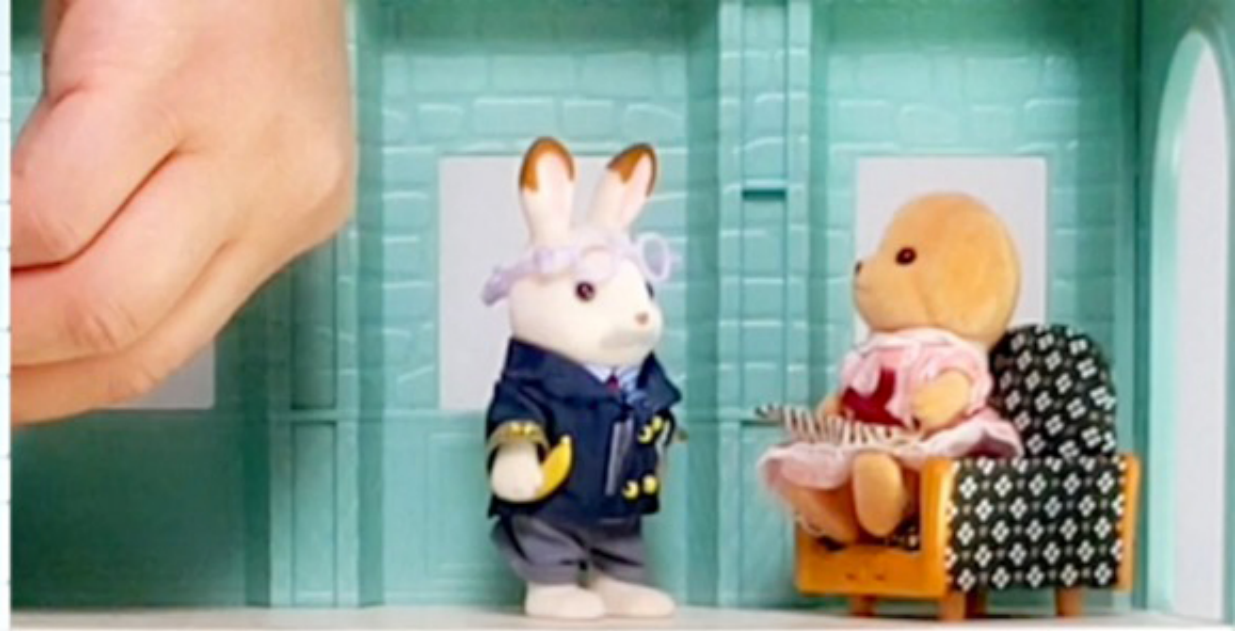












(how?)





VMyPlace.vue

```
<template>  
<BasePlace>
```

```
  <!--can insert visual content here-->
```

```
</BasePlace>  
</template>
```

BasePlace.vue

```
<template>  
<h1>{{ placeName }}</h1>
```

```
• <slot />  
</template>
```



## VMyPlace.vue

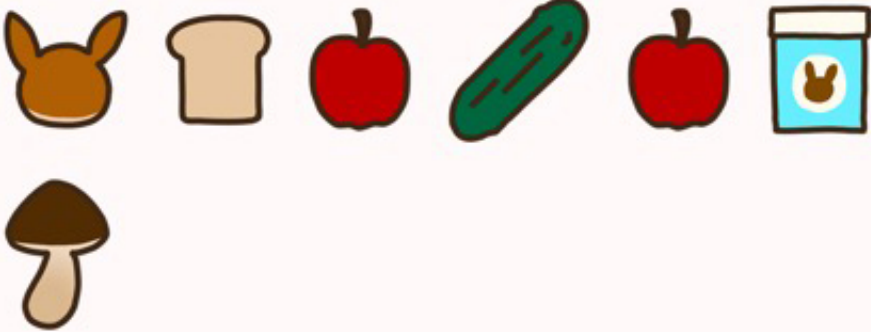
```
<template>
<BasePlace>
  <Pancake />
  <Toast />
  <Apple />
  <Cucumber />
  <Apple />
  <IceCream />
  <Mushroom />
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot />
</template>
```

My Place



The image shows a window titled "My Place" with a light blue header and a light pink body. In the top right corner of the header, there are small icons of a green house and an orange carrot. The main area of the window contains a row of seven icons: a brown rabbit head, a slice of white bread, a red apple, a green cucumber, another red apple, and a blue container with a white rabbit head on its label. Below this row, on the left side, is a single brown mushroom icon.

My Place

About



(she, her)



The image shows a digital interface titled "My Place". On the left, there is a vertical panel labeled "About" containing a circular profile picture of a woman with dark hair and a red top, with the text "(she, her)" below it. To the right of this panel is a larger area containing seven icons arranged in two rows. The top row features a brown rabbit head, a slice of white bread, and a red apple. The bottom row features a green cucumber, a red apple, a blue container with a white rabbit icon on its label, and a brown mushroom. The interface has a light blue background and a teal header bar with a small green crown icon and an orange carrot icon in the top right corner.

## VMyPlace.vue

```
<template>
<BasePlace>
  <Pancake />
  <Toast />
  <Apple />
  <Cucumber />
  <Apple />
  <IceCream />
  <Mushroom />
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>
• <slot />

<slot />
</template>
```

## VMyPlace.vue

```
<template>
<BasePlace>
  <Pancake />
  <Toast />
  <Apple />
  <Cucumber />
  <Apple />
  <IceCream />
  <Mushroom />
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot />
</template>
```

VMyPlace.vue

```
<template>
<BasePlace>
• <template v-slot:about>
  
  <p>(she, her)</p>
</template>

<Pancake />
<Toast />
<...etc>

</BasePlace>
</template>
```

BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot />
</template>
```

## VMyPlace.vue

```
<template>
<BasePlace>
  <template v-slot:about>
    
    <p>(she, her)</p>
  </template>
  • <Pancake />
    <Toast />
    <...etc>
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot />
</template>
```

## VMyPlace.vue

```
<template>
<BasePlace>
  <template v-slot:about>
    
    <p>(she, her)</p>
  </template>

  <template v-slot:default>
    <Pancake />
    <Toast />
    <...etc>

  </template>
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot />
</template>
```



## VMyPlace.vue

```
<template>
<BasePlace>
  <template v-slot:about>
    
    <p>(she, her)</p>
  </template>

  <template v-slot:default>
    <Pancake />
    <Toast />
    <...etc>

  </template>
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

• <slot name="default" />
</template>
```

## VMyPlace.vue

```
<template>
<BasePlace>
  <template v-slot:about>
    
    <p>(she, her)</p>
  </template>

  <template v-slot:default>
    <Pancake />
    <Toast />
    <...etc>

  </template>
</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot name="default">
  <Banana />
  <Banana />
  <Banana />
</slot>
</template>
```

## VMyPlace.vue

```
<template>
<BasePlace>
  <template v-slot:about>
    
    <p>(she, her)</p>
  </template>

</BasePlace>
</template>
```

## BasePlace.vue

```
<template>
<h1>{{ placeName }}</h1>

<slot name="about" />

<slot name="default">
  <Banana />
  <Banana />
  <Banana />
</slot>
</template>
```

My Place

About



(she, her)



why slots are good actually

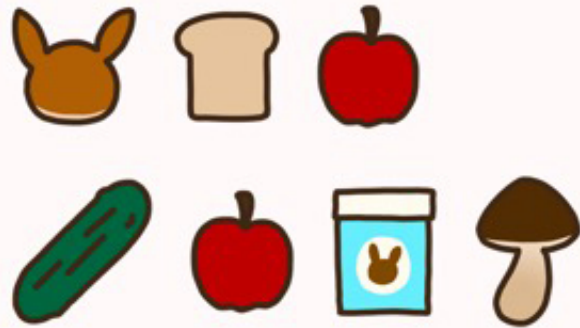
- Repeating frames for visual content
- Parent owns data; child owns framing
- Injected content replaces initial
- Potential use cases for slots:
  - *Buttons*
  - *Layouts*
  - *Modals*

## JasonSpace

About



(he, him)

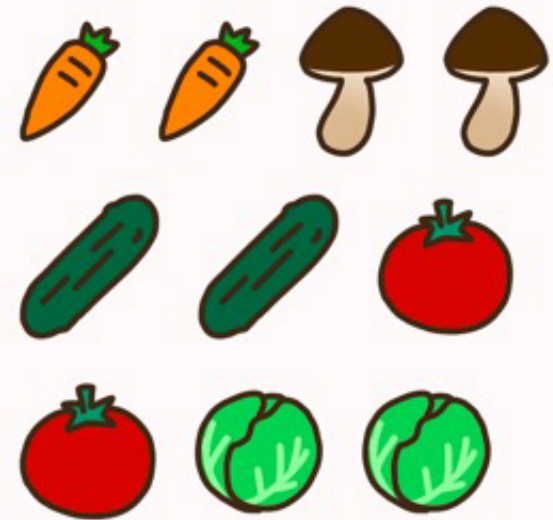


## Matan's Veggie Garden

About



(he, him)



## VJasonSpace.vue

```
<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <Pancake />
    <Toast />
    <...etc>

  </template>
</BasePlace>
</template>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```



## VJasonSpace.vue

```
<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>
    • <button
      name="send banana"
      @click="sendBanana"
    ><Banana /><Banana /><Banana /></button>
  </template>

  <template v-slot:default>
    <...etc>

    <teleport
      v-if="bananas"
      to="#theBananaDimension"
    >
      <Banana v-for="banana in bananas" />
    </teleport>
  </template>
</BasePlace>
</template>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```

## VJasonSpace.vue

```

<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>

    <button
      name= "send banana"
      @click="sendBanana"
    ><Banana /><Banana /><Banana /></button>
  </template>

  <template v-slot:default>
    <...etc>

    <teleport
      v-if="bananas"
      to="#theBananaDimension"
    >
      <Banana v-for="banana in bananas" />
    </teleport>
  </template>
</BasePlace>
</template>

```

## VMatansVeggieGarden.vue

```

<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    • <div id="theBananaDimension" />

      <Carrot />
      <Carrot />
      <Mushroom />
      <...etc>

    </template>
  </BasePlace>
</template>

```

## VJasonSpace.vue

```
<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>

    <button
      name= "send banana"
      @click="sendBanana"
    ><Banana /><Banana /><Banana /></button>
  </template>

  <template v-slot:default>
    <...etc>

    <teleport
      v-if="bananas"
      to="#theBananaDimension"
    >
      <Banana v-for="banana in bananas" />
    </teleport>
  </template>
</BasePlace>
</template>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```

## VJasonSpace.vue

```
<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>

    <button
      name= "send banana"
      @click="sendBanana"
    ><Banana /><Banana /><Banana /></button>
  </template>

  <template v-slot:default>
    <...etc>

    <teleport
      v-if="bananas"
      to="#theBananaDimension"
    >
      <Banana v-for="banana in bananas" />
    </teleport>
  </template>
</BasePlace>
</template>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```

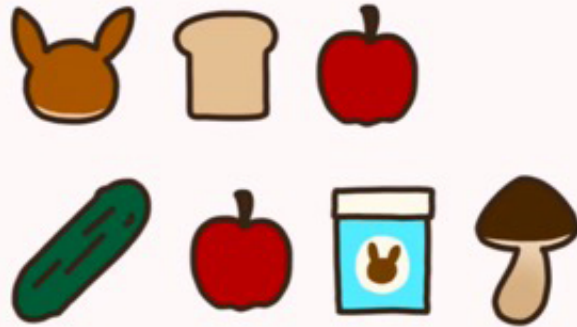


# JasonSpace

About



(he, him)

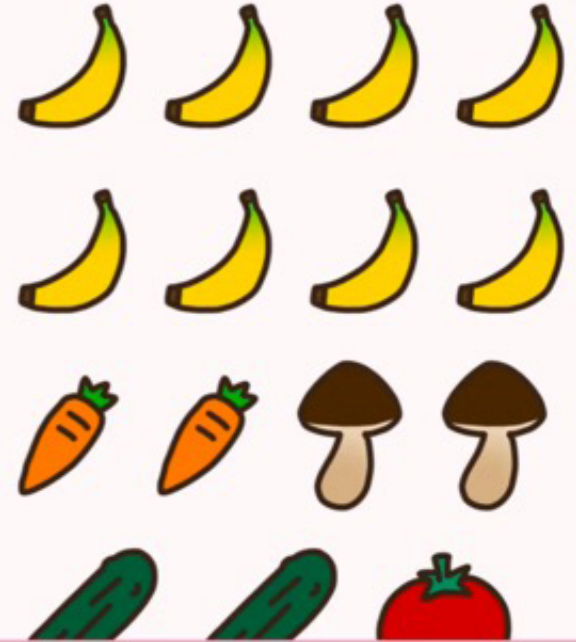


# Matan's Veggie Garden

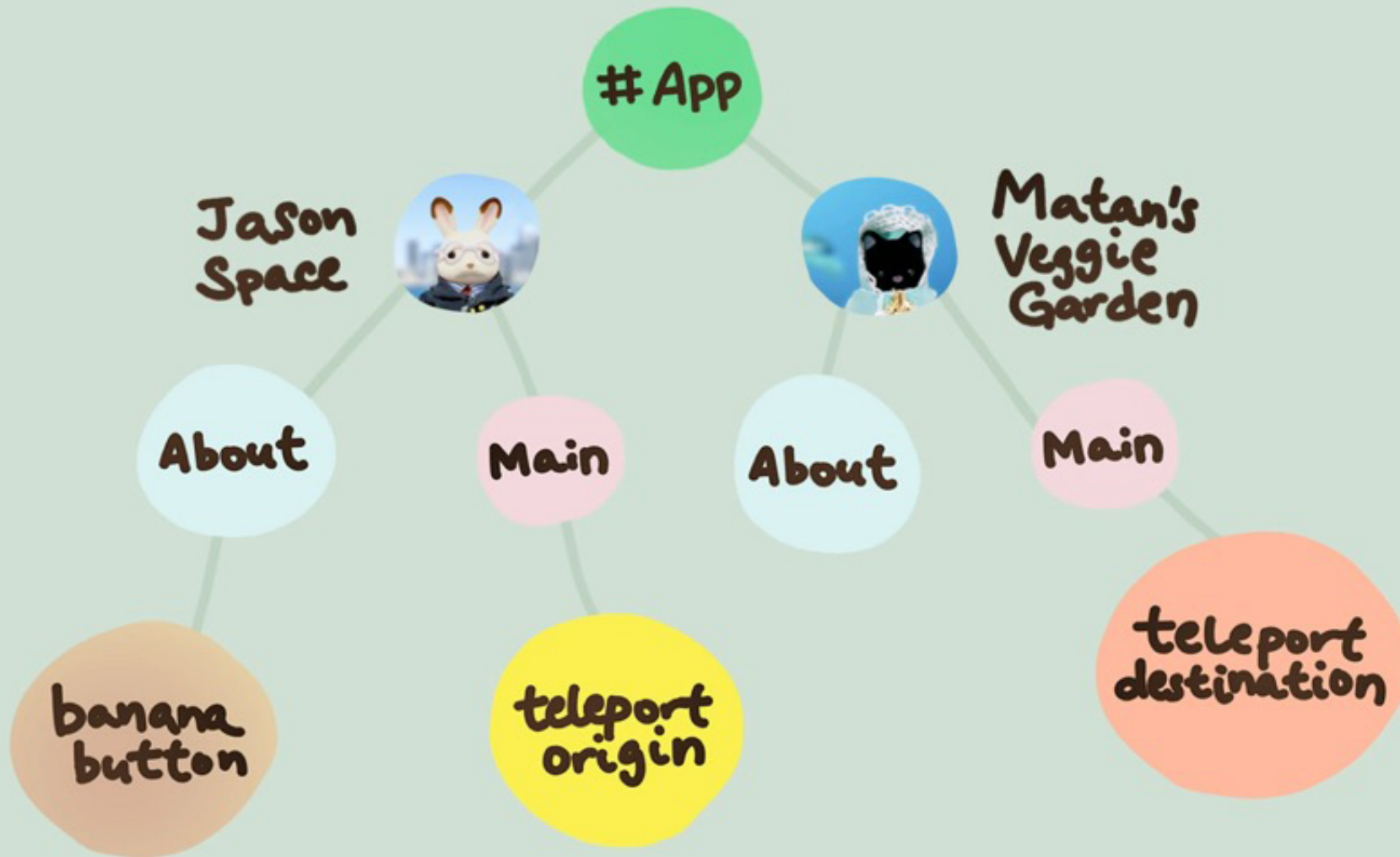
About

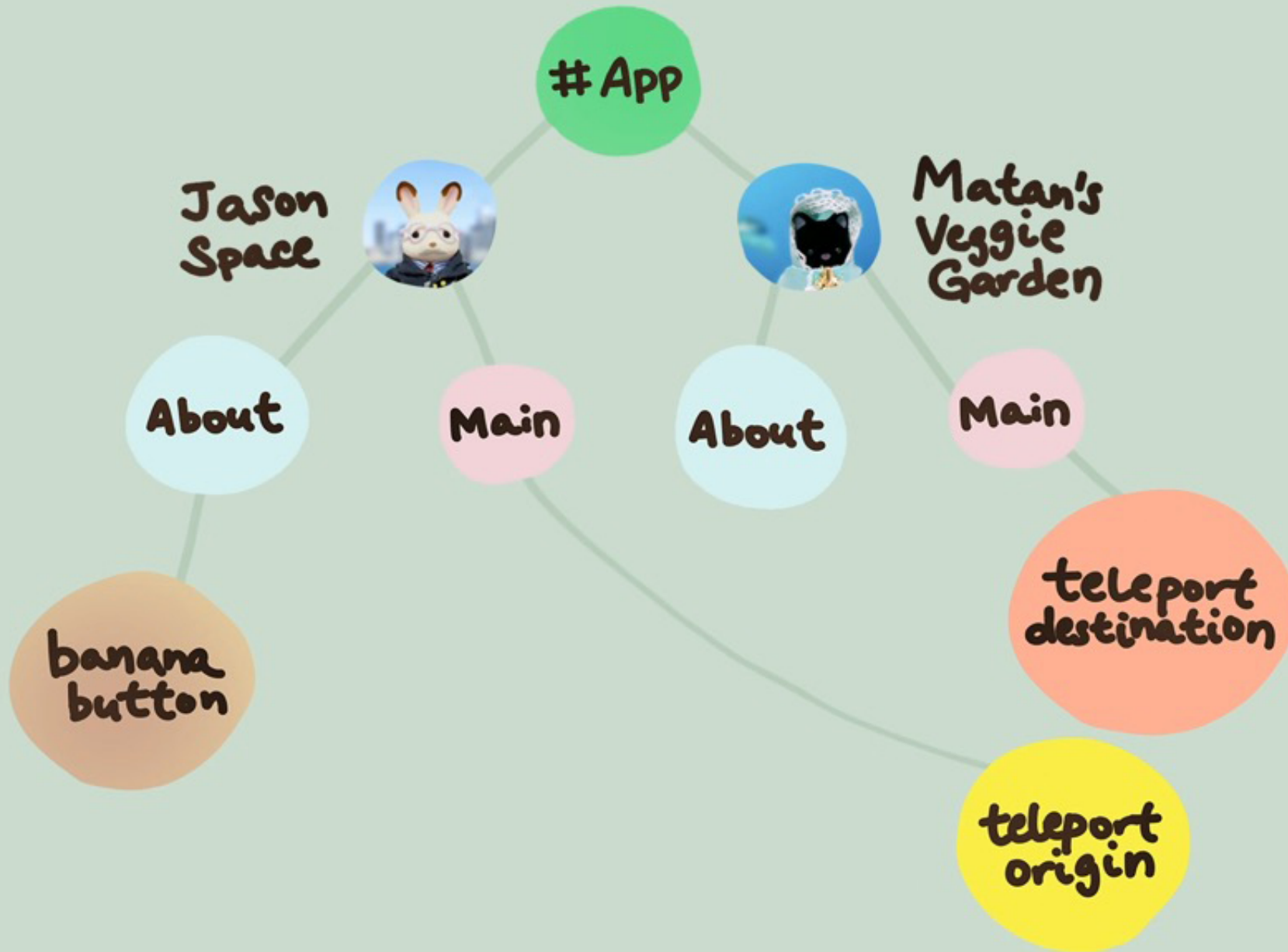


(he, him)









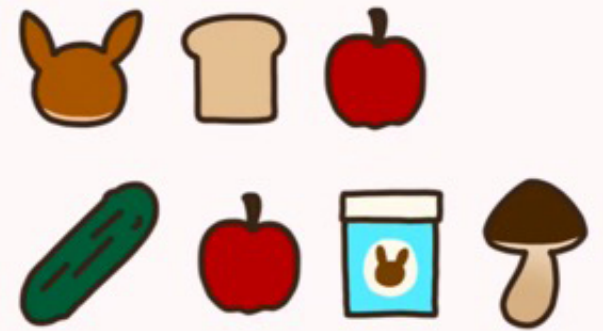


# JasonSpace

About

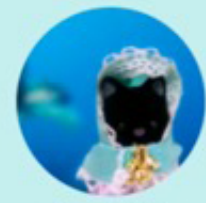


(he, him)

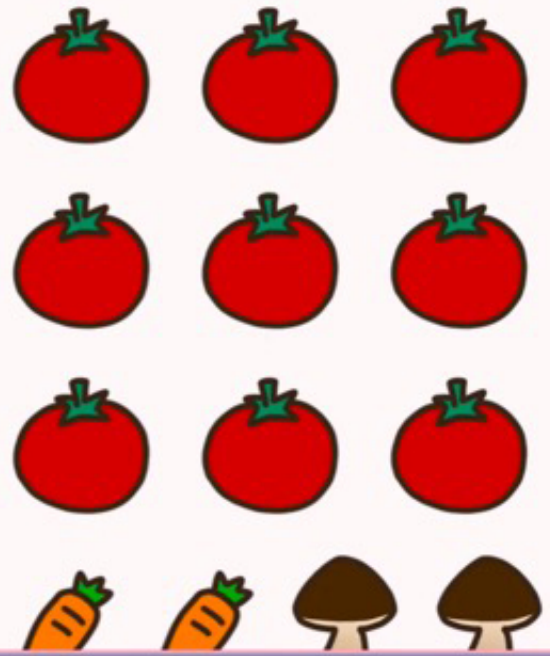


# Matan's Veggie Garden

About



(he, him)



```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    • <div id="theBananaDimension" />

      <Carrot />
      <Carrot />
      <Mushroom />
      <...etc>

    </template>
  </BasePlace>
</template>
```

```
<style lang="scss">
.MatansVeggieGarden {
  .Banana {
    background: url('tomato.png') no-repeat;
    background-size: contain;
    width: 80px; // Tomato width
    padding-left: 80px; // Push banana out of view
  }
}
</style>
```

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```

```
<style lang="scss">
.MatansVeggieGarden {
  • .Banana {
    background: url('tomato.png') no-repeat;
    background-size: contain;
    width: 80px; // Tomato width
    padding-left: 80px; // Push banana out of view
  }
}
</style>
```



## VJasonSpace.vue

```

<template>
<BasePlace placeName="JasonSpace">
  <template v-slot:about>
    
    <p>(he, him)</p>

    <button @click="sendBanana">
      Send Banana!
    </button>
  </template>

  <template v-slot:default>
    <...etc>

    <teleport
      v-if="bananas"
      disabled
      to="#theBananaDimension"
    >
      <Banana v-for="banana in bananas" />
    </teleport>
  </template>
</BasePlace>
</template>

```

## VMatansVeggieGarden.vue

```

<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>

<style lang="scss">
.MatansVeggieGarden {
  .Banana {
    // ...etc
  }
}
</style>

```

# JasonSpace

About



(he, him)



# Matan's Veggie Garden

About



(he, him)



## VJasonSpace.vue

```
<script>
export default {
  data: _ => ({
    bananas: 0
  }),

  methods: {
    sendBanana() {
      ++this.bananas
    }
  }
}
</script>

<template>
<BasePlace placeName="JasonSpace">
  <...etc>

  <teleport
    v-if="bananas"
    to="#theBananaDimension"
  >
    <Banana v-for="banana in bananas" />
  </teleport>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```

## VJasonSpace.vue

```
<script>
export default {
  data: _ => ({
    ● bananas: 0
  }),

  methods: {
    sendBanana() {
      ++this.bananas
    }
  }
}
</script>

<template>
<BasePlace placeName="JasonSpace">
  <...etc>

  <teleport
    v-if="bananas"
    to="#theBananaDimension"
  >
    <Banana v-for="banana in bananas" />
  </teleport>
```

## VMatansVeggieGarden.vue

```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```



## VJasonSpace.vue

```
<script>
export default {
  data: _ => ({
    bananas: 1
  }),

  methods: {
    sendBanana() {
      ++this.bananas
    }
  }
}
</script>

<template>
<BasePlace placeName="JasonSpace">
  <...etc>

  <teleport
    v-if="bananas"
    to="#theBananaDimension"
  >
    <Banana v-for="banana in bananas" />
  </teleport>
```

## VMatansVeggieGarden.vue

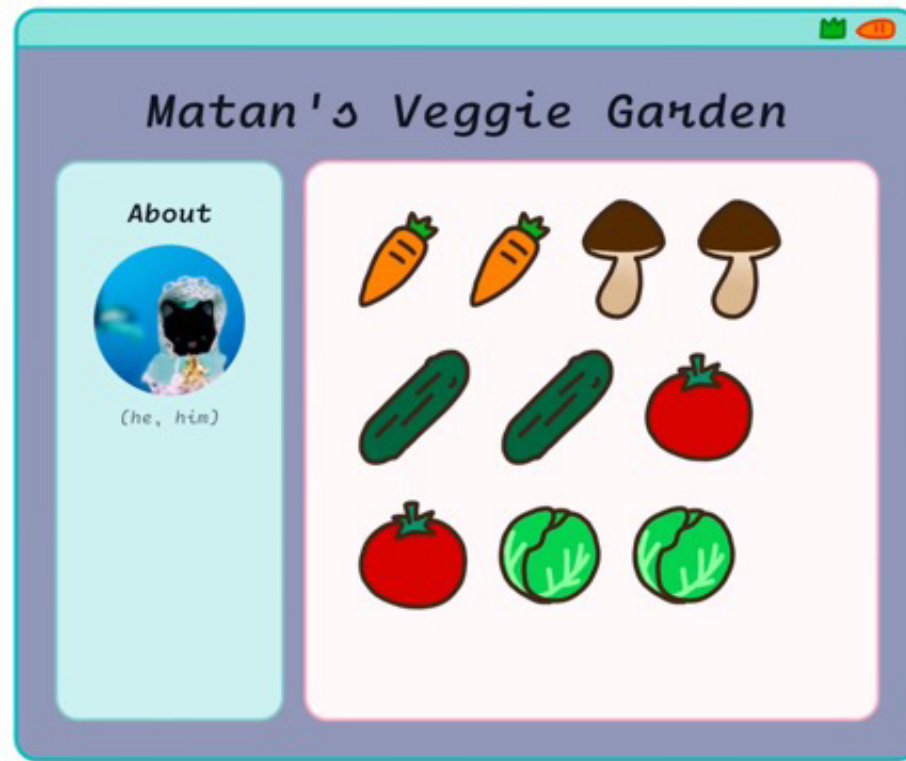
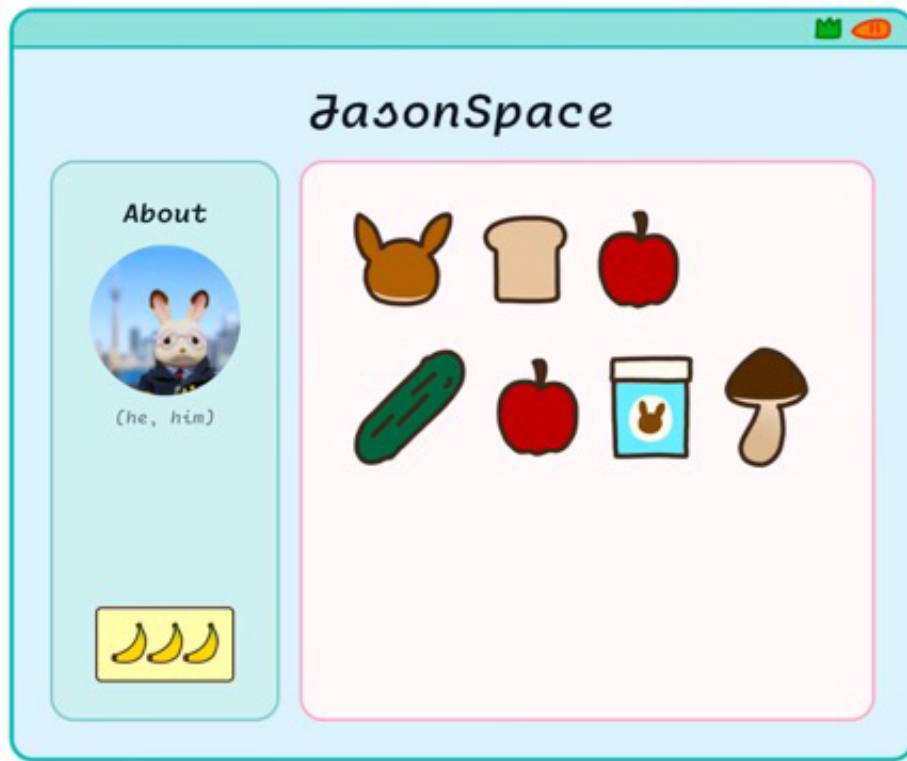
```
<template>
<BasePlace placeName="Matan's Veggie Garden">
  <template v-slot:about>
    
    <p>(he, him)</p>
  </template>

  <template v-slot:default>
    <div id="theBananaDimension" />

    <Carrot />
    <Carrot />
    <Mushroom />
    <...etc>

  </template>
</BasePlace>
</template>
```





Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Applicati

Filter Output

[HMR] Waiting for update signal from WDS...

ⓘ You are running a development build of Vue.  
Make sure to use the production build (\*.prod.js) when deploying for production.

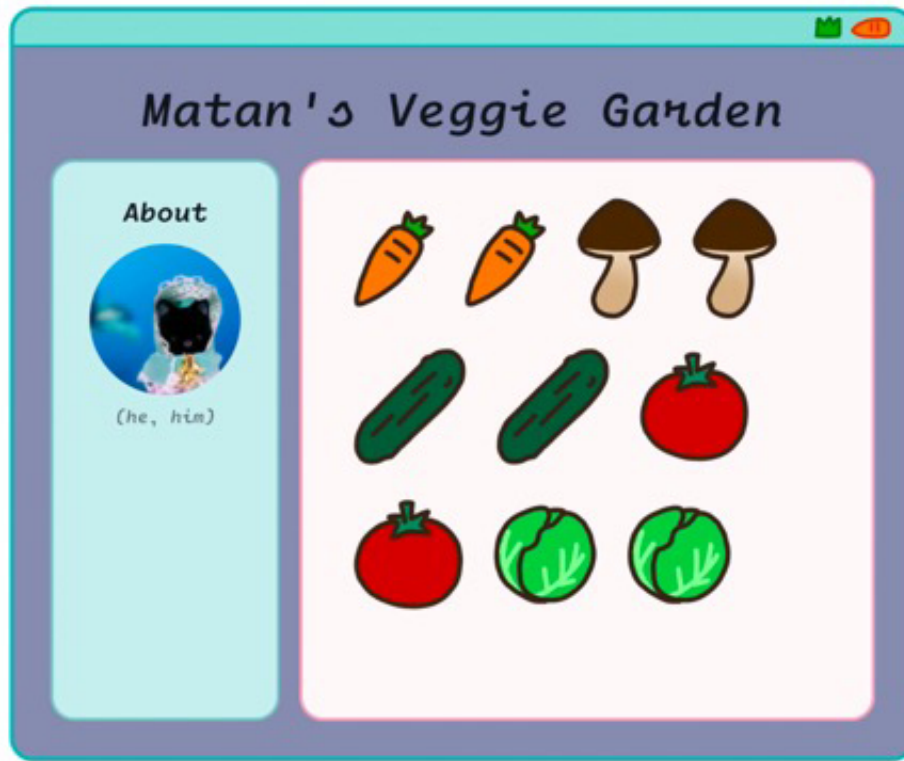
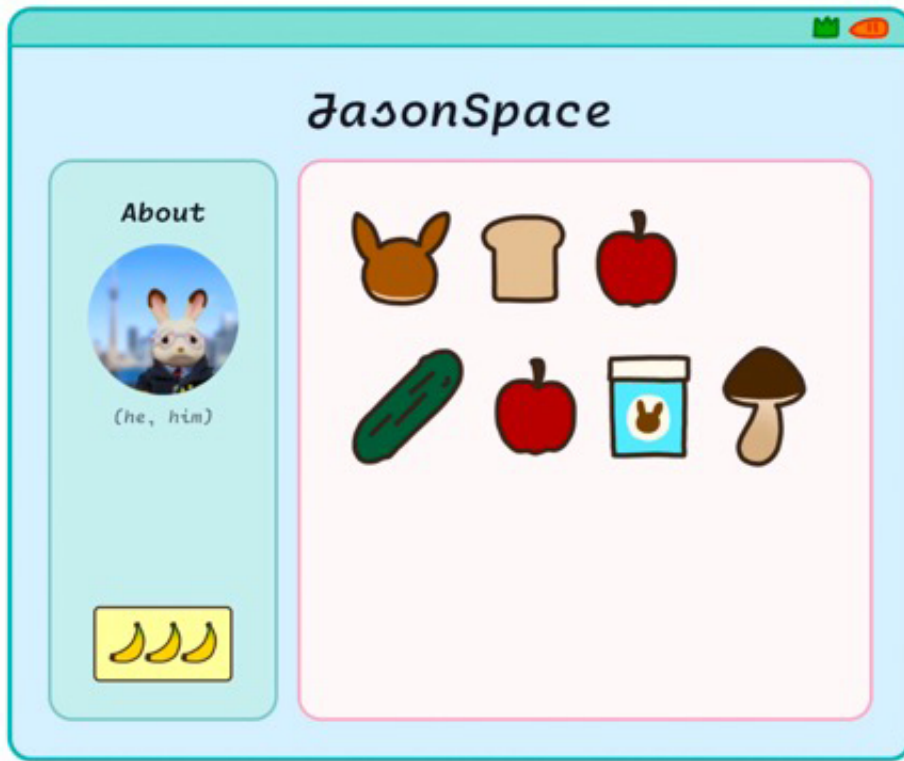
⚠ [Vue warn]: Failed to locate Teleport target with selector "#theBananaDimension". Note the target element must exist before the component is mounted, and cannot be rendered by the component itself, and ideally should be outside of the entire Vue component tree.

```
at <BasePlace placeName="JasonSpace" >  
at <BaseWindow class="VJasonsPlace" >  
at <VMultiPlace key=2 >  
at <App>
```

⚠ [Vue warn]: Invalid Teleport target on mount: null (object)

```
at <BasePlace placeName="JasonSpace" >  
at <BaseWindow class="VJasonsPlace" >  
at <VMultiPlace key=2 >  
at <App>
```

>>



Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Applicati

Filter Output

[Vue warn]: Unhandled error during execution of scheduler flush. This is likely a Vue internals bug. Please open an issue at <https://new-iss>  
next  
at <BasePlace placeName="JasonSpace" >  
at <BaseWindow class="VJasonsPlace" >  
at <VMultiPlace key=2 >  
at <App>

Uncaught (in promise) TypeError: can't access property "parentNode", node is null  
parentNode runtime-dom.esm-bundler.js:30  
patchBlockChildren runtime-core.esm-bundler.js:4292  
process runtime-core.esm-bundler.js:1517  
patch runtime-core.esm-bundler.js:3994  
patchBlockChildren runtime-core.esm-bundler.js:4296  
processFragment runtime-core.esm-bundler.js:4347  
patch runtime-core.esm-bundler.js:3984  
patchBlockChildren runtime-core.esm-bundler.js:4296  
processFragment runtime-core.esm-bundler.js:4347  
patch runtime-core.esm-bundler.js:3984

when to go bananae

- Subverts the DOM
- Places content regardless of genealogy
- Sends node to static destination
- Potential use cases for teleports:
  - *Toasts*
  - *Tooltips*
  - *Modals*
- Injected content appended to initial



Manish's Room

request pancake



Bobby's Room

request pancake



Em's Room

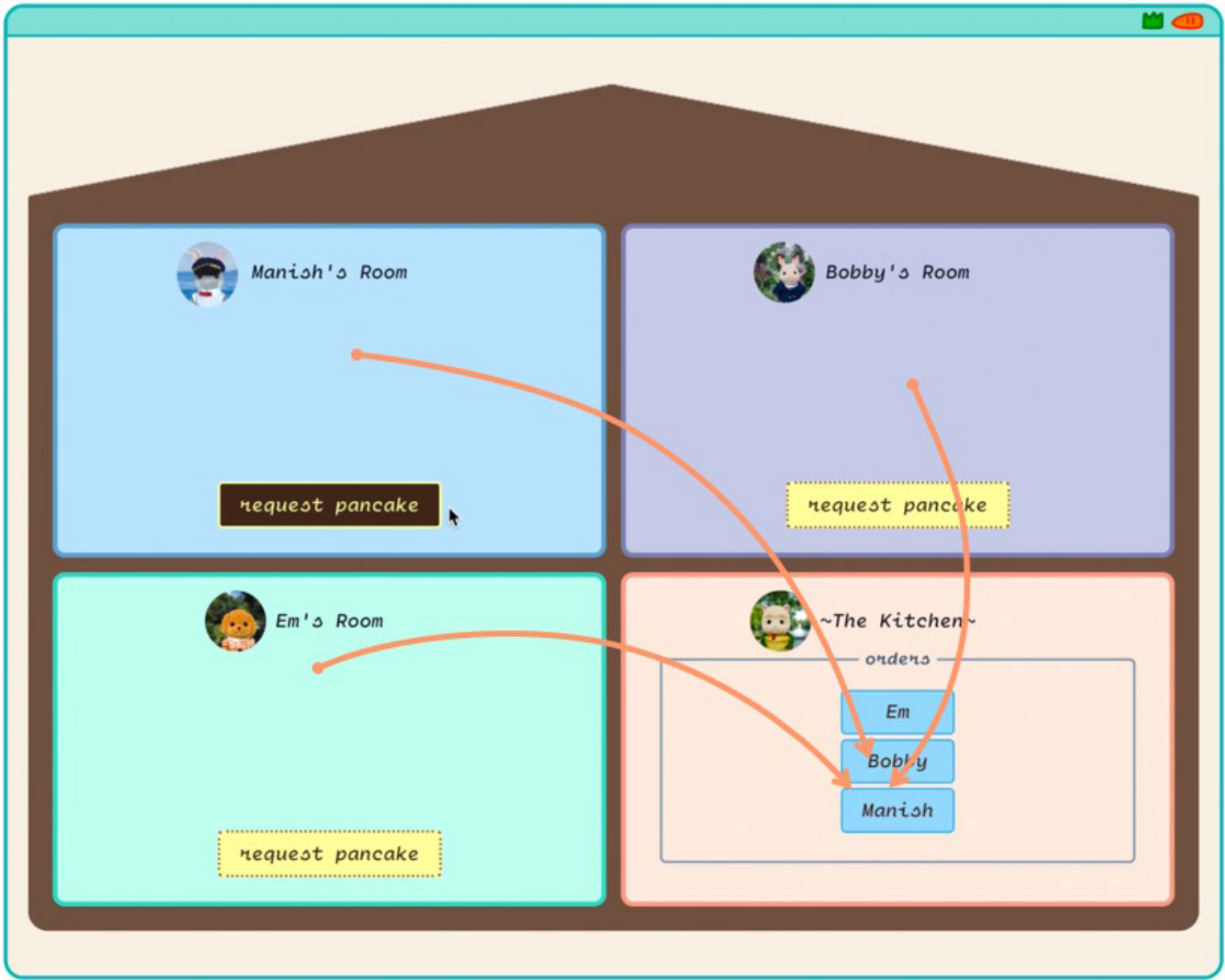
request pancake



~The Kitchen~

orders





VPancakeHouse.vue

```
<template>
<RoomManish />
<RoomBobby />
<RoomEm />
<RoomKitchen />
</template>
```

RoomKitchen.vue

```
<template>
<h1>~The Kitchen~</h1>

<h2>orders</h2>

<div id="foodOrders" />
</template>
```

## BaseRoom.vue

```
<template>
<h1>{{ roomOwner }}'s Room</h1>

<Pancake v-for="pancake in pancakesServed" />

<button @click="requestPancake">
  request pancake
</button>

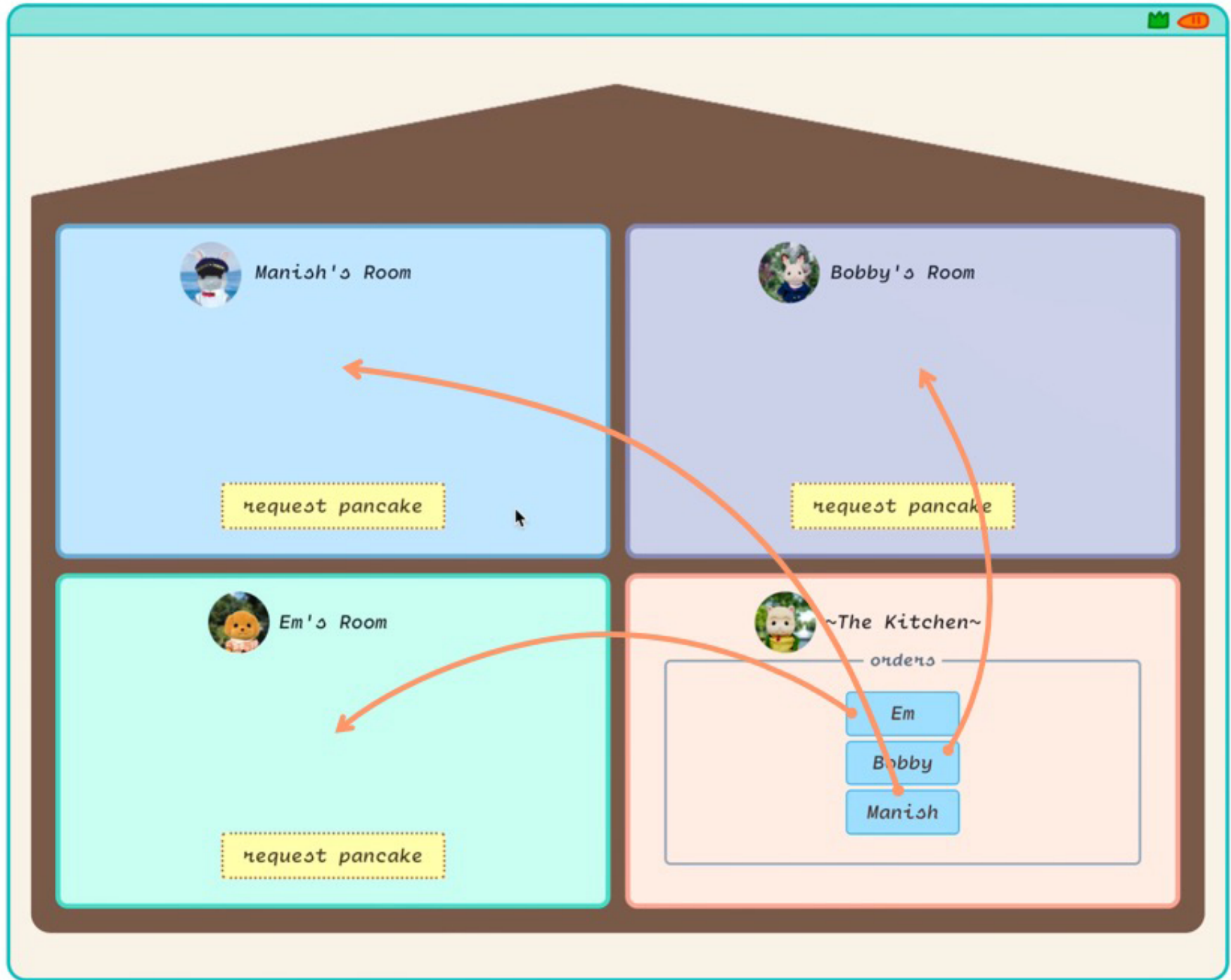
<teleport
  v-if="pancakeRequested"
  to="#foodOrders"
>
  <button>{{ roomOwner }}</button>
</teleport>
</template>
```

## RoomKitchen.vue

```
<template>
<h1>~The Kitchen~</h1>

<h2>orders</h2>

<div id="foodOrders" />
</template>
```



## BaseRoom.vue

```
<script>
export default {
  // ...etc

  methods: {
    requestPancake() { // ...etc },

    servePancake() {
      this.pancakeRequested = false
      this.pancakesServed++
    },
  },
}
</script>

<template>
<...etc>

<teleport
  v-if="pancakeRequested"
  to="#foodOrders"
>
• <button @click.stop="servePancake">
  {{ roomOwner }}
</button>
```

## RoomKitchen.vue

```
<template>
<h1>~The Kitchen~</h1>

<h2>orders</h2>

<div id="foodOrders" />
</template>
```





Manish's Room



request pancake



Bobby's Room



request pancake



Em's Room



request pancake



~The Kitchen~

orders

A large empty rectangular box with a thin blue border, intended for listing orders. A mouse cursor is visible inside the box.

RoomKitchen.vue

```
<template>  
<h1 @click="bonusPancake">~The Kitchen~</h1>
```

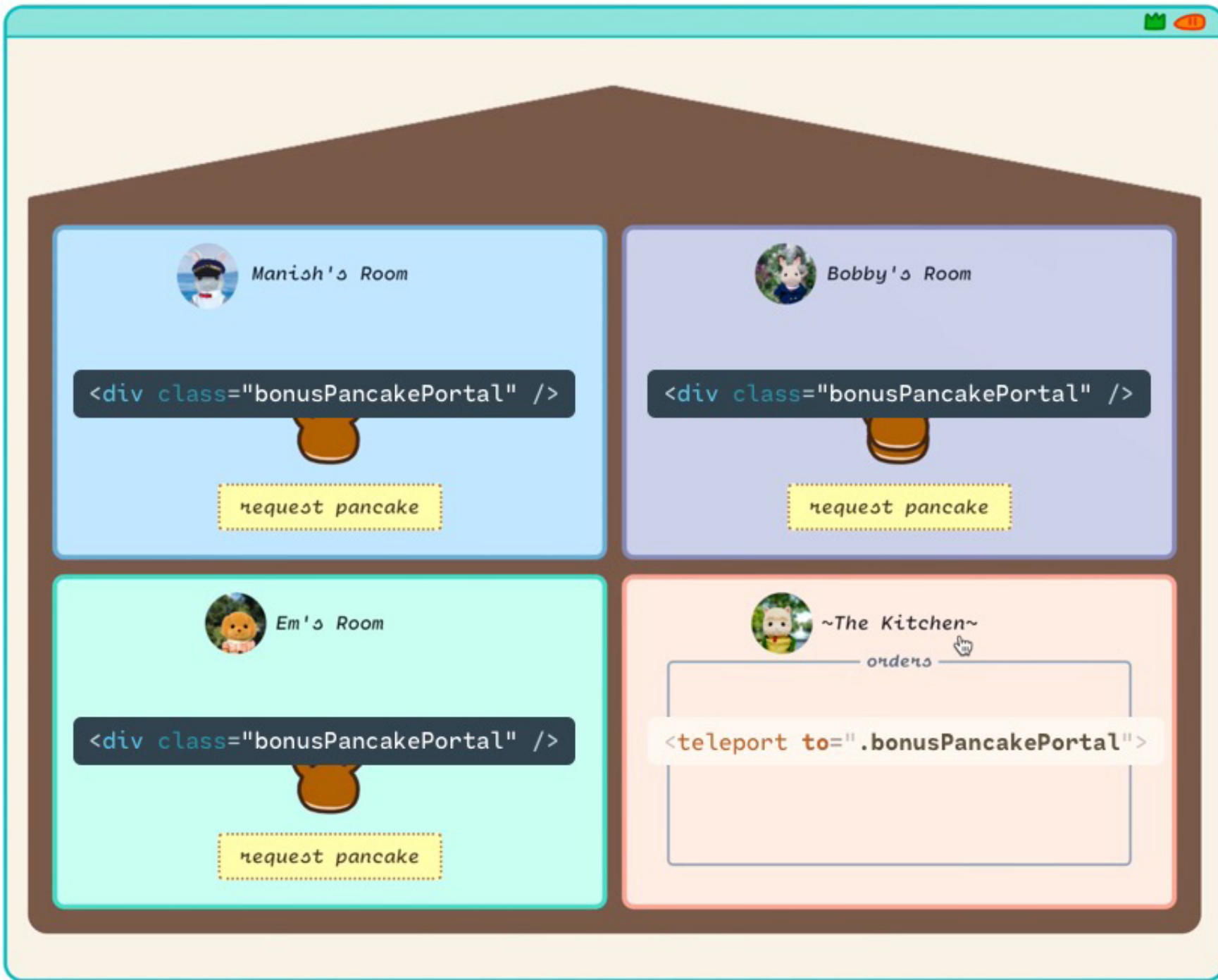
```
• <teleport  
  v-if="numBonusPancakes"  
  to=".bonusPancakePortal"  
>  
  <Pancake  
    v-for="bonusPancake in numBonusPancakes"  
  />  
</teleport>
```

```
<h2>orders</h2>
```

```
<div id="foodOrders" />  
</template>
```

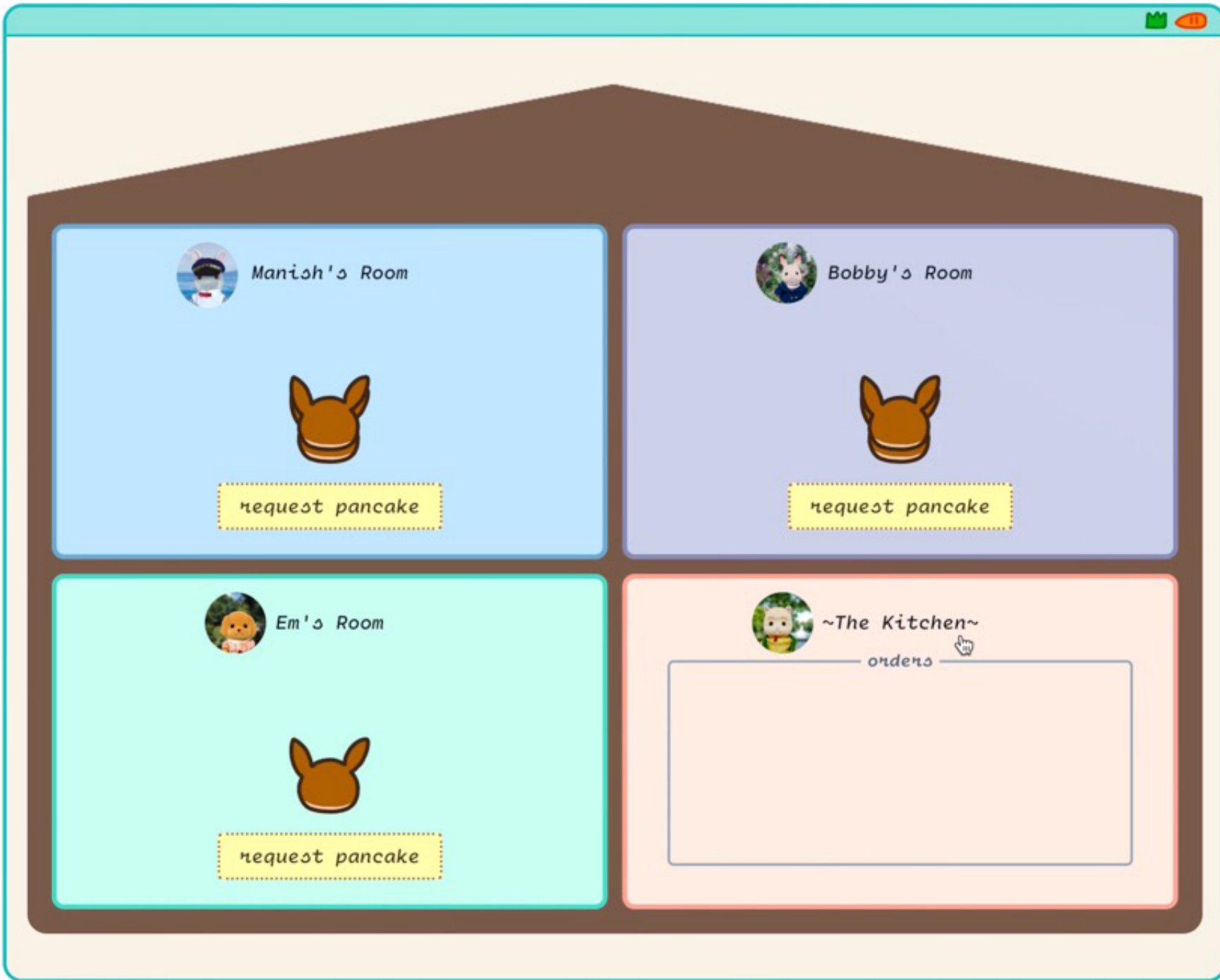
BaseRoom.vue

```
<template>  
<h1>{{ roomOwner }}'s Room</h1>  
  
<div class="bonusPancakePortal" />  
  
<div class="pancakes">  
  <Pancake v-for="pancake in pancakesServed" />  
  <...etc>  
</div>  
</template>
```



VPancakeHouse.vue

```
<template>  
<RoomManish />  
<RoomBobby />  
<RoomEm />  
<RoomKitchen />  
</template>
```



VPancakeHouse.vue

```
<template>  
<RoomManish />  
<RoomBobby />  
<RoomEm />  
<RoomKitchen />  
</template>
```





















## VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

## Laundry.vue

```
<script>
export default {
  props: {
    laundry: Array,
  },
}
</script>

<template>

</template>
```

VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

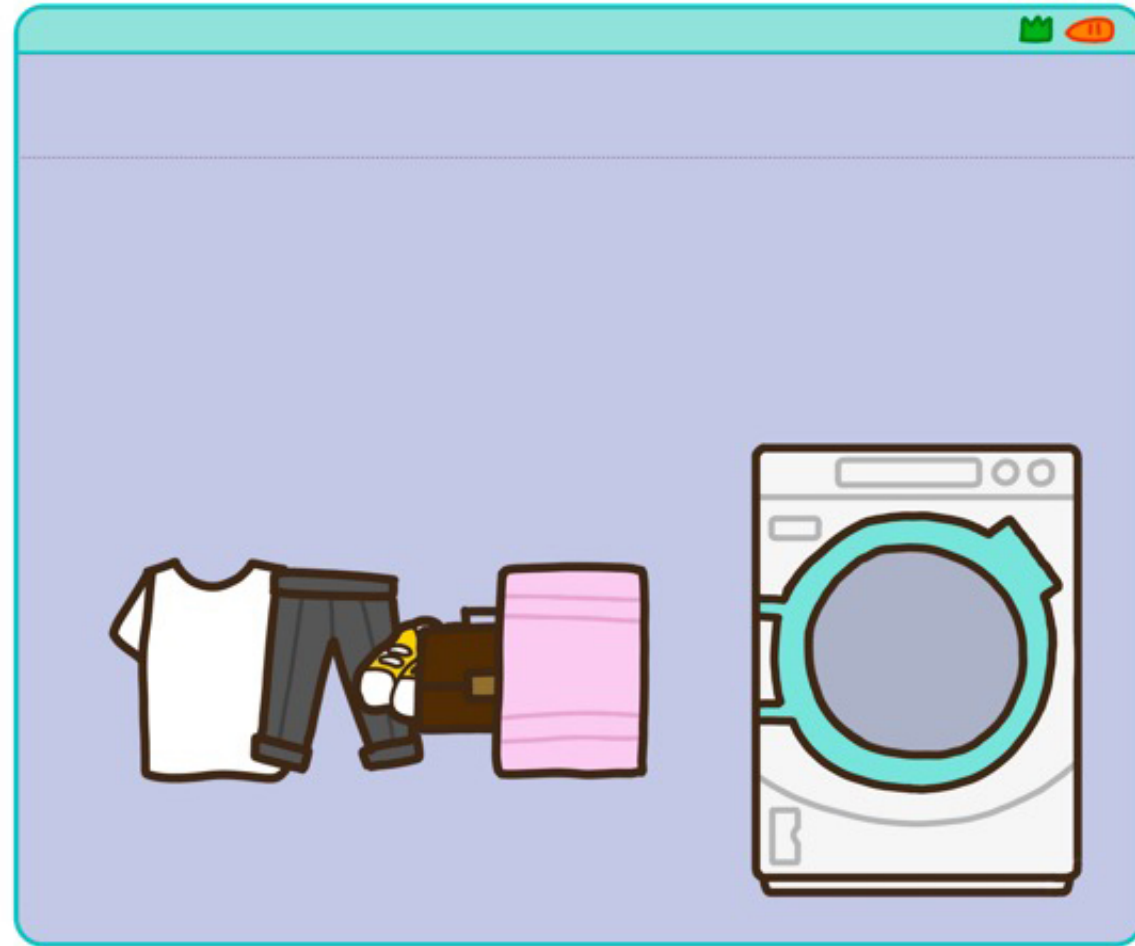
  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

Laundry.vue

```
<script>
export default {
  props: {
    laundry: Array,
  },
}
</script>

<template>

</template>
```



VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

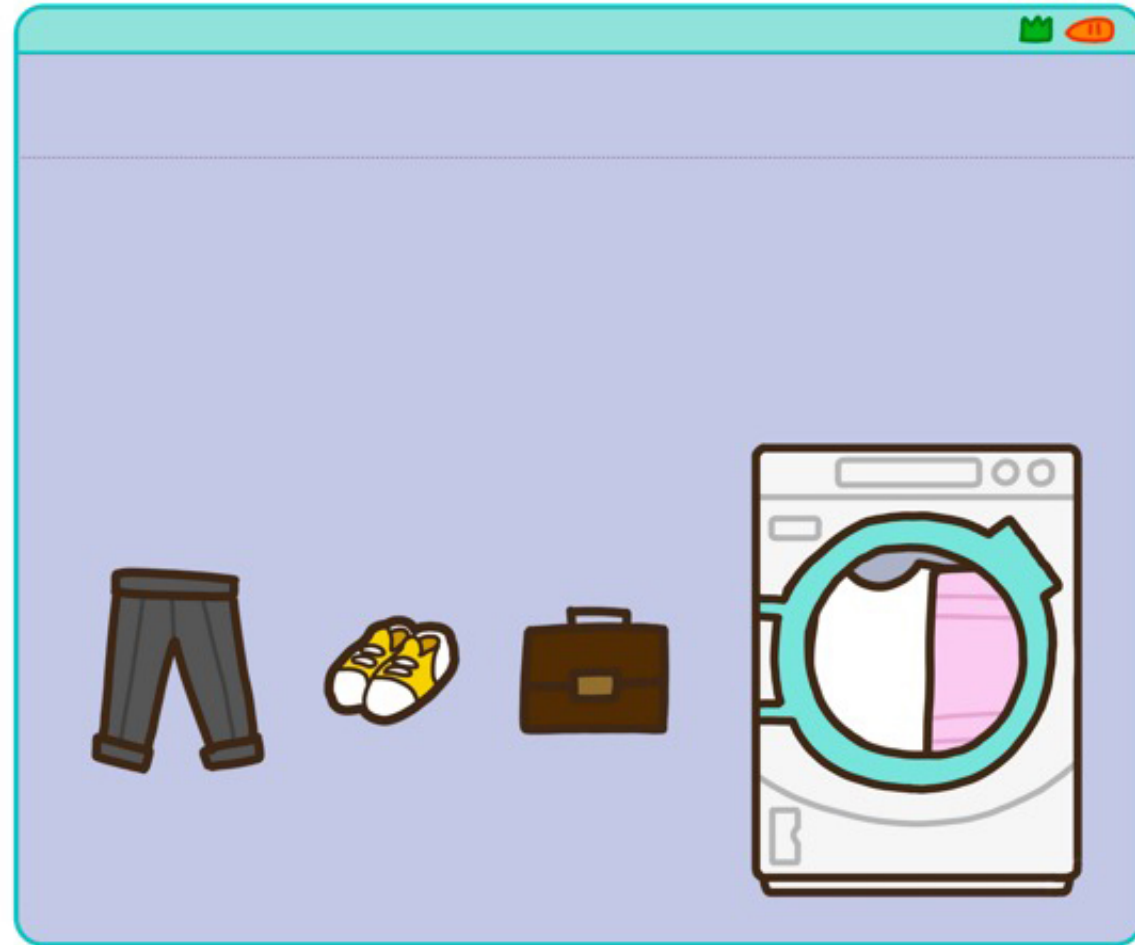
export default {
  components: { Laundry },

  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  

  
</template>
</template>
```





VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

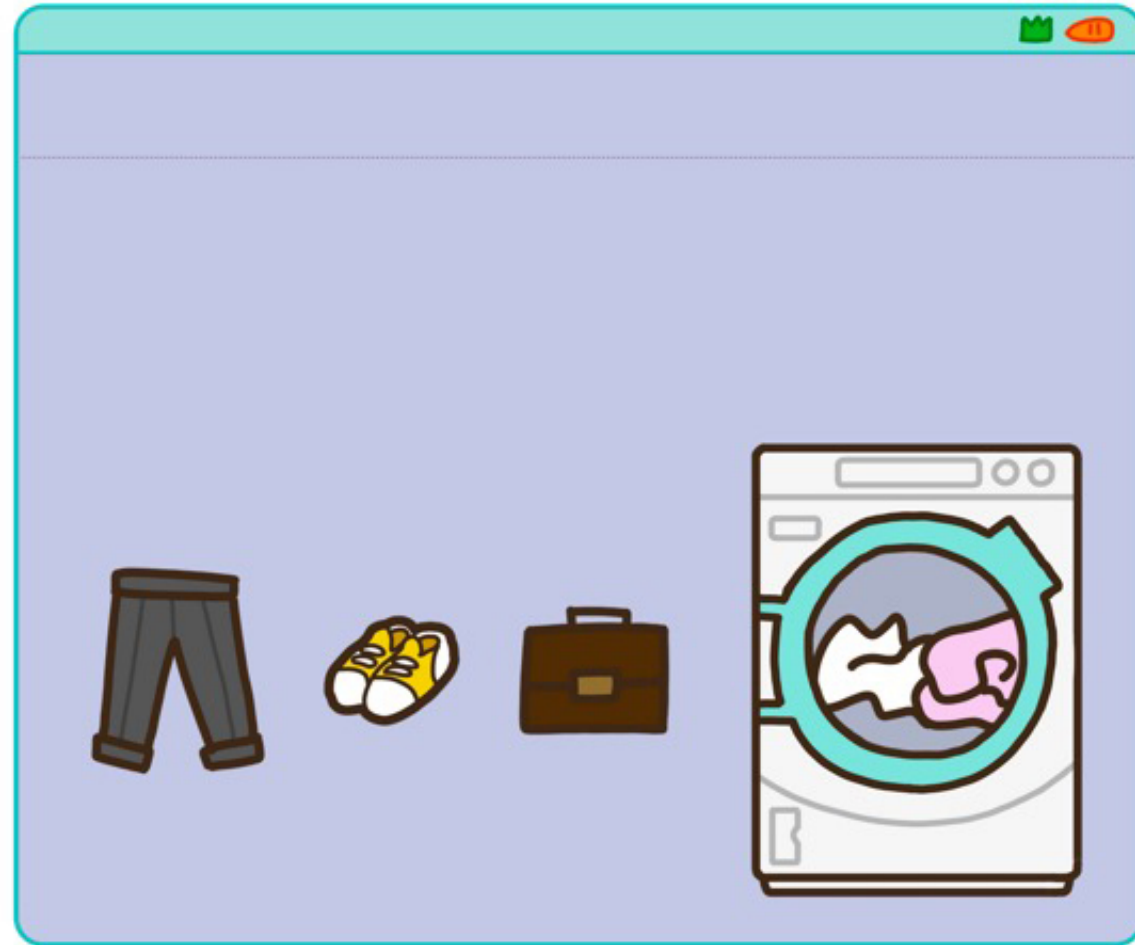
  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  

  • 

  
</template>
</template>
```



## VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

## Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  

  
</template>
```

## VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

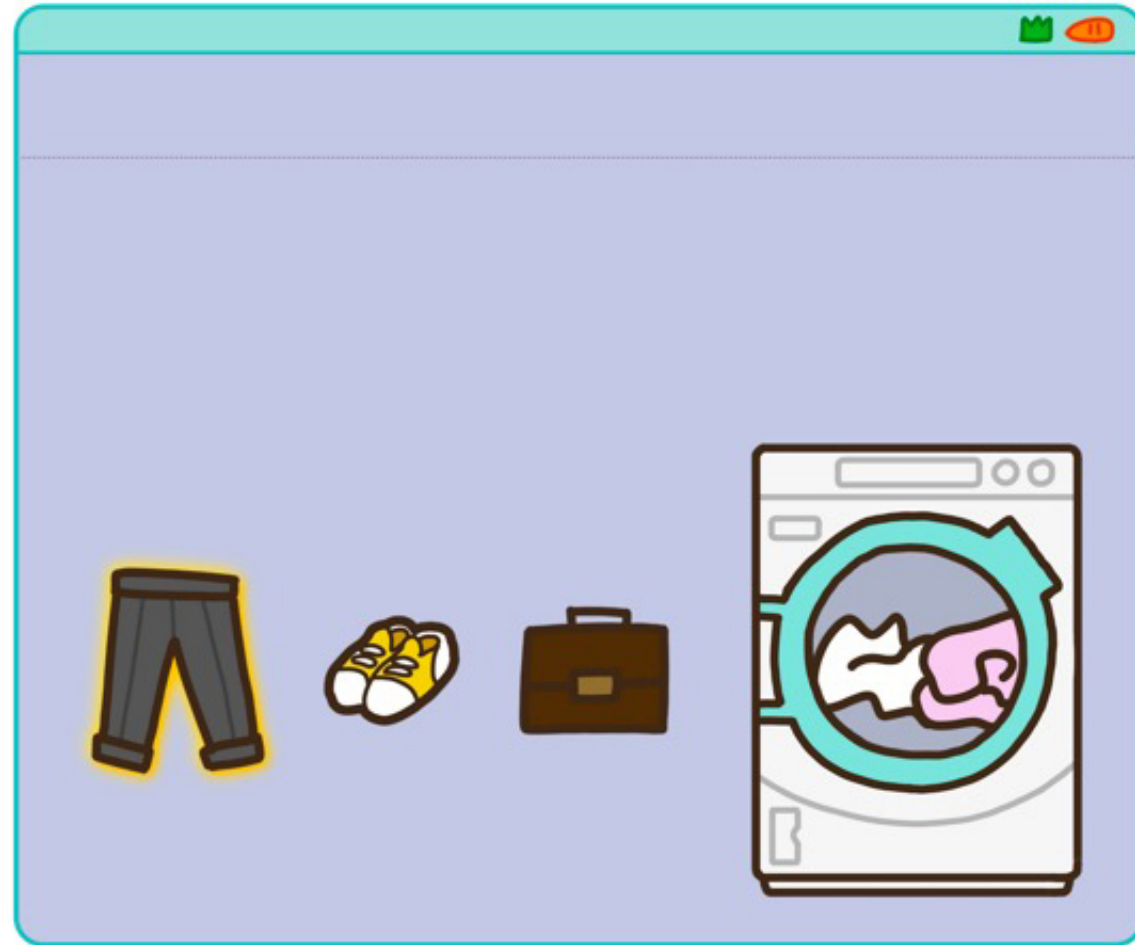
  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

## Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  

  
</template>
```







## VLaundryRoom.vue

```
<script>
import Laundry from './Laundry'

export default {
  components: { Laundry },

  data: _ => ({
    laundry: [
      {
        description: 't-shirt',
        clothingType: 'shirt',
        cleaningInstructions: 'machine wash cold',
      },
      {
        description: 'pants',
        clothingType: 'pants',
        cleaningInstructions: 'dry clean only',
      },
      {
        // ...etc
      }
    ]
  })
}
```

## Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  

  
</template>
```

(. \_ .)











internet

edge

e















slot

props

e



VLaudryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
```

```
• <Laundry :laundry="laundry">
```

```
  <template
```

```
    v-slot:specialCleaning="{ specialCareItem }"
```

```
  >
```

```
    
```

```
    
```

```
  </template>
```

```
<template v-slot="{ machineWashableClothing }">
```

```
<...etc>
```

Laundry.vue

```
<template>
```

```
<template v-for="clothingItem in laundry">
```

```
  <slot
```

```
    name="specialCleaning"
```

```
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
```

```
    v-bind:specialCareItem="clothingItem"
```

```
  >
```

```
    
```

```
  </slot>
```

```
</template>
```

```
<template v-for="clothingItem in laundry">
```

```
  <slot
```

```
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
```

```
    v-bind:machineWashableClothing="clothingItem"
```

```
  >
```

```
    
```

```
  </slot>
```

```
</template>
```

```
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
  • v-slot:specialCleaning="slotProps"
  >
    

    
  </template>

  <template v-slot="slotProps">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaudryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```



VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

VLaundryRoom.vue

```
<template>
<Laundry :laundry="laundry">
  <template
    v-slot:specialCleaning="{ specialCareItem }"
  >
    

    
  </template>

  <template v-slot="{ machineWashableClothing }">
    <...etc>
```

Laundry.vue

```
<template>
<template v-for="clothingItem in laundry">
  <slot
    name="specialCleaning"
    v-if="!isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:specialCareItem="clothingItem"
  >
    
  </slot>
</template>

<template v-for="clothingItem in laundry">
  <slot
    v-if="isMachineWashable(
      clothingItem.cleaningInstructions)"
    v-bind:machineWashableClothing="clothingItem"
  >
    
  </slot>
</template>
</template>
```

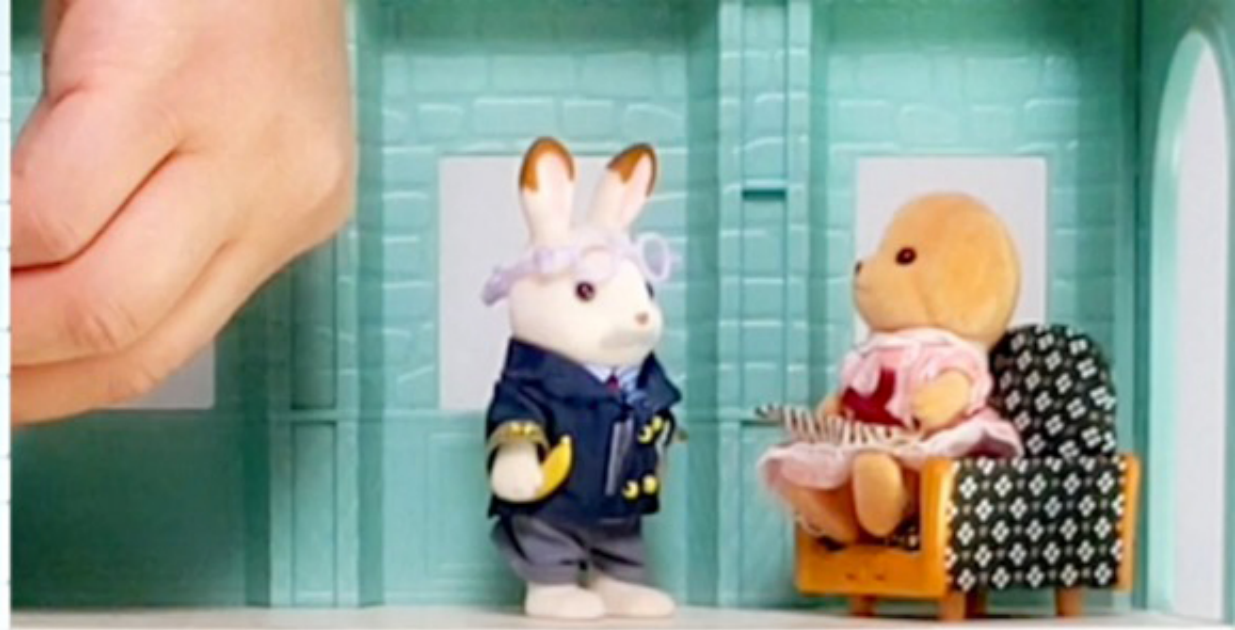






(re-eh-eh-ed, re-eh-eh-ed)

- Overlapping visual & functional content
- Child owns data and context
- Parent-child protocol needed
- Flexible reusable component
- Potential use cases for scoped slots:
  - *Lists*
  - *Overpowered UI elements*
- Injected content replaces initial



(will! it! nest!)



(yes)

- **Slots: yes**
- **Scoped slots**
  - *# directive shorthand may not always work*
- **Teleport**
  - *Origins can nest other slots*
  - *Destinations can be nested*
  - *Destinations cannot nest other slots\**

recap

- **Slots (general)**
  - Parent controls visual content
  - Child controls framing
- **Named slots**
  - Categorize/group injected content
- **Scoped slots**
  - Child owns data and framing
  - Good for altering appearance
  - Injected data + injected template



- **Teleport**

- Subverts DOM tree
- Destinations are dependable
- Destination must exist before origin

- **\$slots**

- Access slot content in script tag
- **Not** reactive
- Different shape in Vue 3

### Michael Thiessen: 6 Levels of Reusability

<https://michaelnthiessen.com/6-levels-of-reusability>

### Enjoy the Vue: Navigating Scoped Slots

<https://enjoythevue.io/episodes/31>

### Enjoy the Vue: Teleportals!

*(coming soon)*

### Documentation

<https://vuejs.org/v2/guide/components-slots>

<https://v3.vuejs.org/guide/teleport>

### Portal-Vue

<https://portal-vue.linusb.org>



*thank you  
for listening*

punching holes in the sky  
mutating the dom via wrinkles in space

