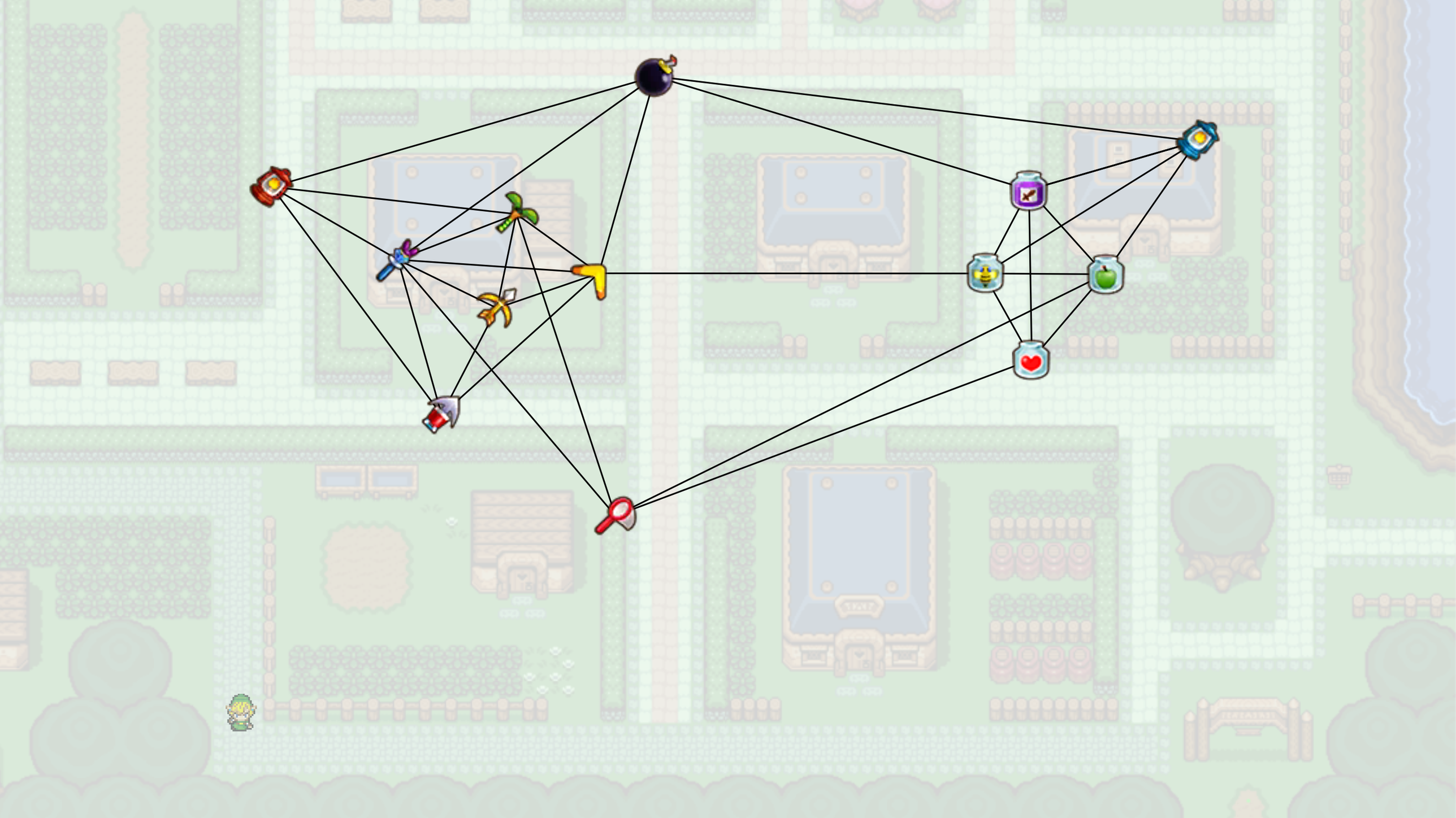
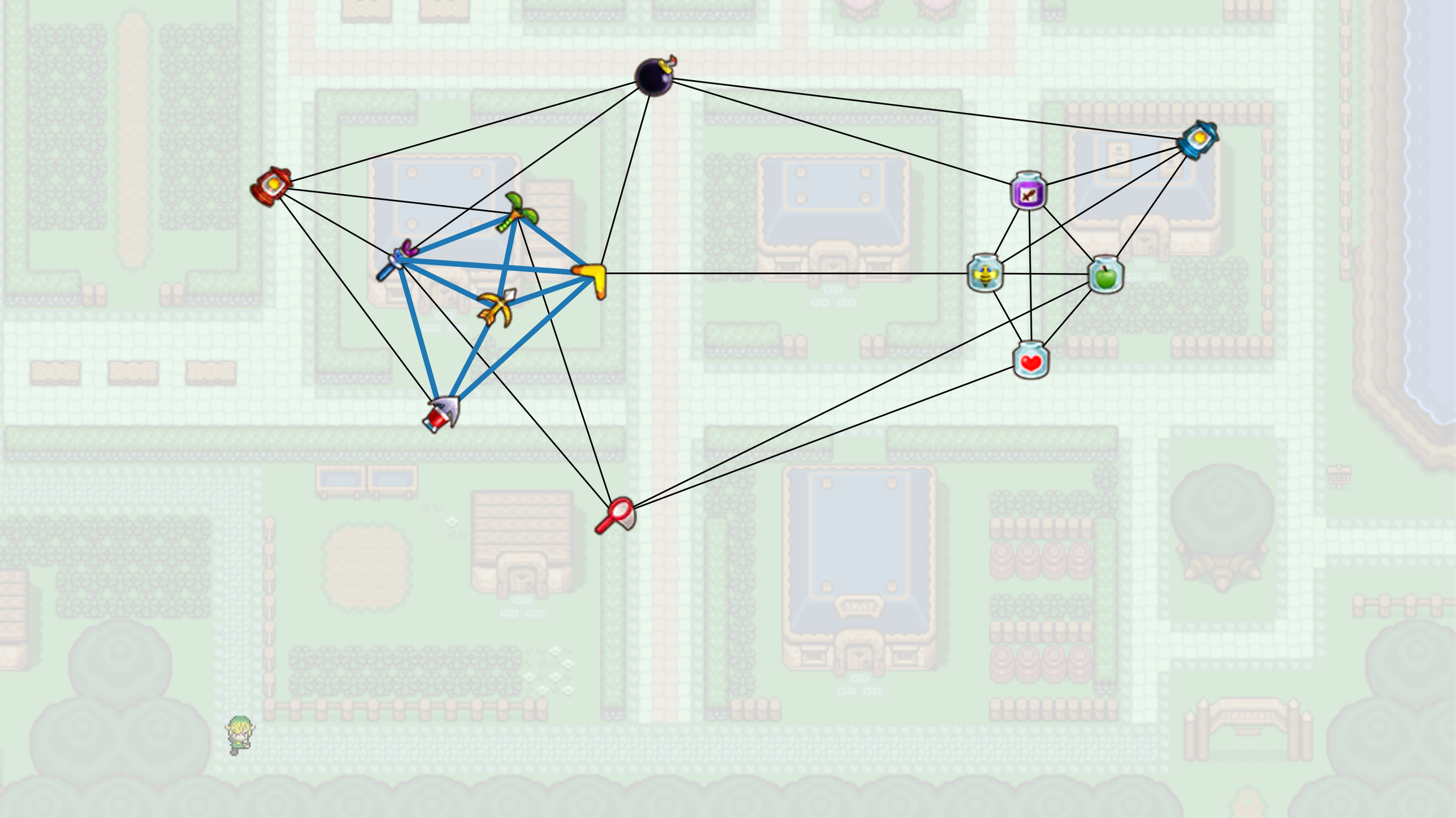
UNIVERSITÄT TRIER

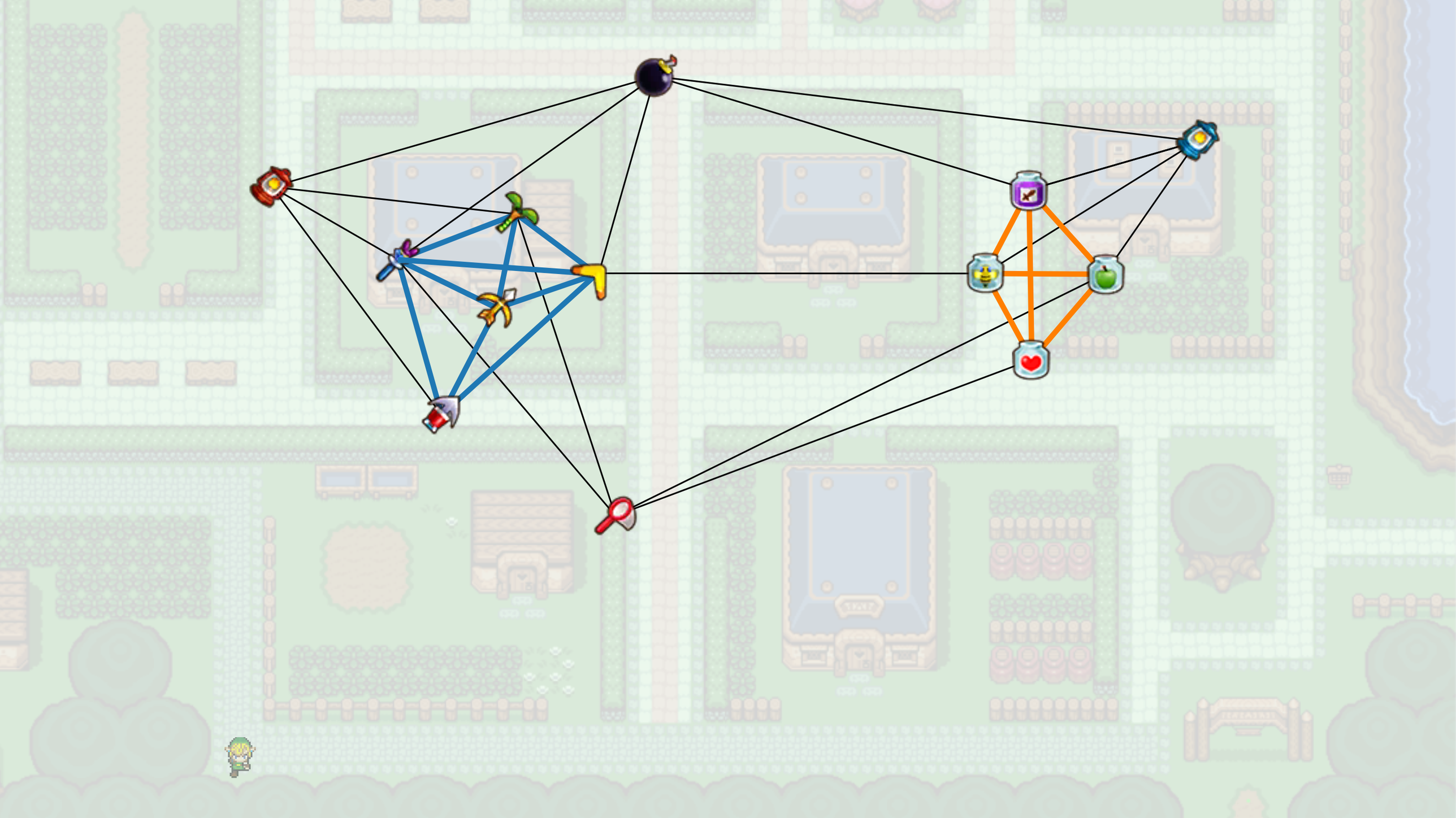
Julius-Maximilians-UNIVERSITÄT WÜRZBURG

The Computational Complexity of the ChordLink Model

Philipp Kindermann
Jan Sauer Alexander Wolff

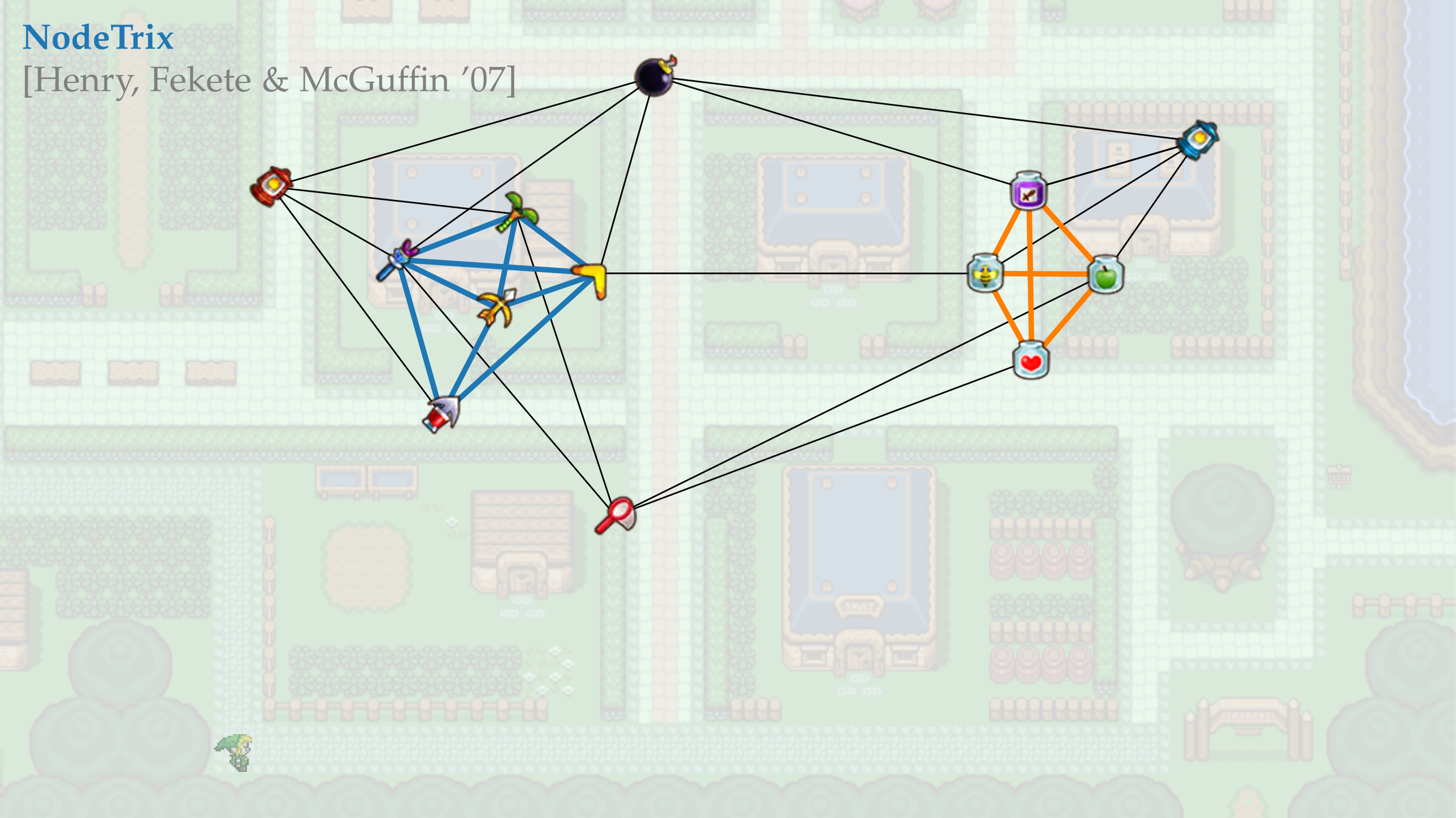






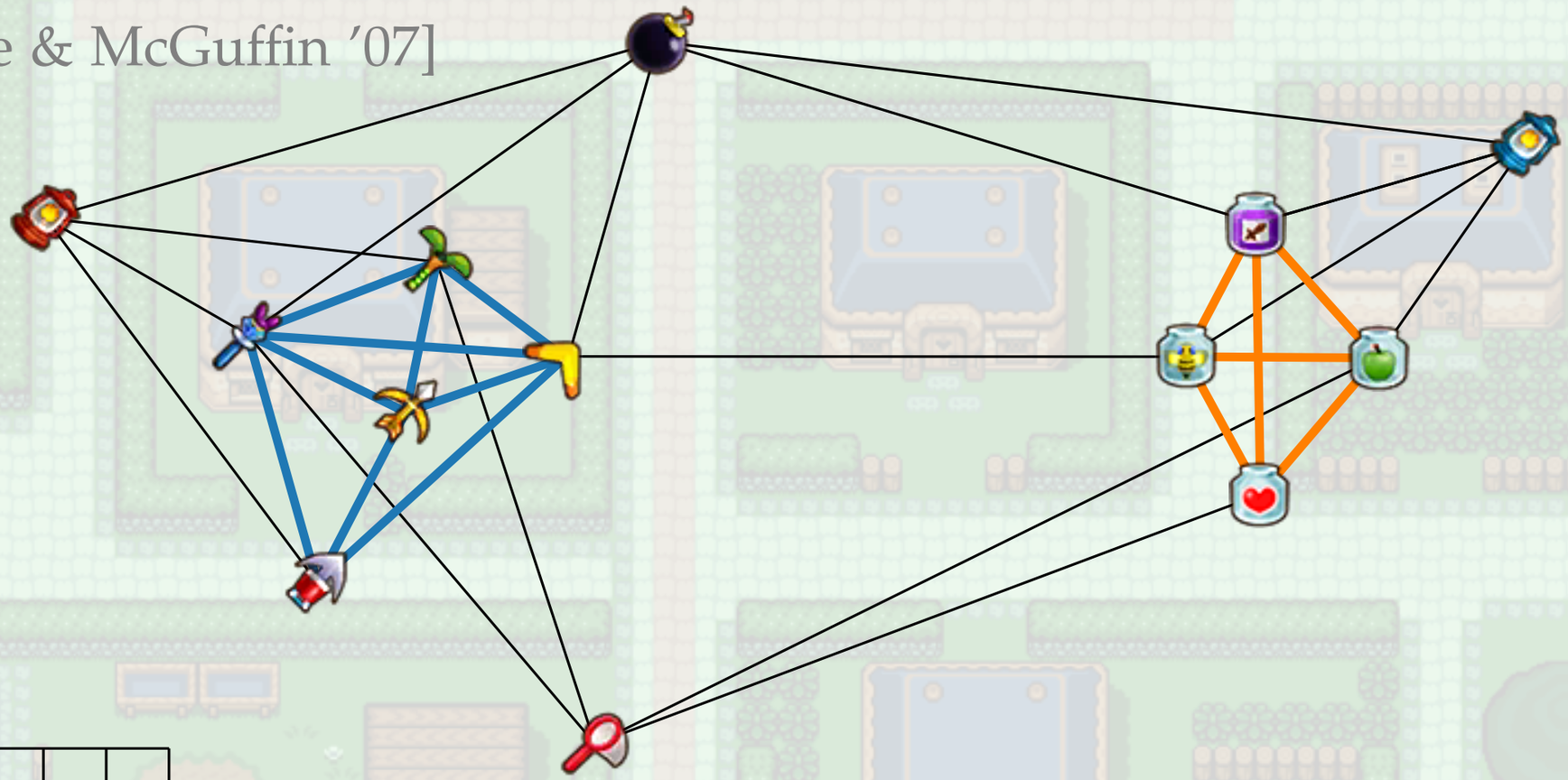
NodeTrix

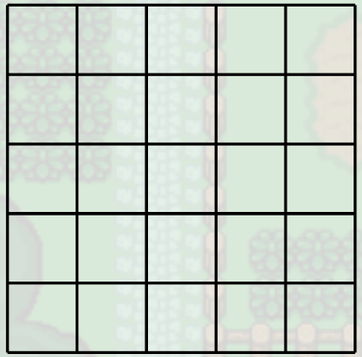
[Henry, Fekete & McGuffin '07]



NodeTrix

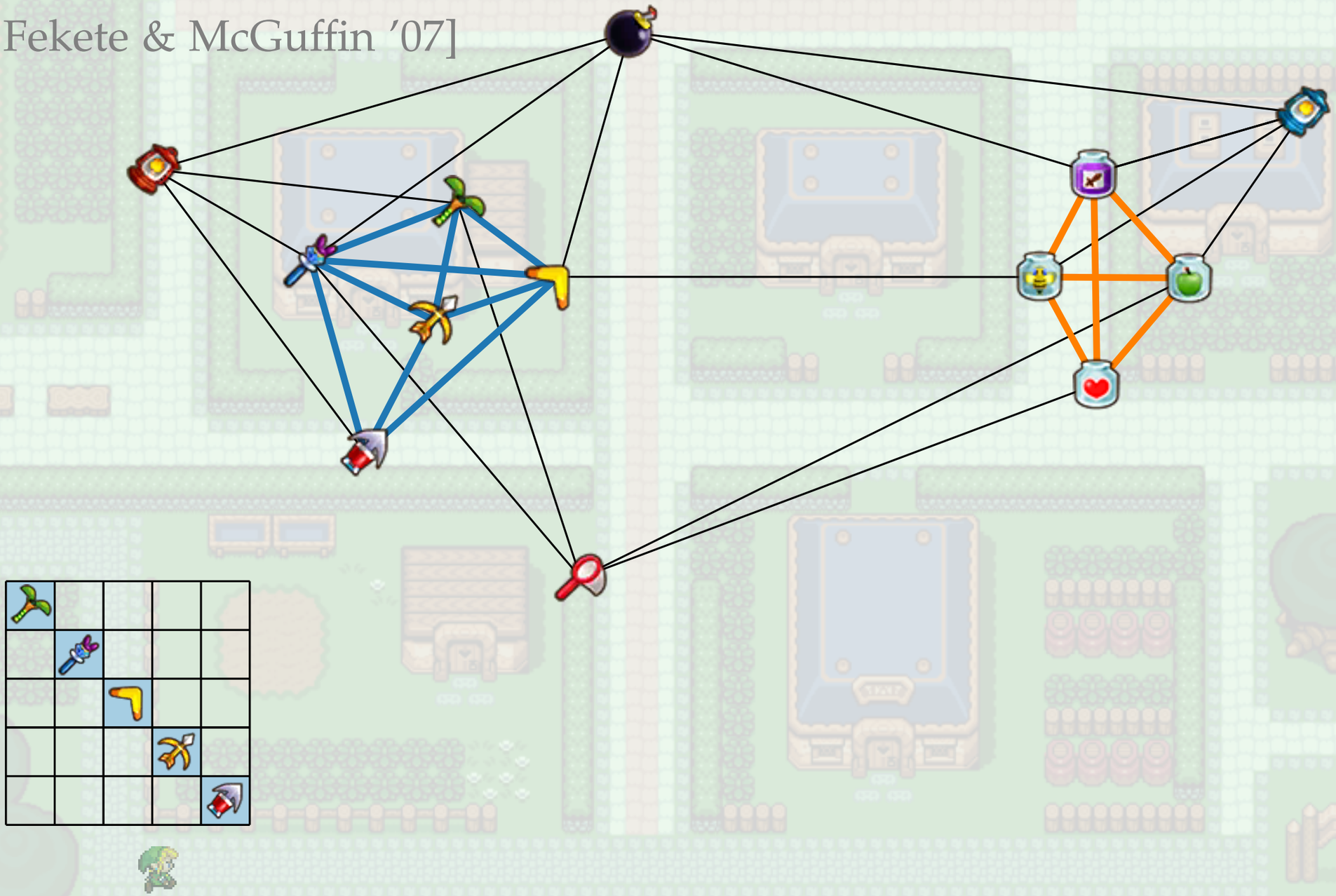
[Henry, Fekete & McGuffin '07]





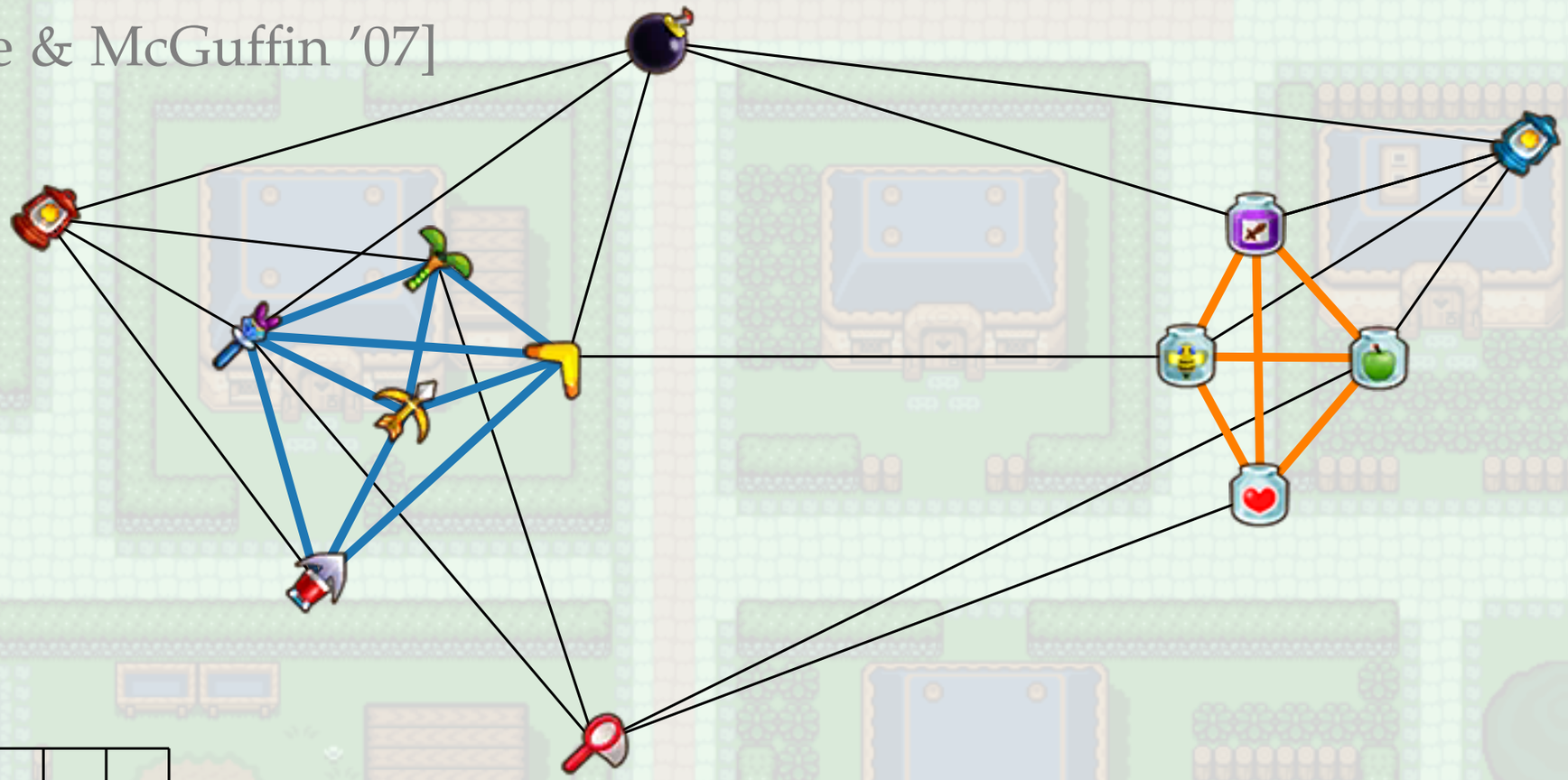
NodeTrix

[Henry, Fekete & McGuffin '07]



NodeTrix

[Henry, Fekete & McGuffin '07]

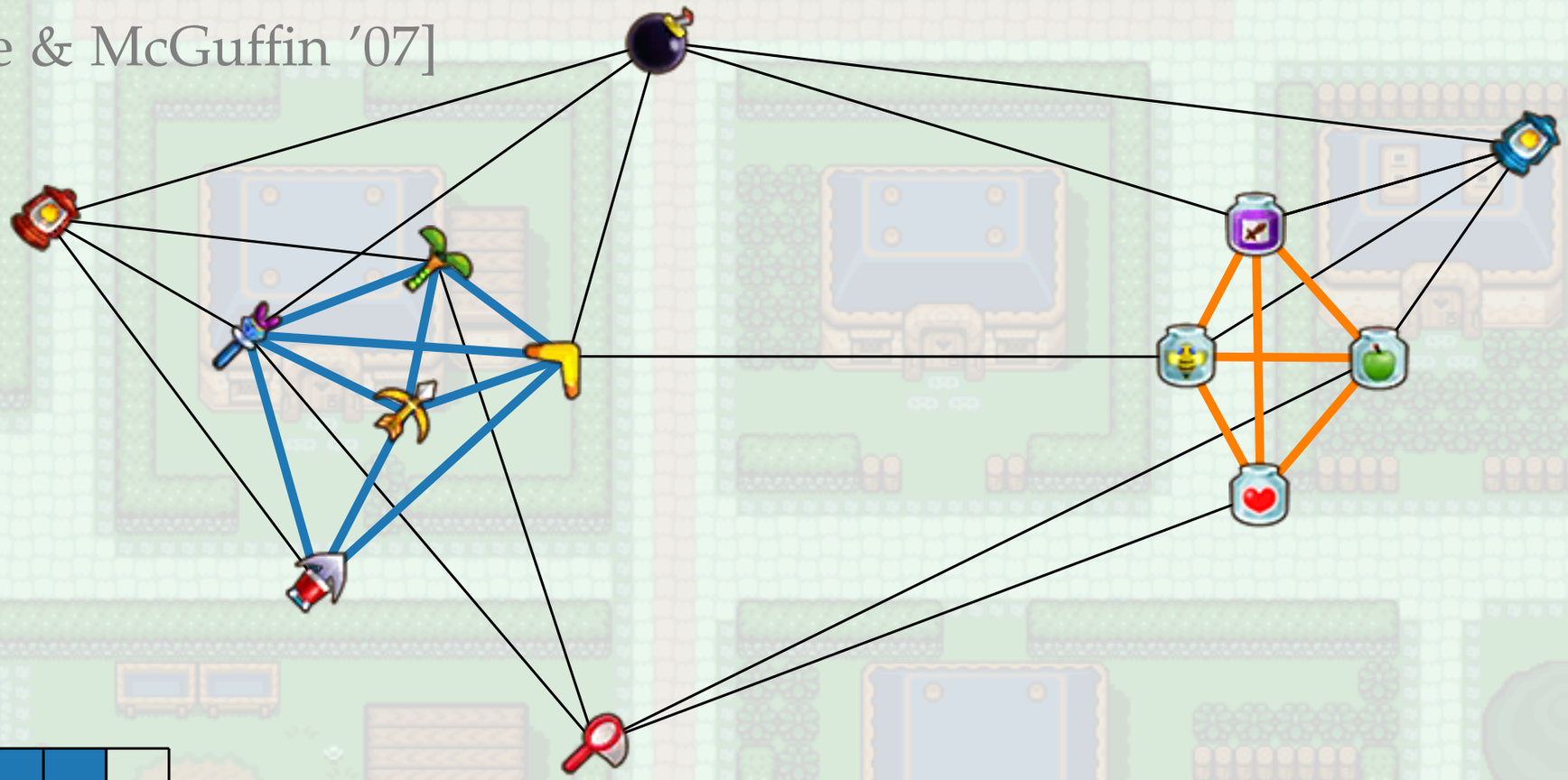


NodeTrix

[Henry, Fekete & McGuffin '07]

NodeTrix

[Henry, Fekete & McGuffin '07]

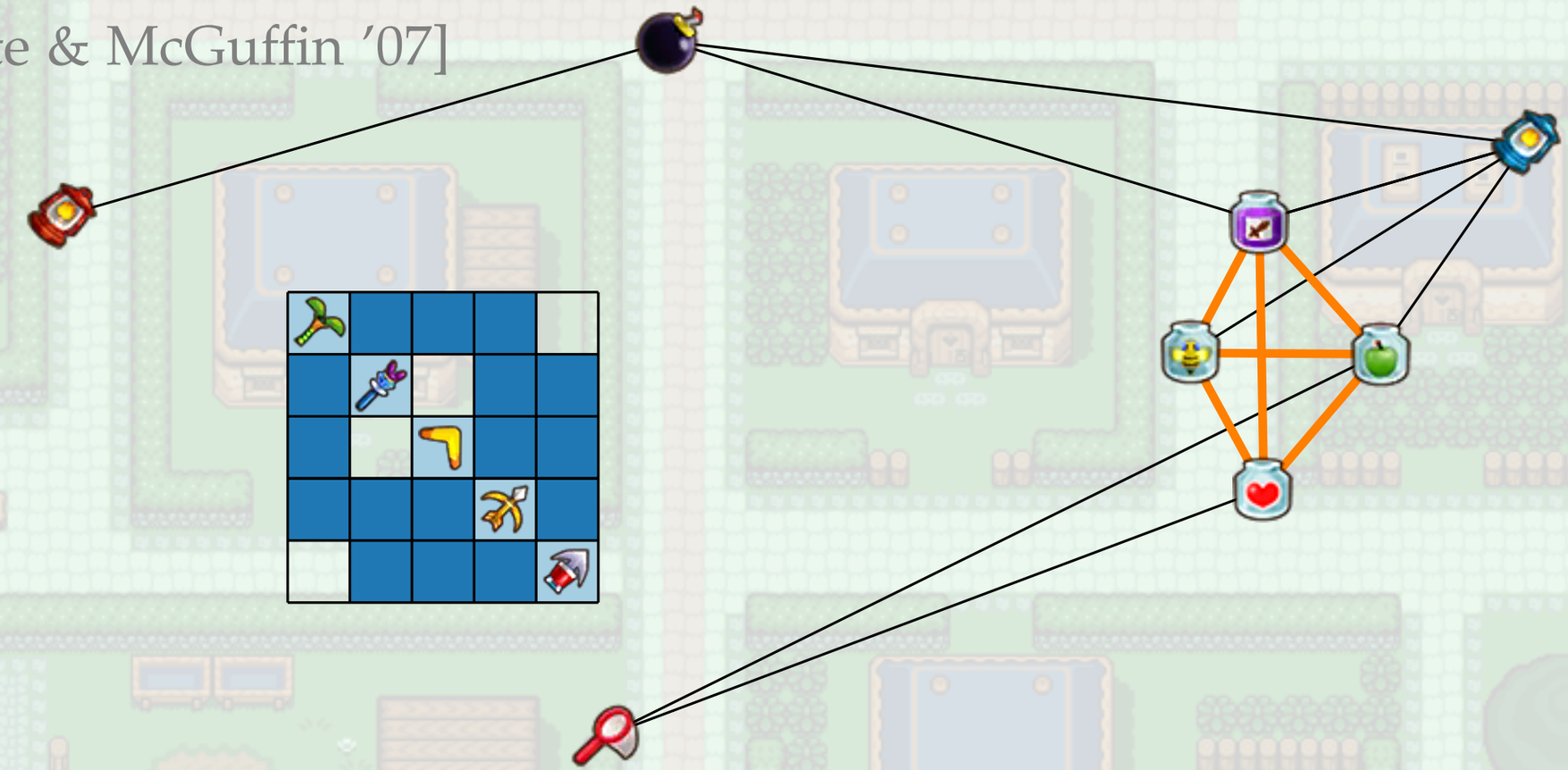


NodeTrix

[Henry, Fekete & McGuffin '07]

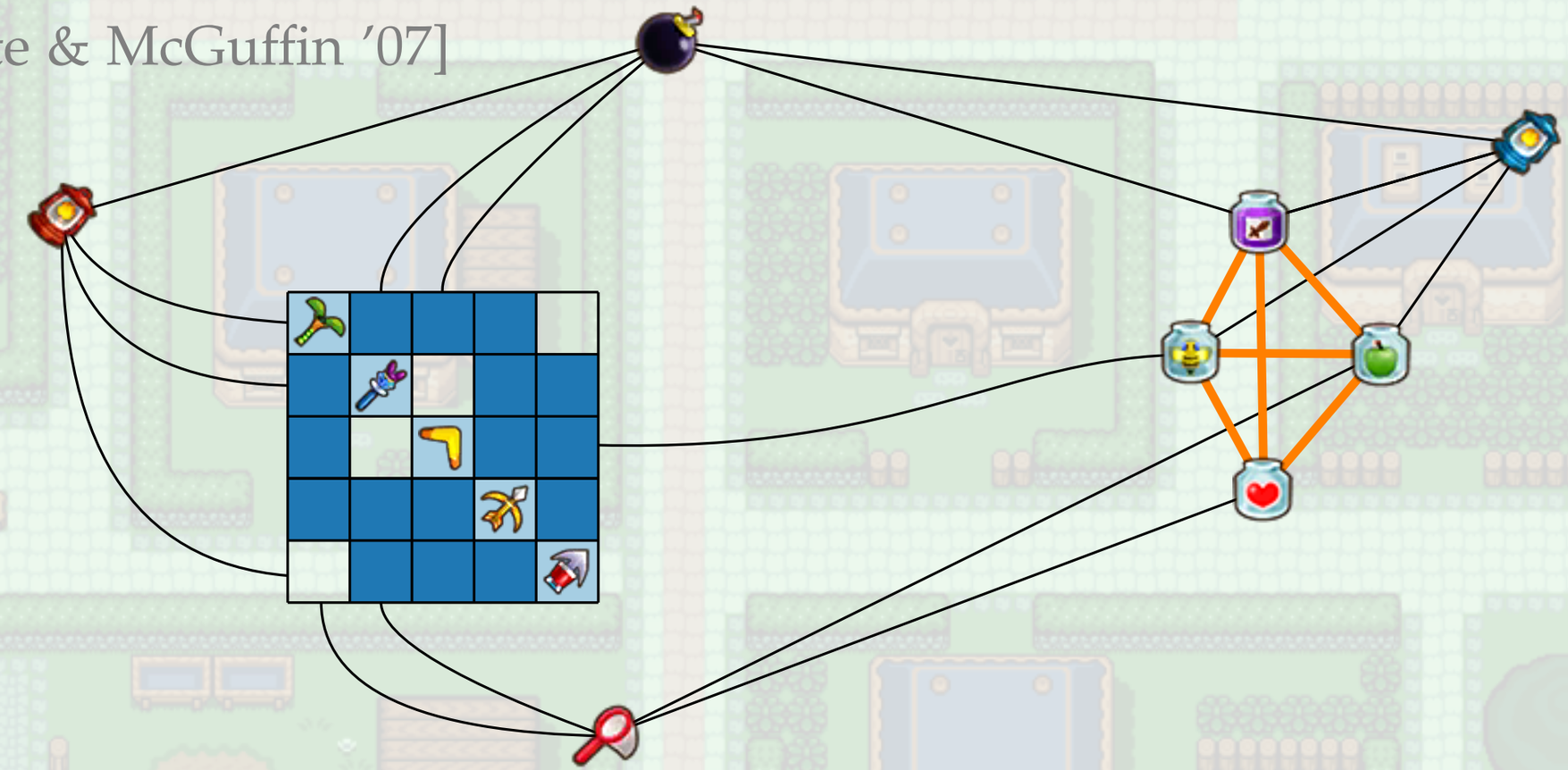
NodeTrix

[Henry, Fekete & McGuffin '07]



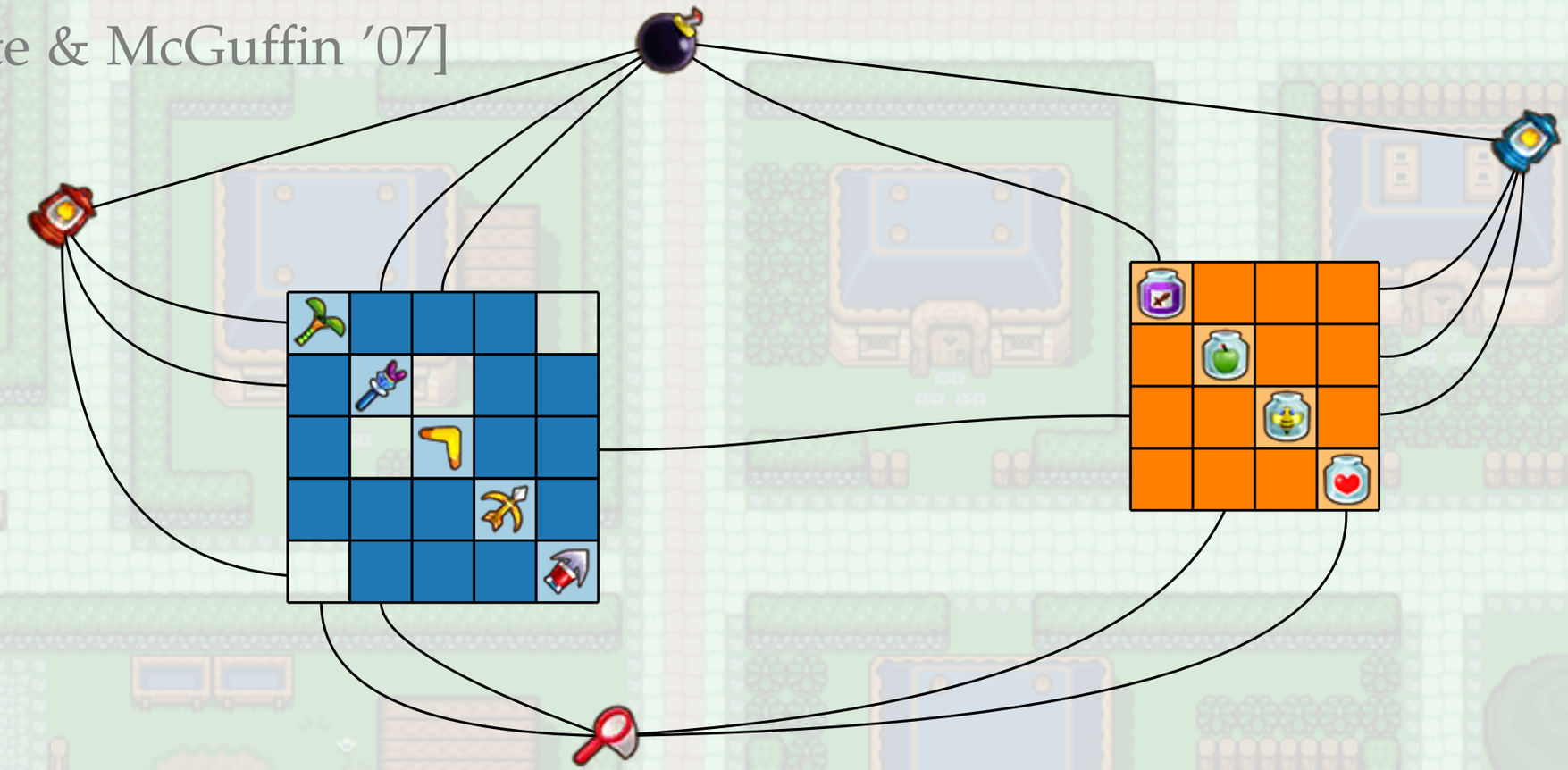
NodeTrix

[Henry, Fekete & McGuffin '07]



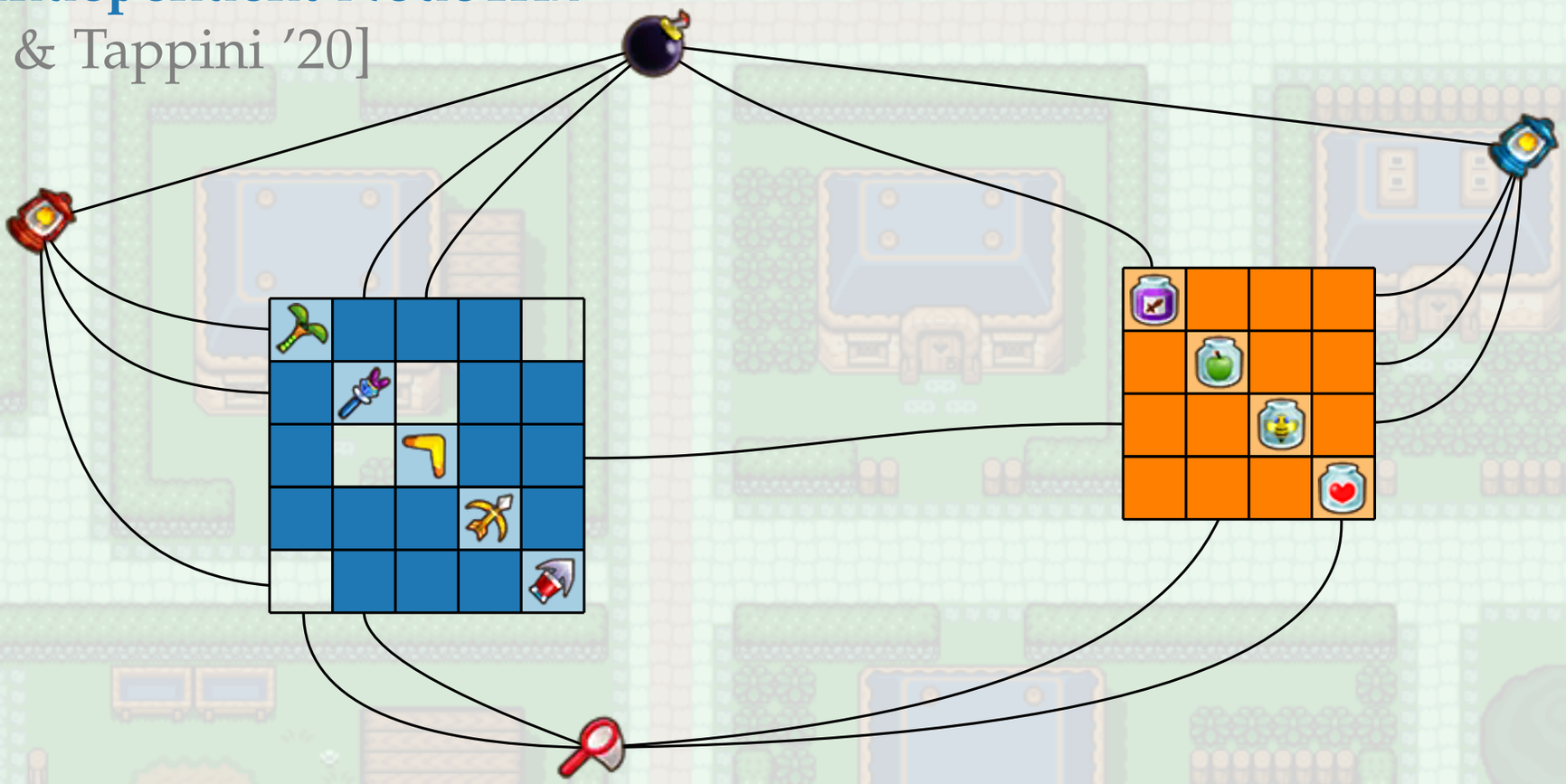
NodeTrix

[Henry, Fekete & McGuffin '07]



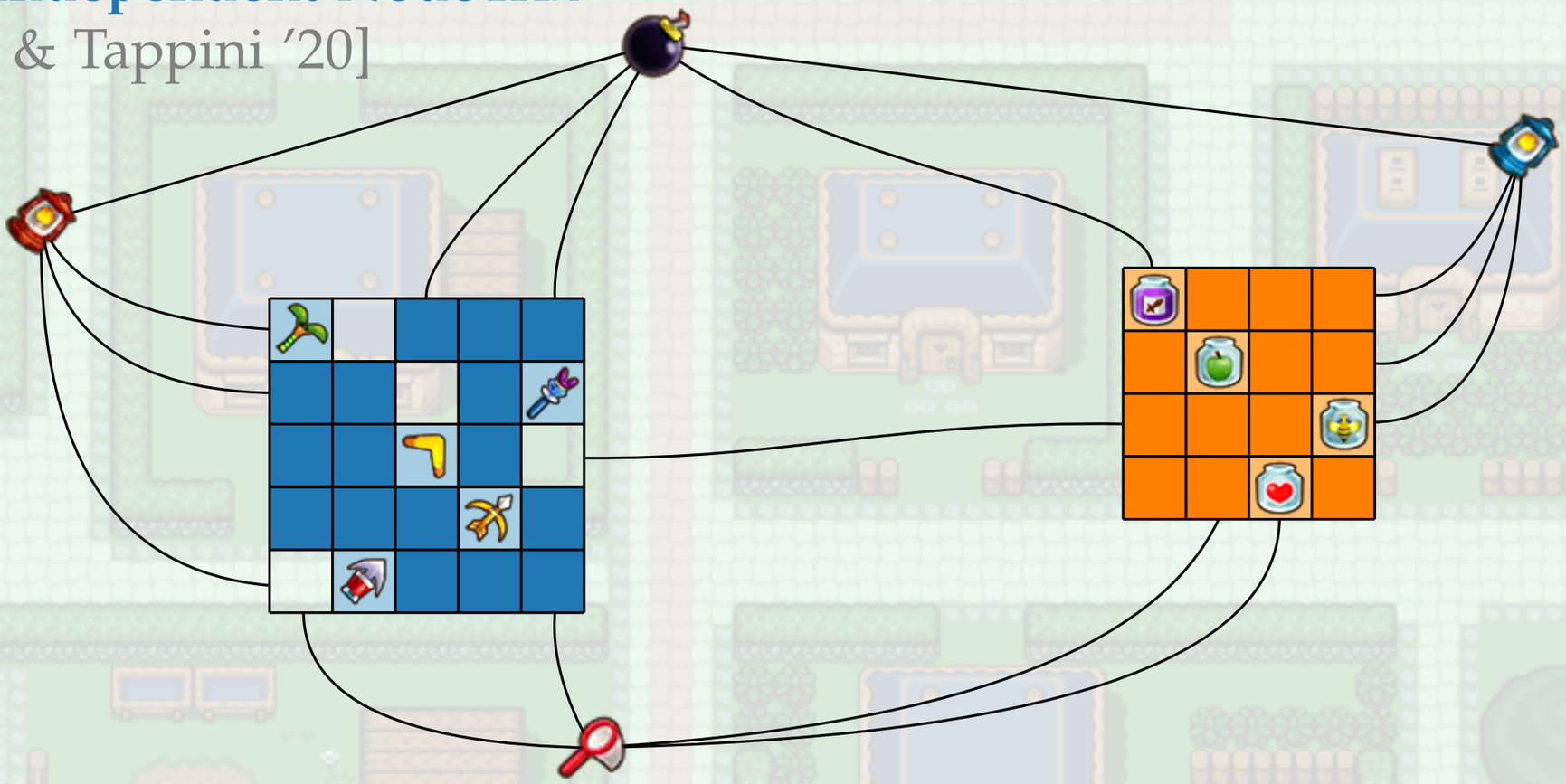
RowColumn Independent NodeTrix

[Liotta, Rutter & Tappini '20]



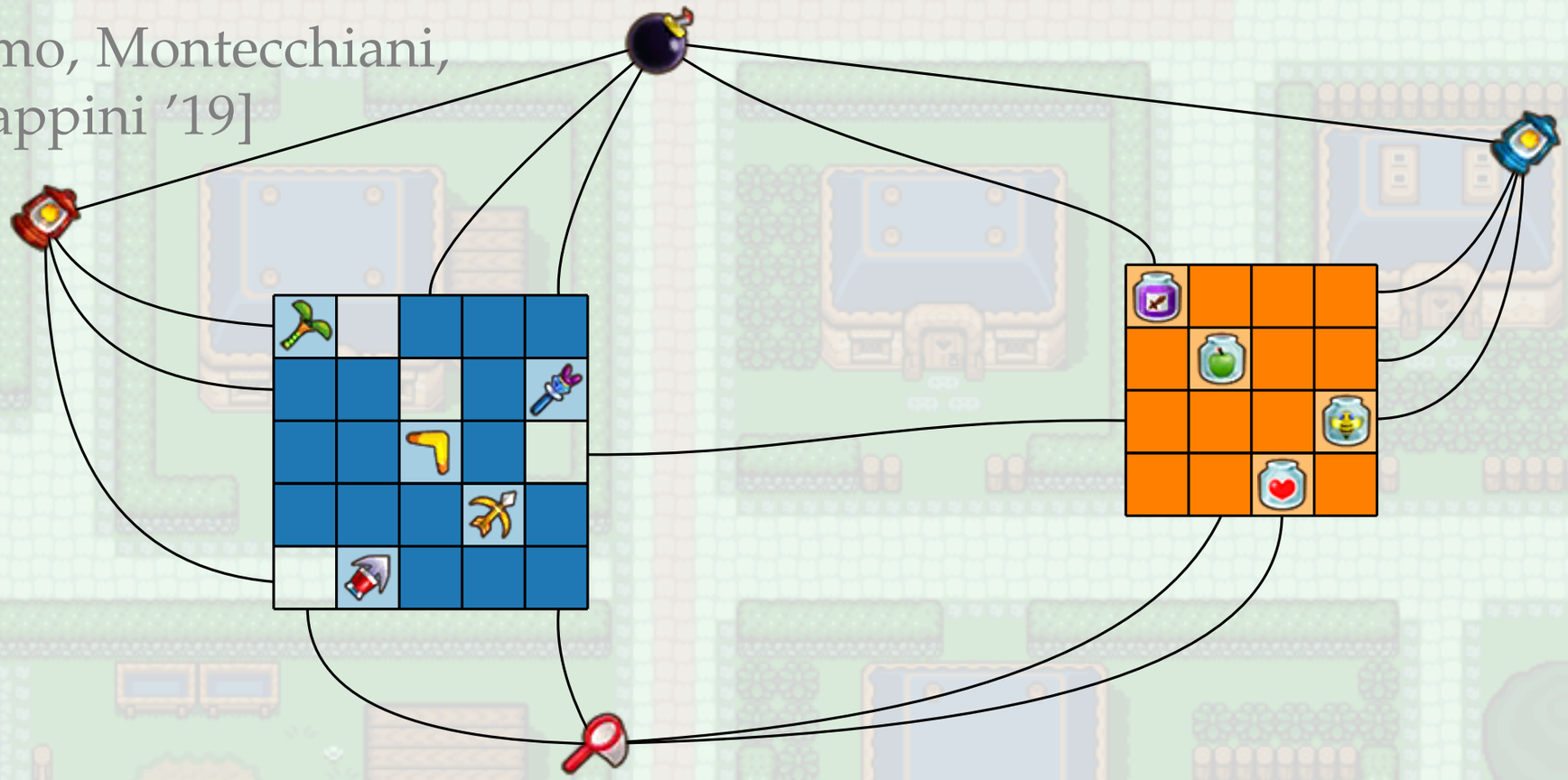
RowColumn Independent NodeTrix

[Liotta, Rutter & Tappini '20]



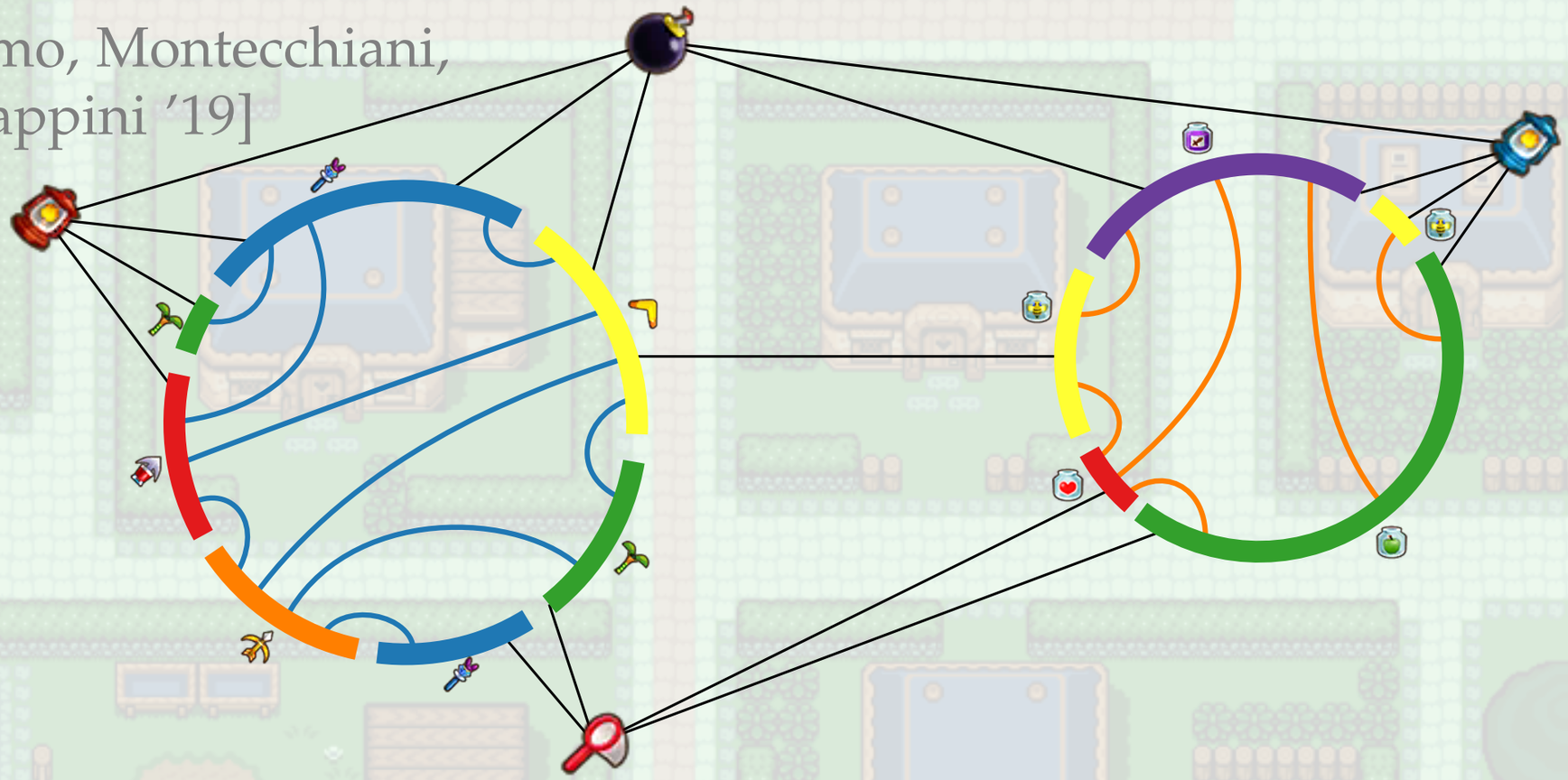
ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



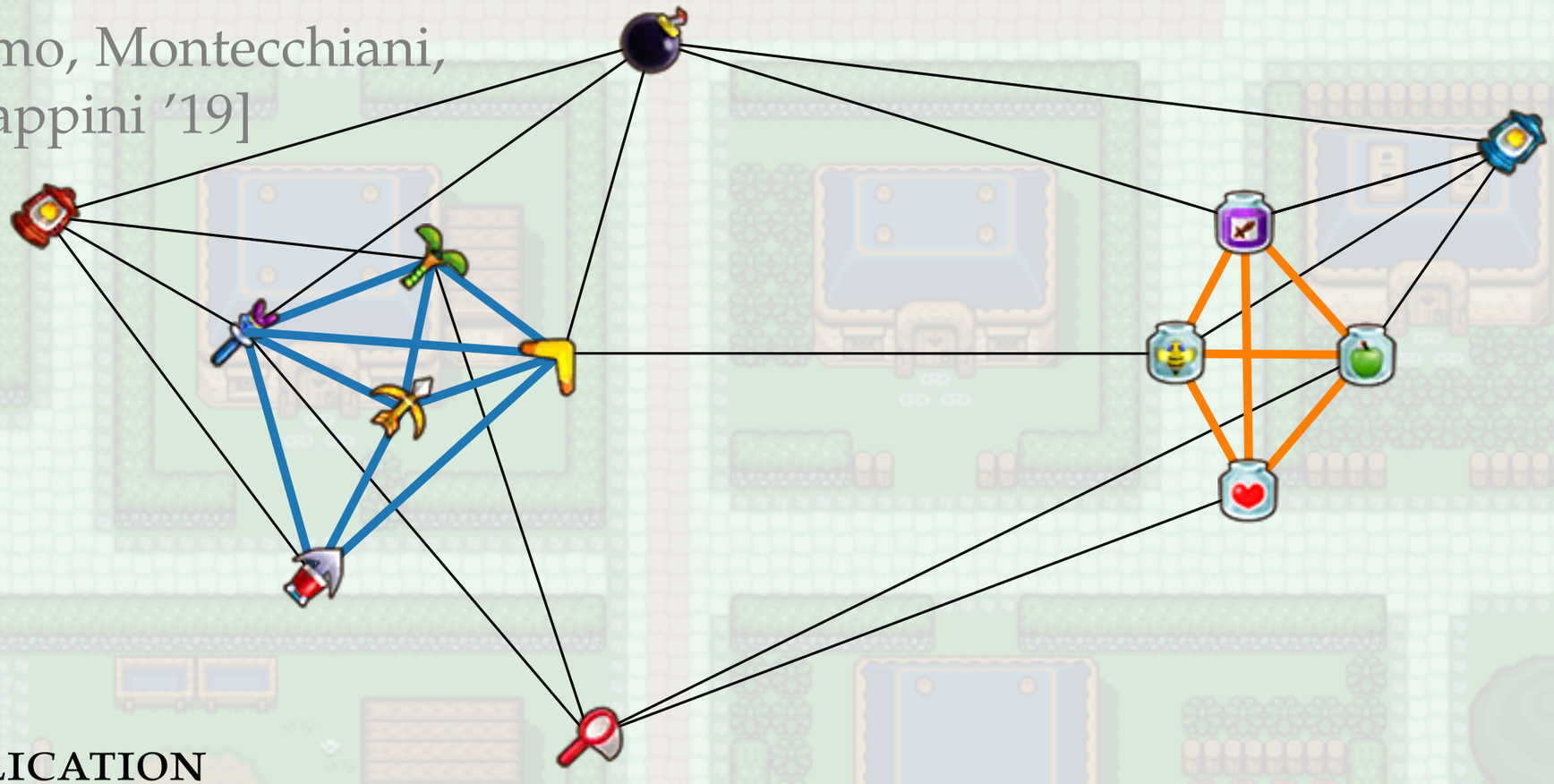
ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



ChordLink

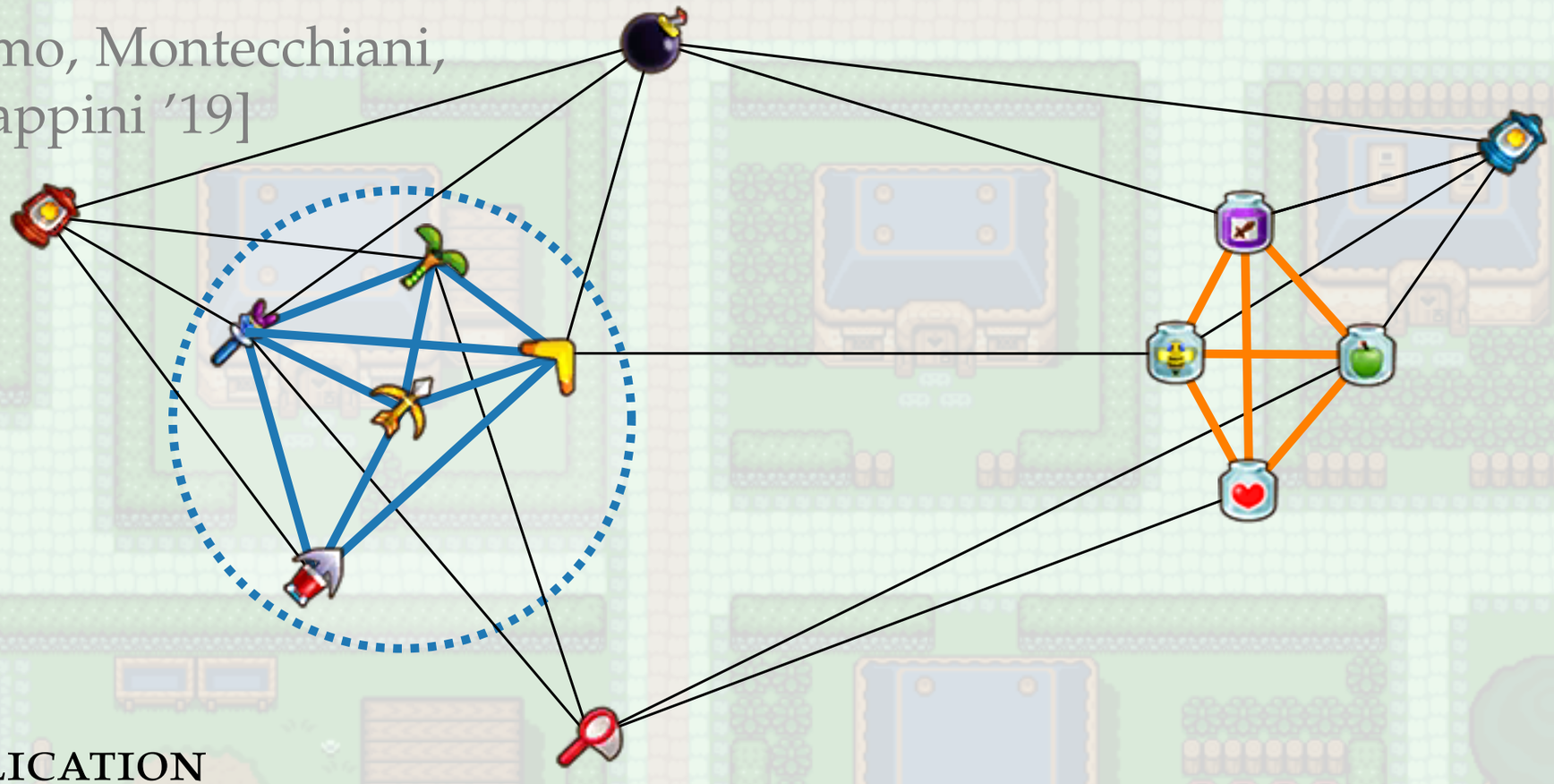
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

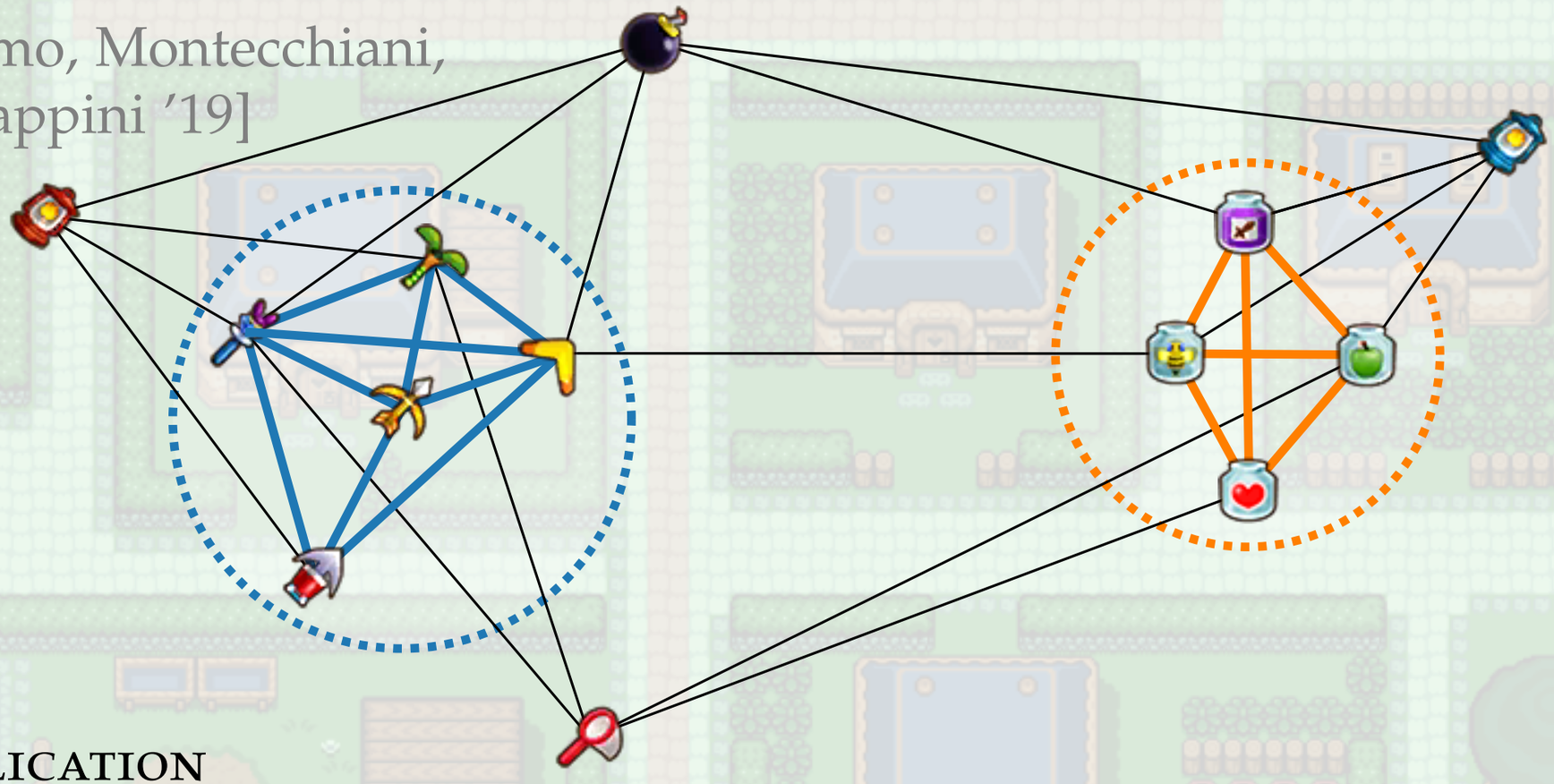
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

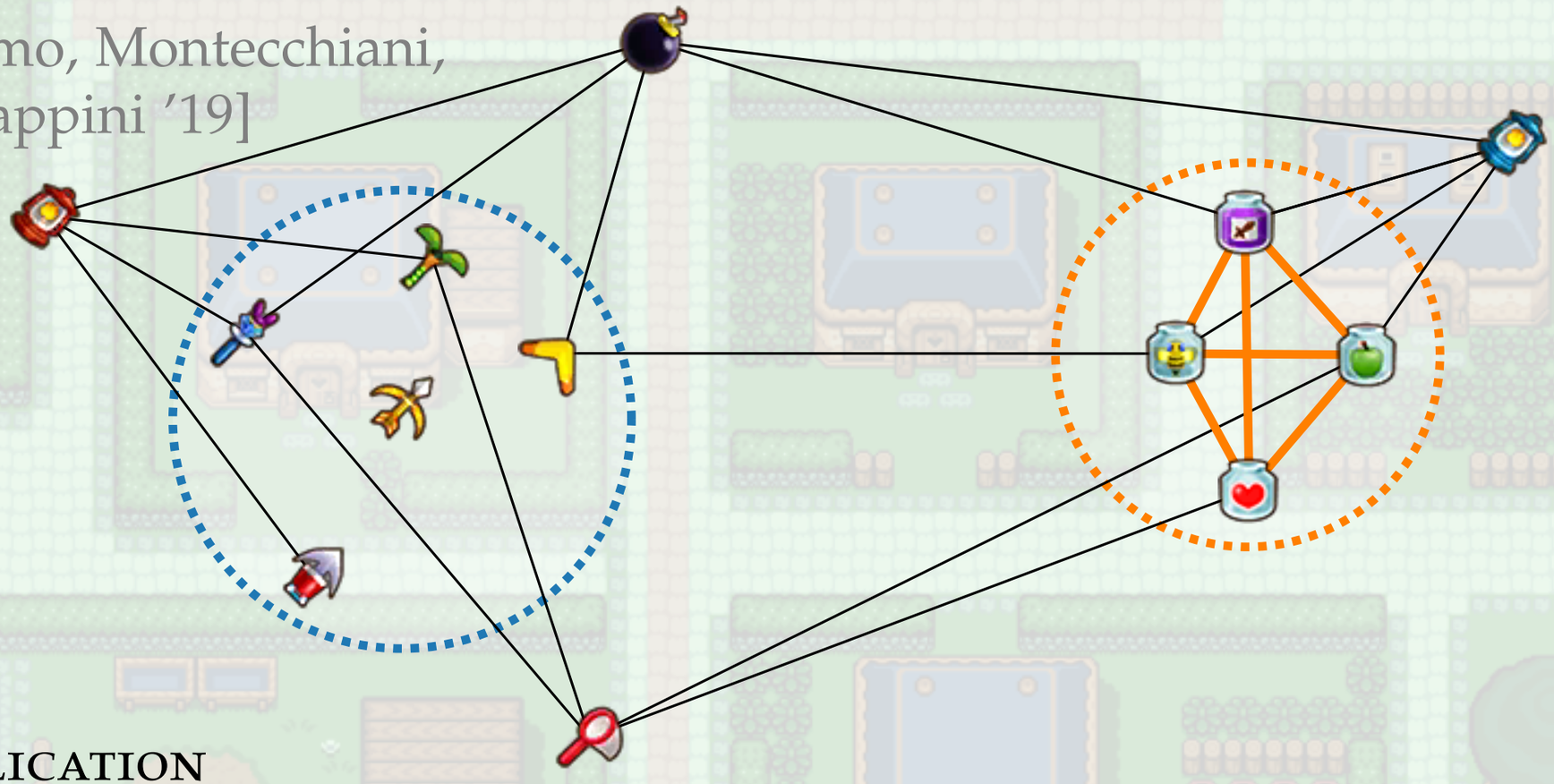
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

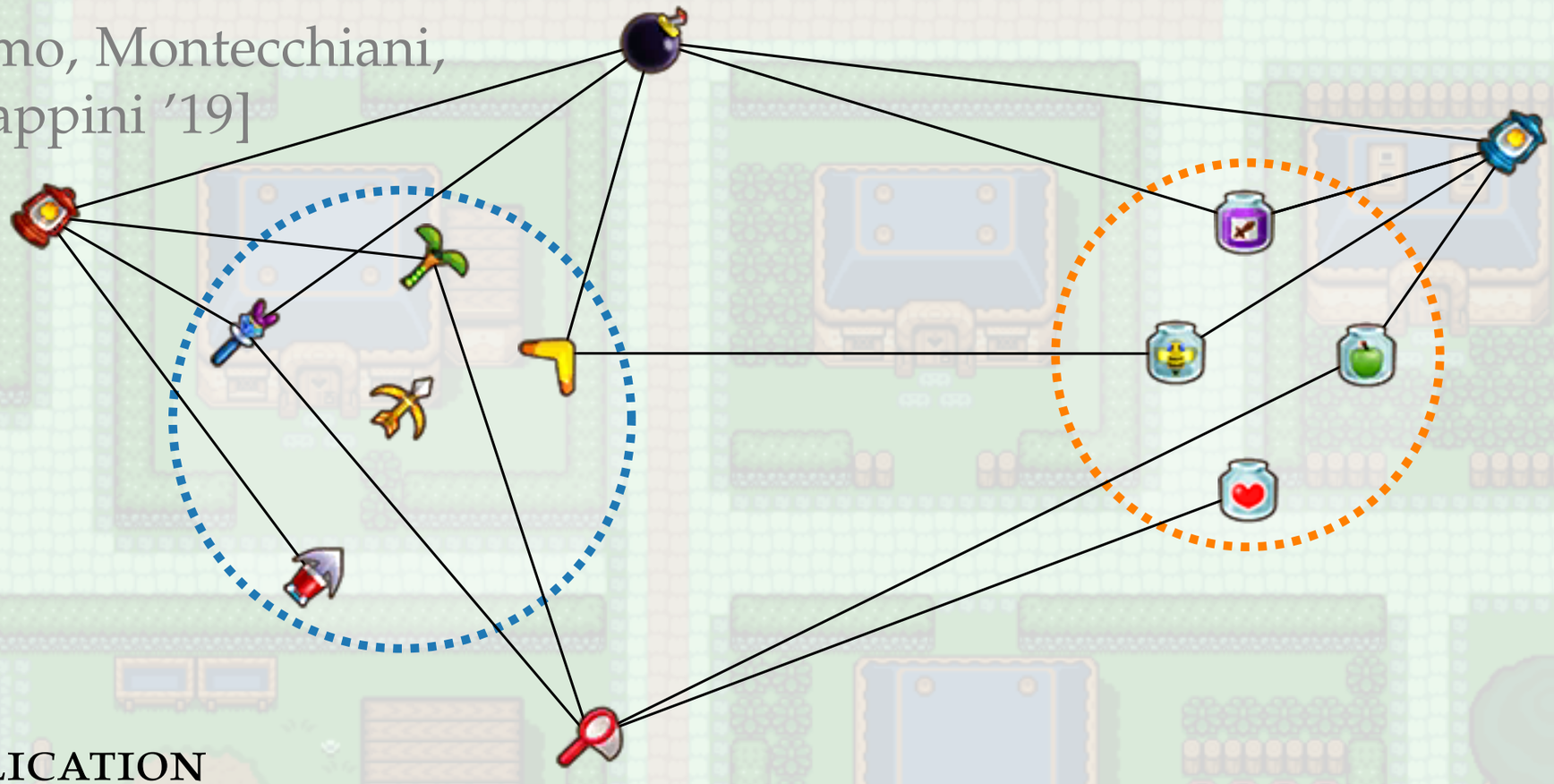
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

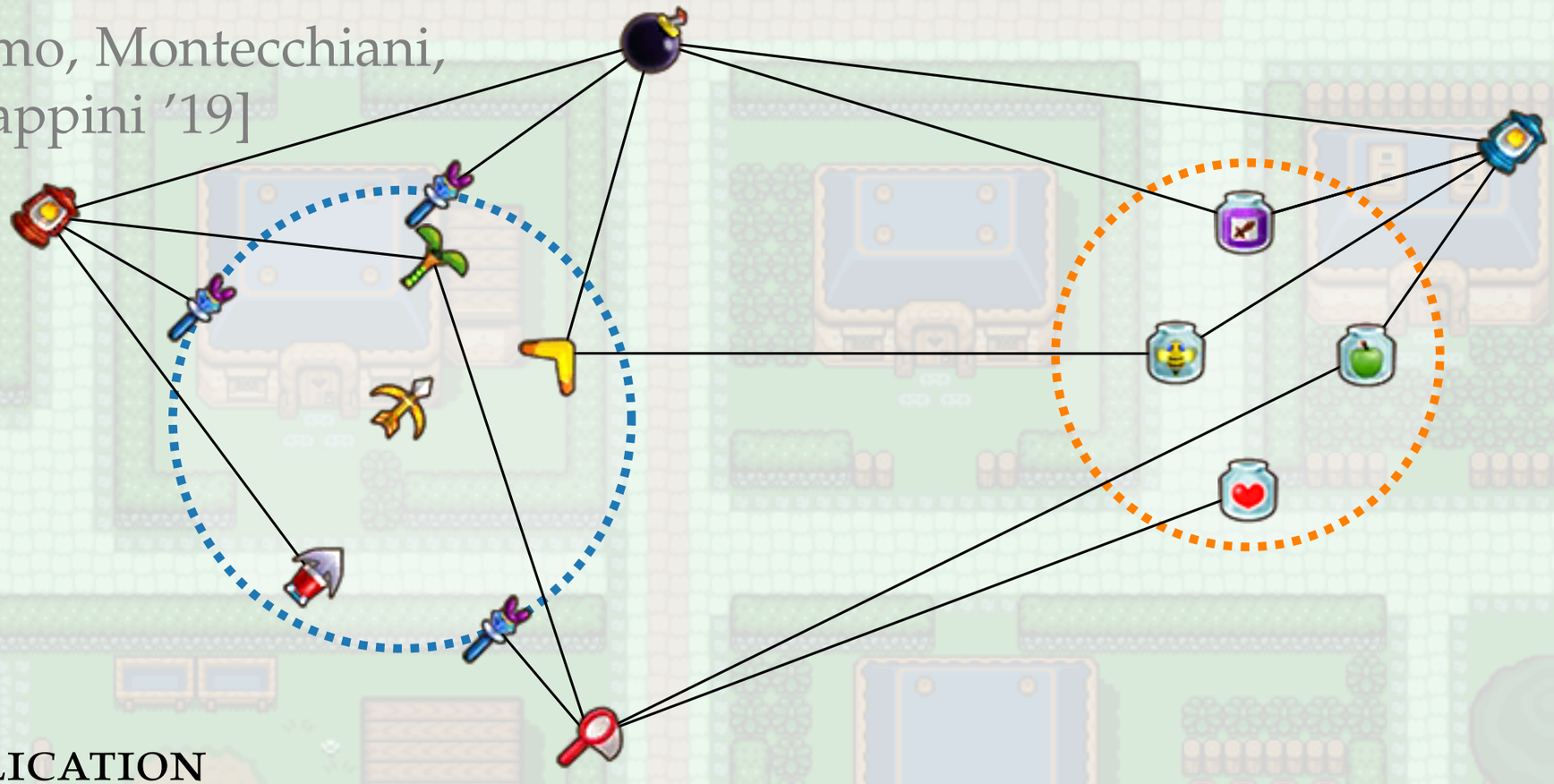
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

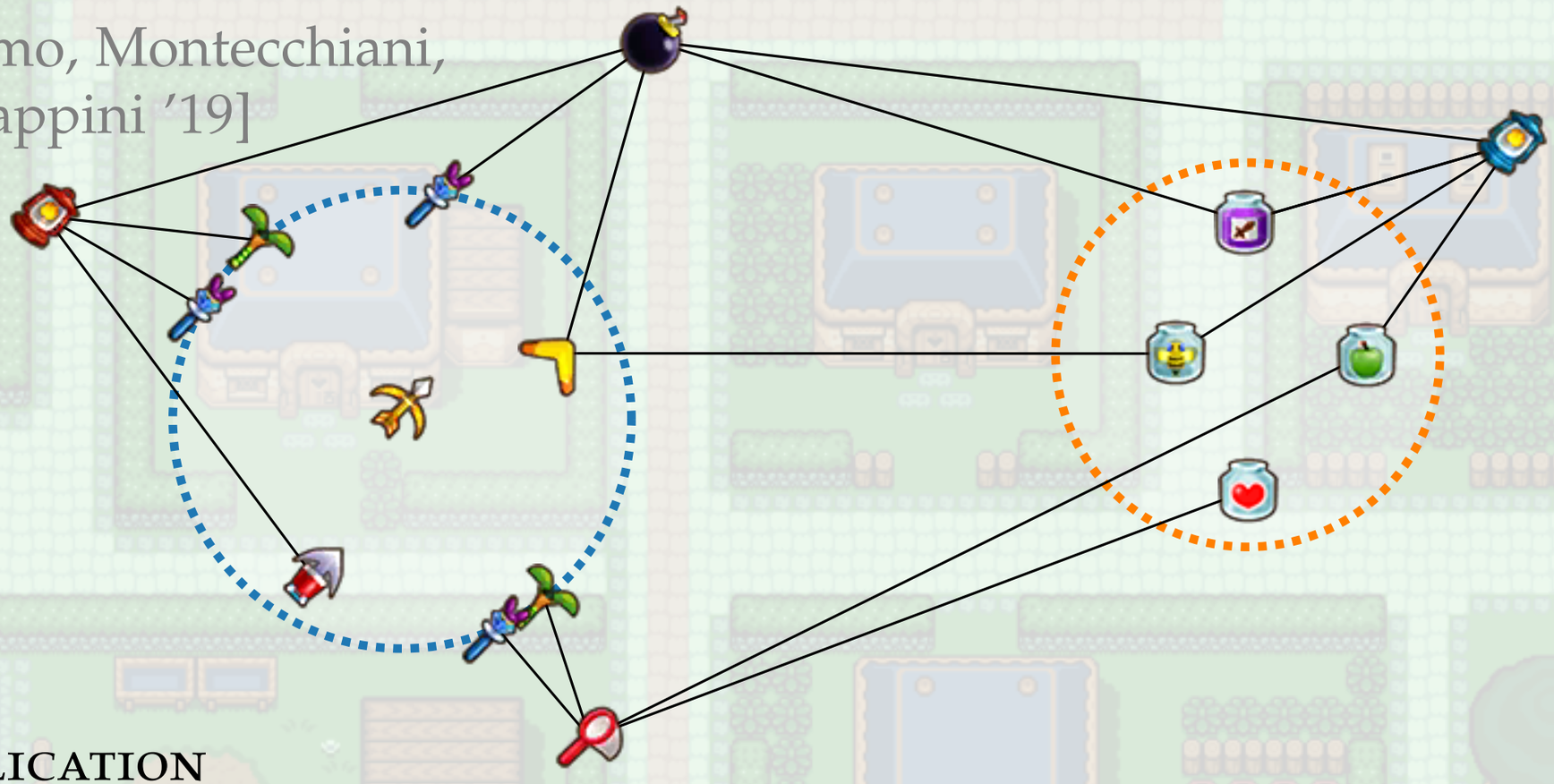
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

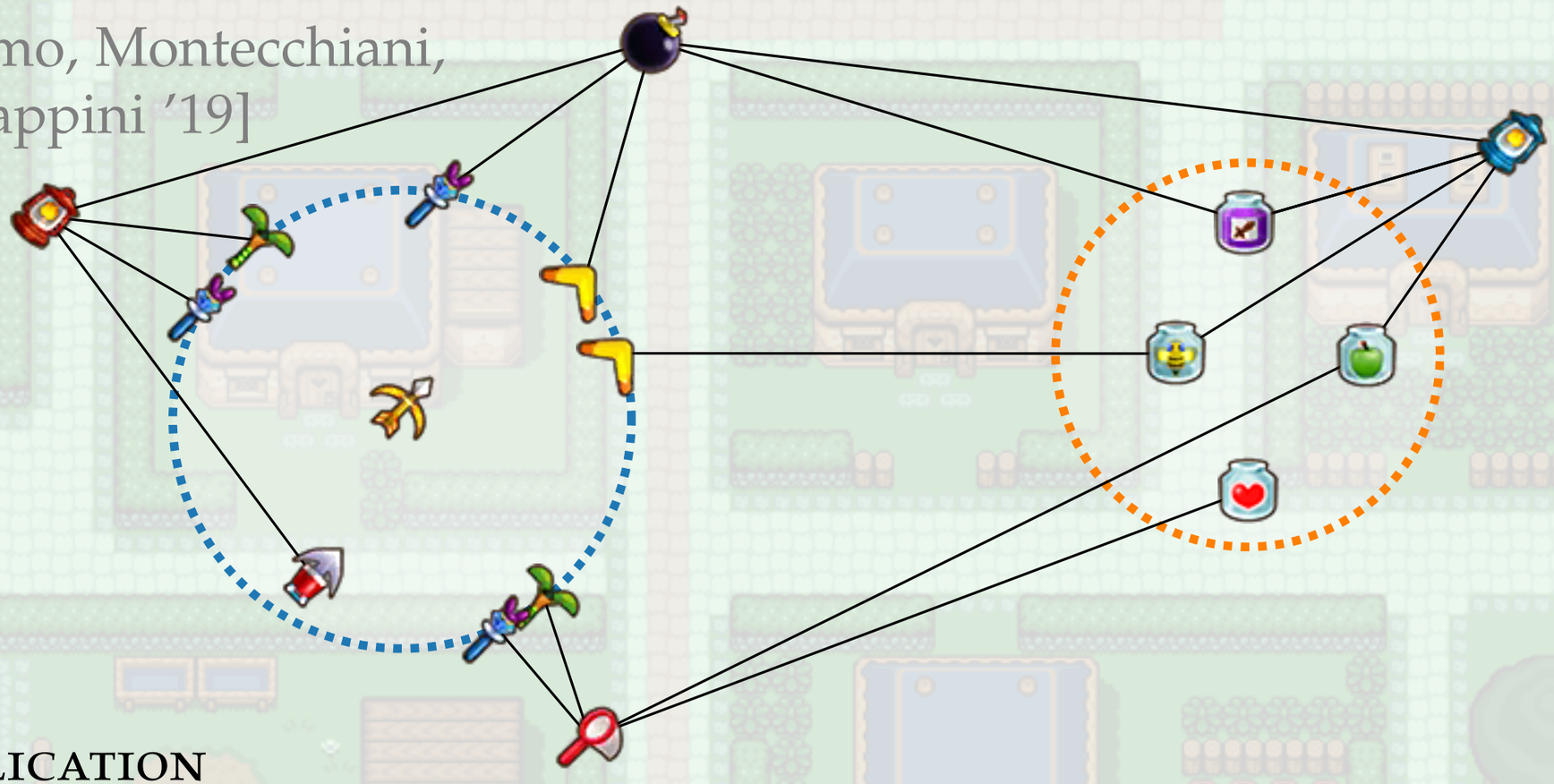
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

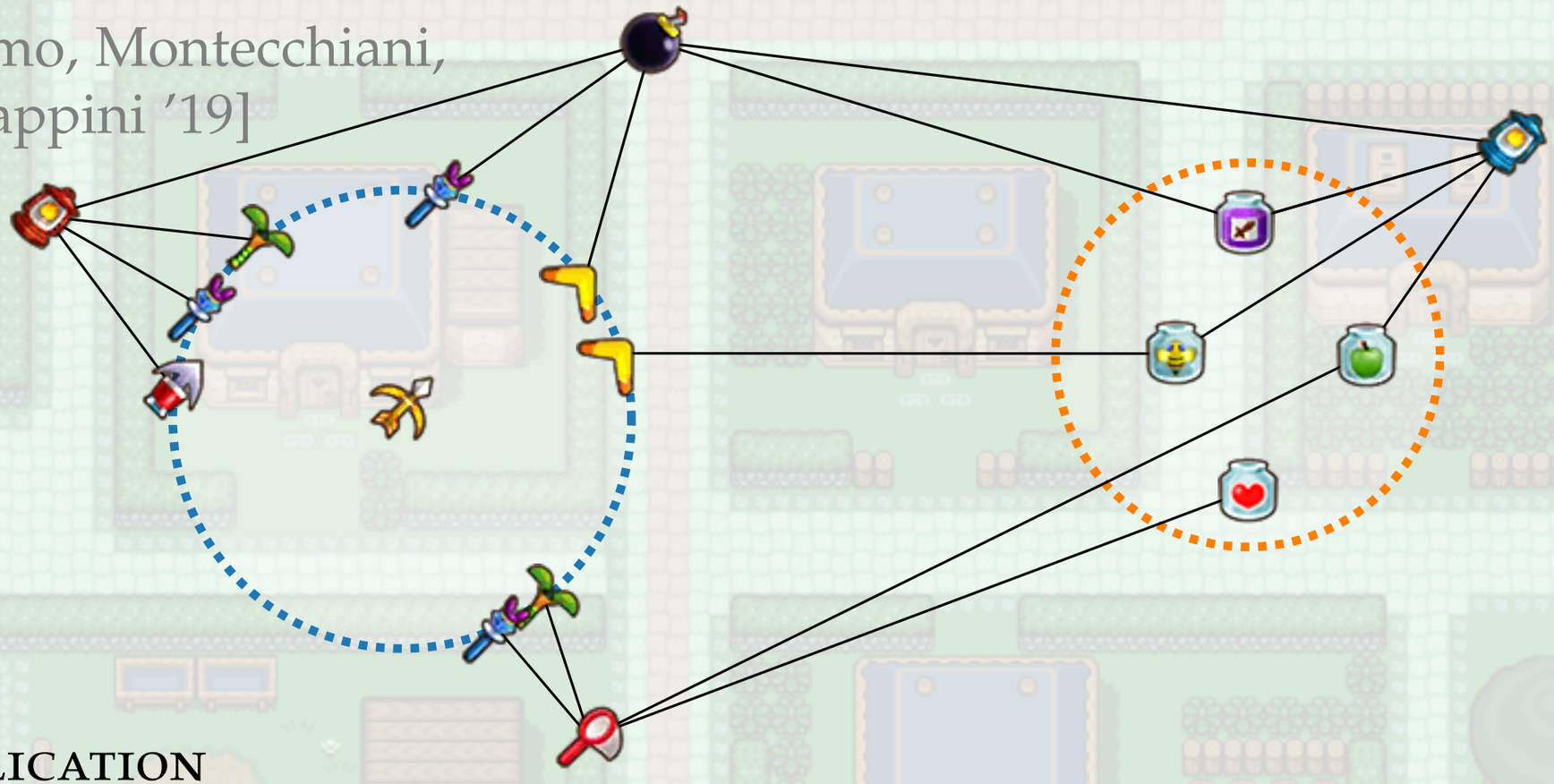
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

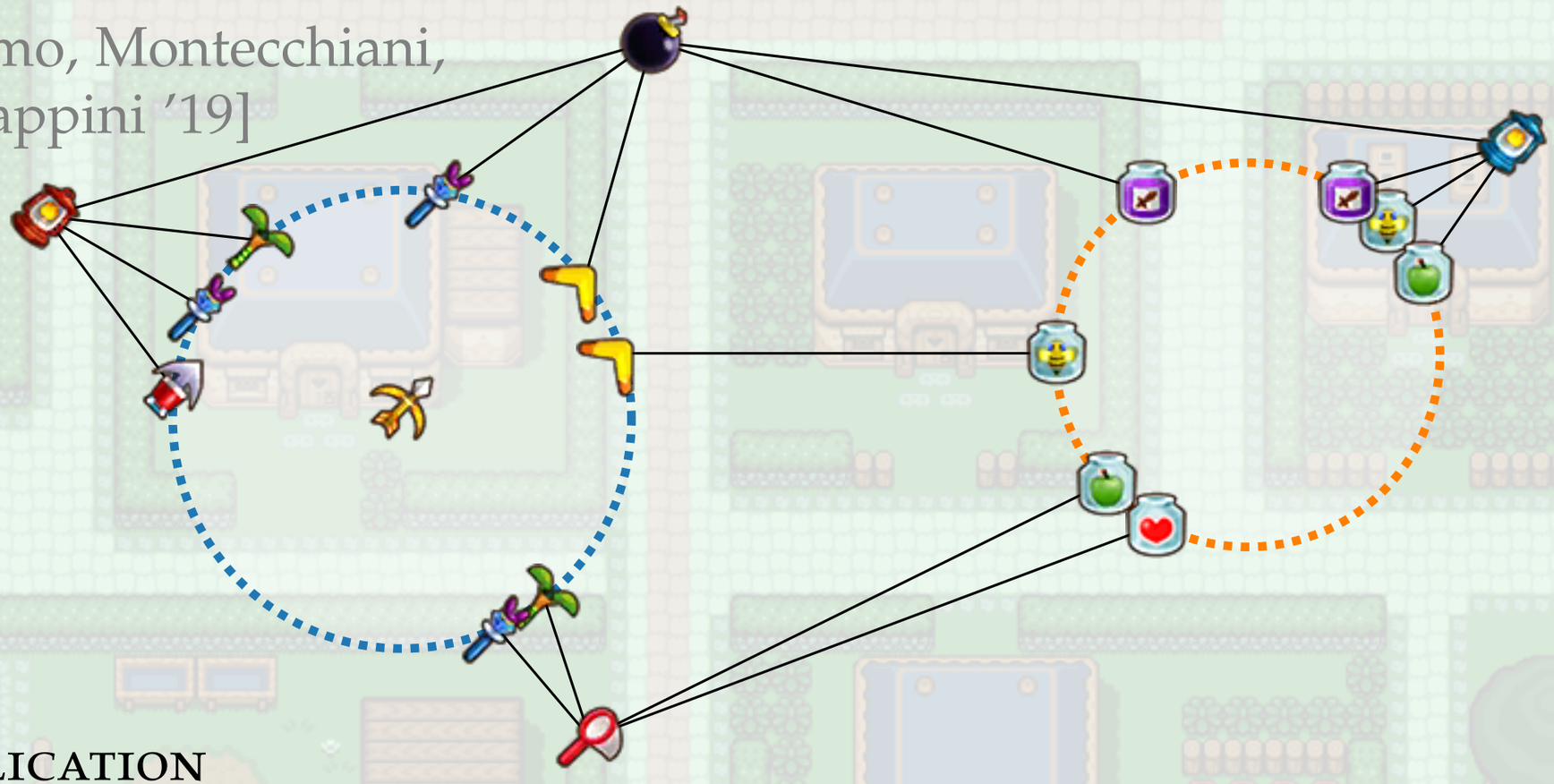
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

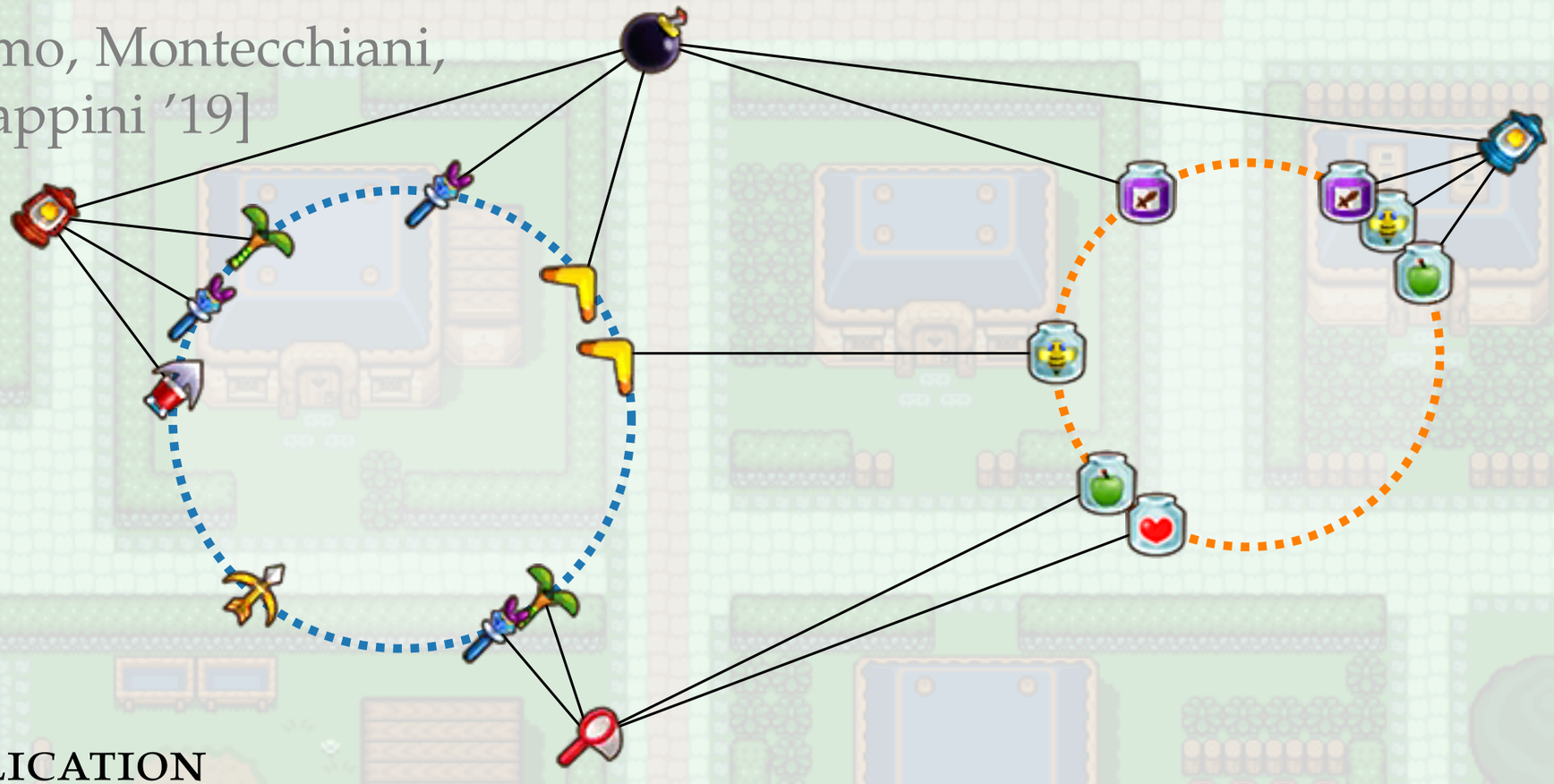
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

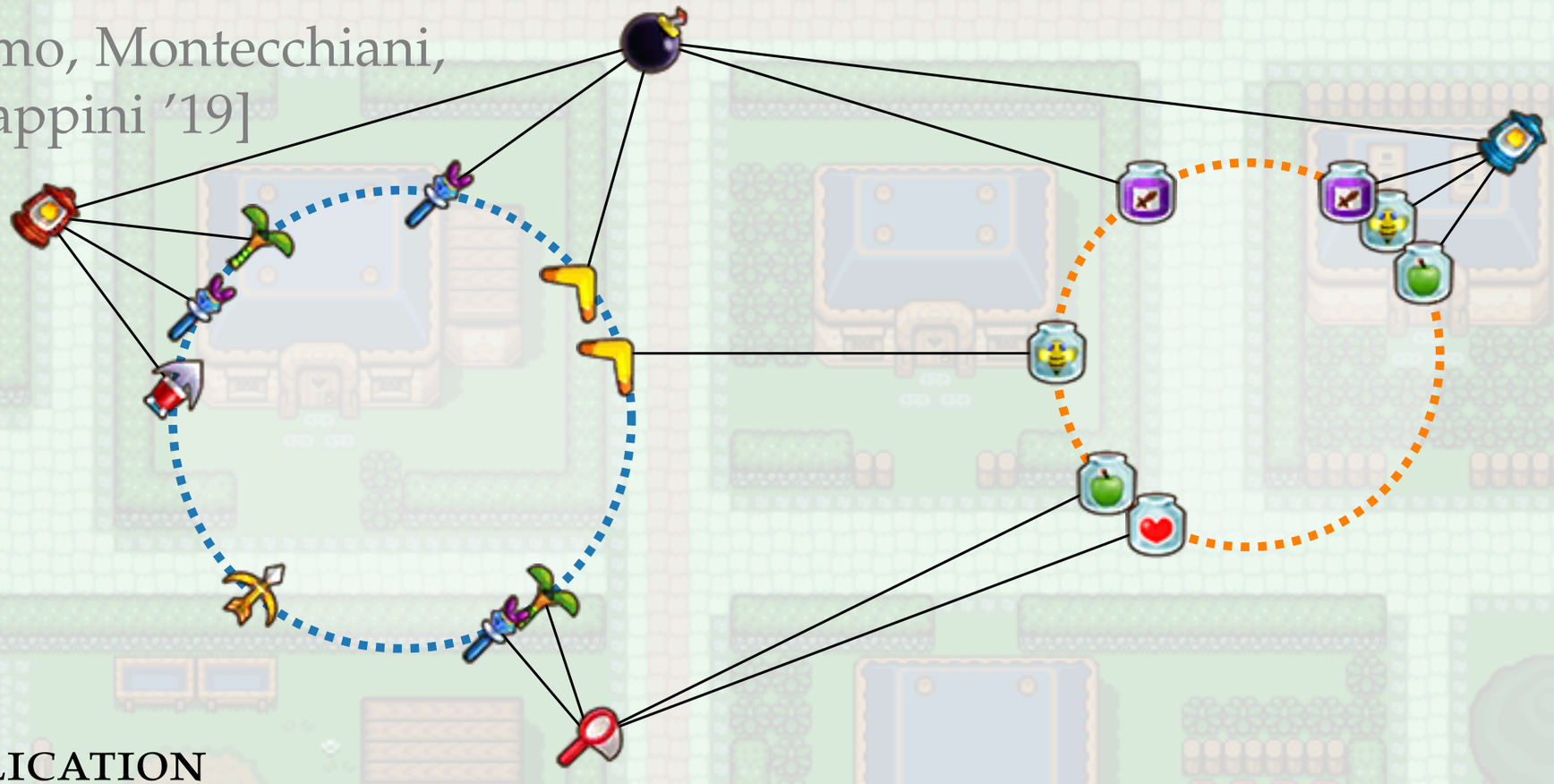
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication

ChordLink

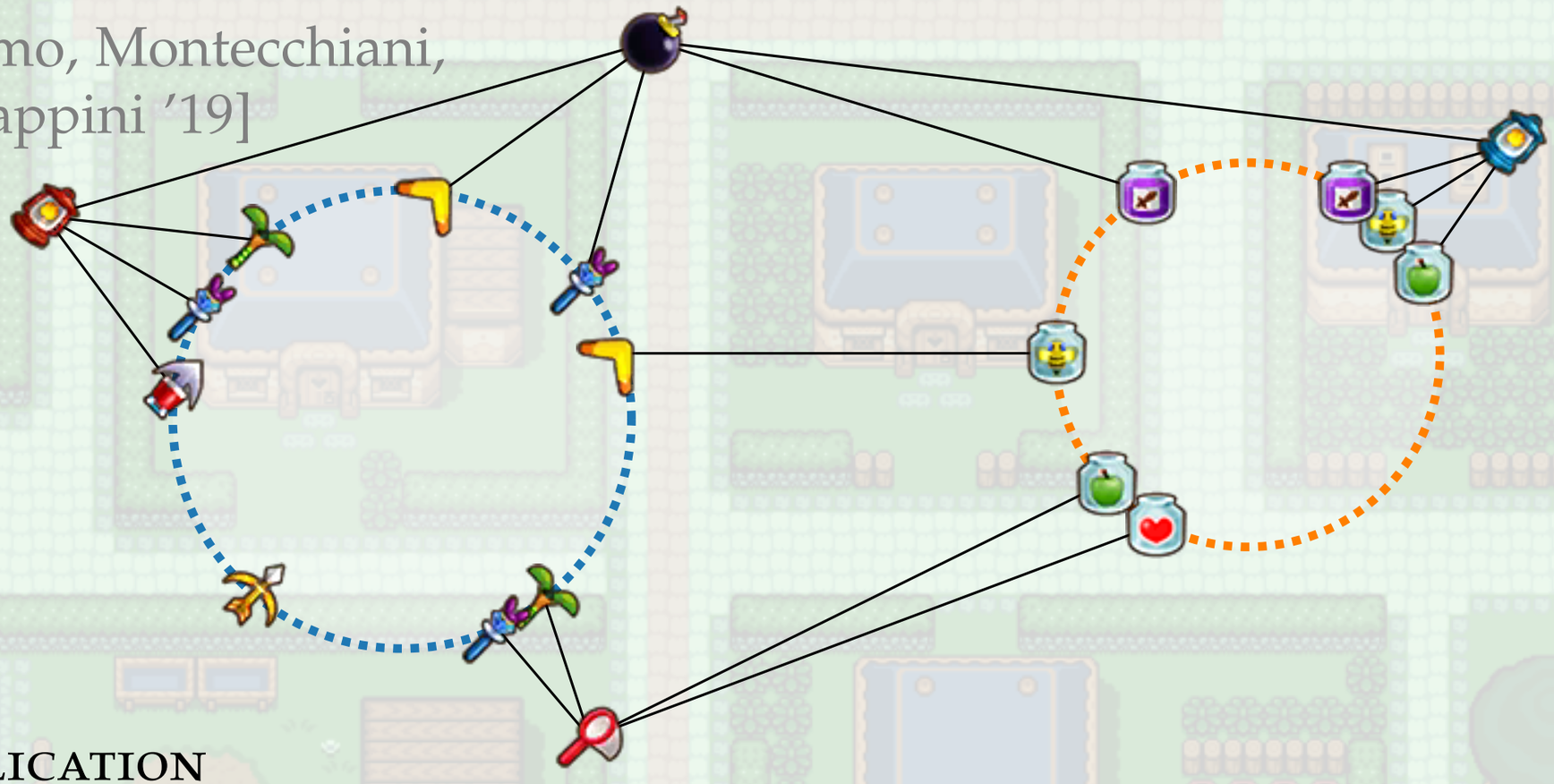
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation

ChordLink

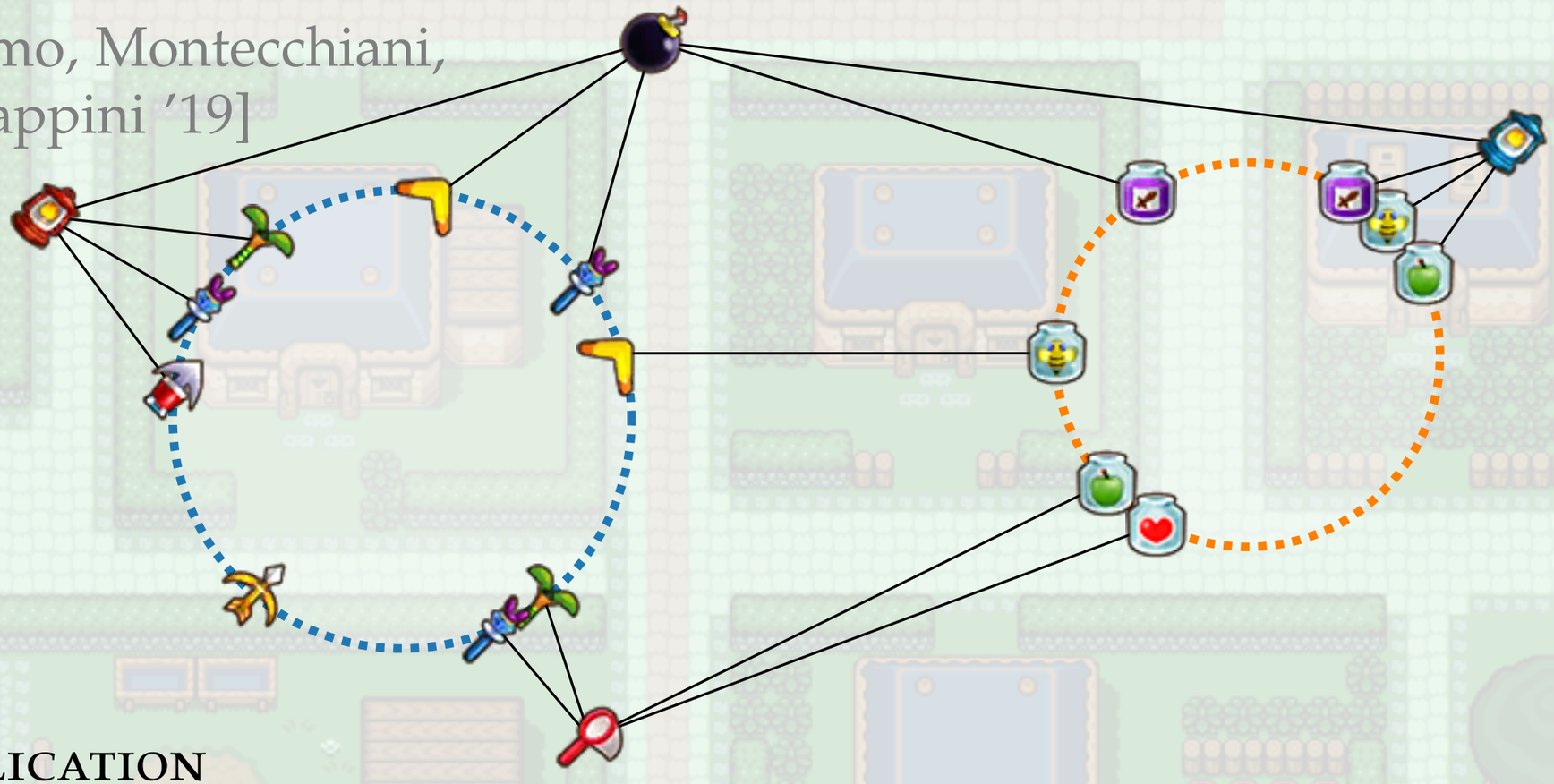
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation

ChordLink

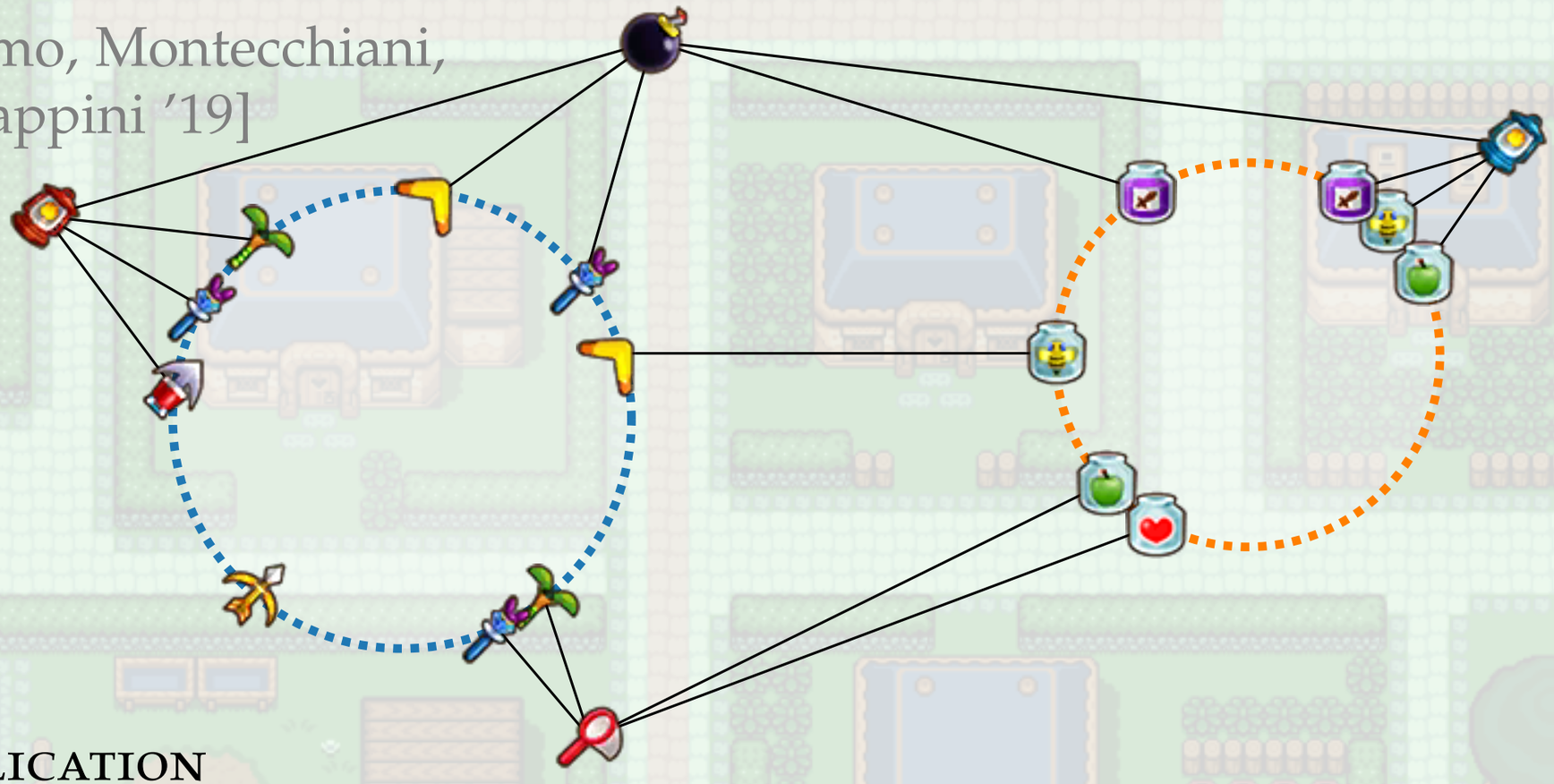
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation Can flip nodes with common neighbor

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



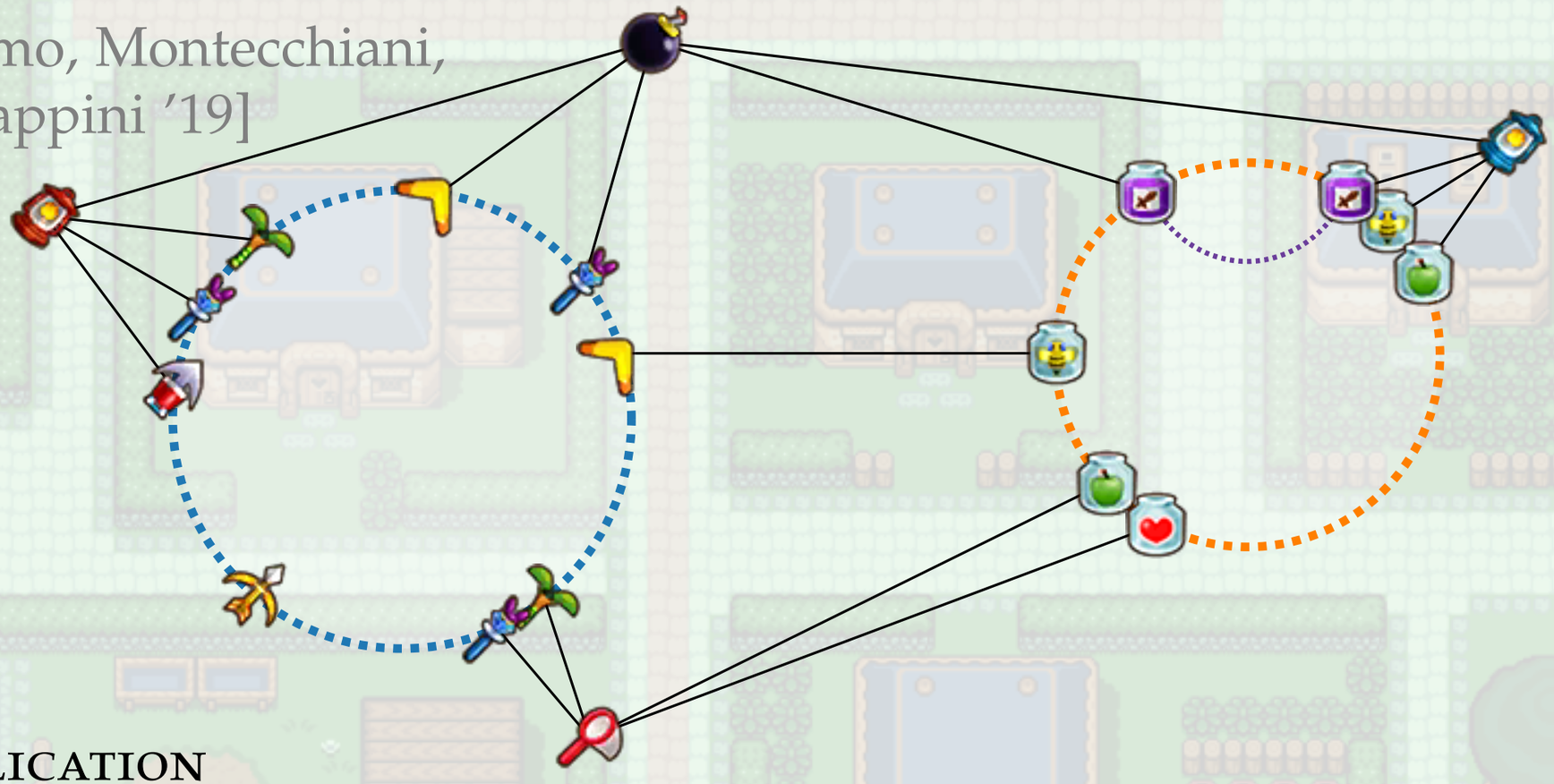
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



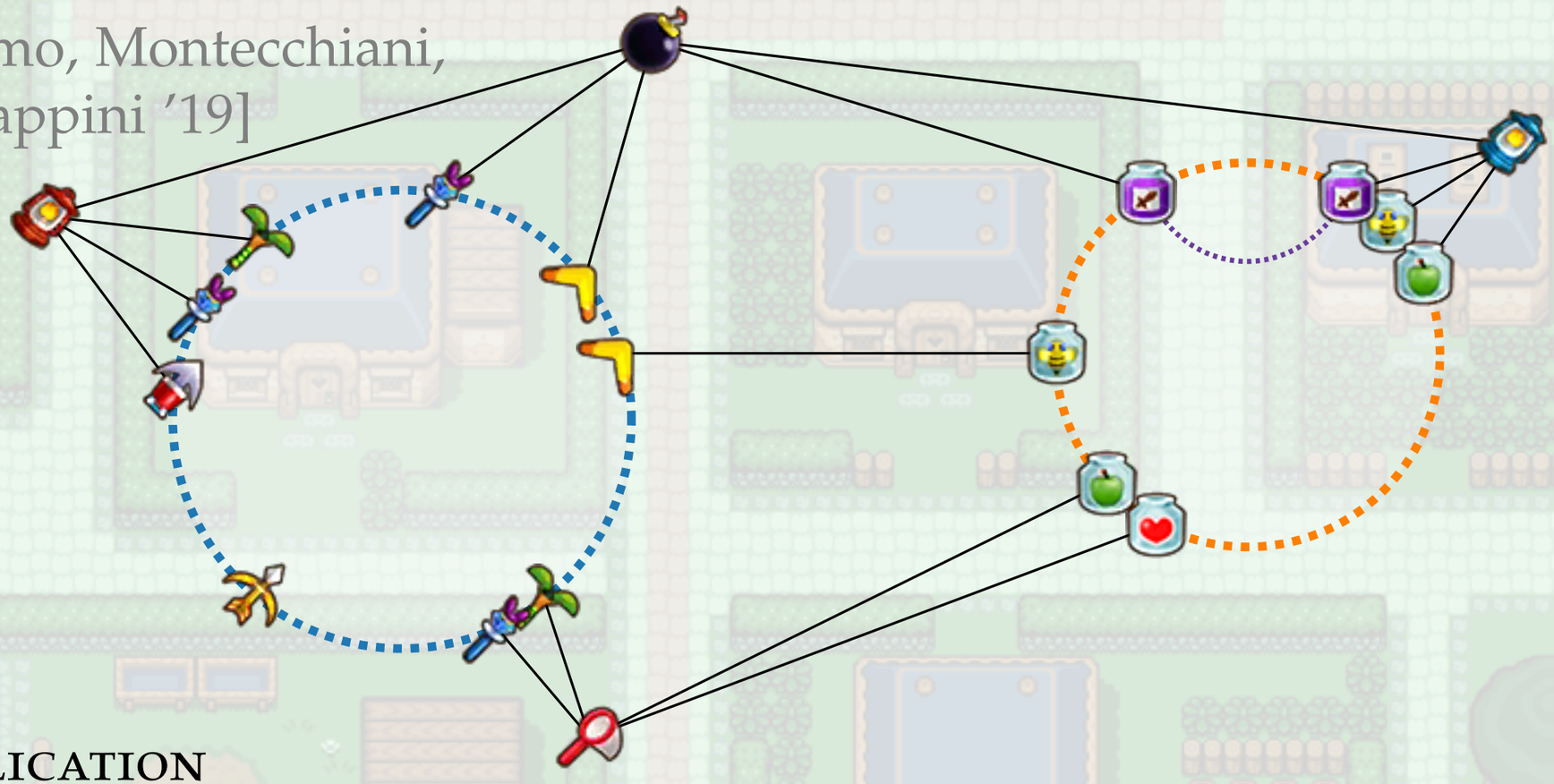
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



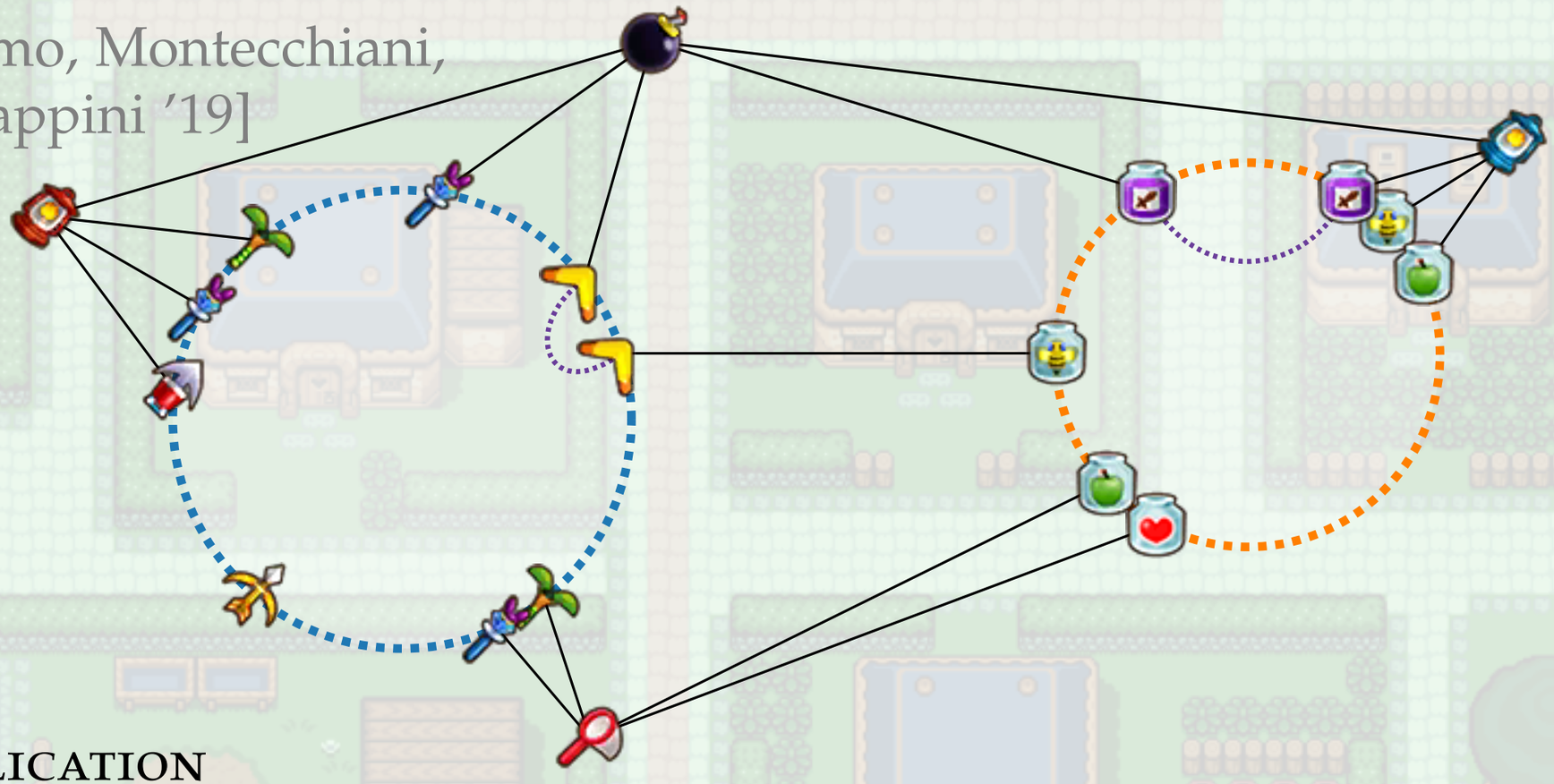
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



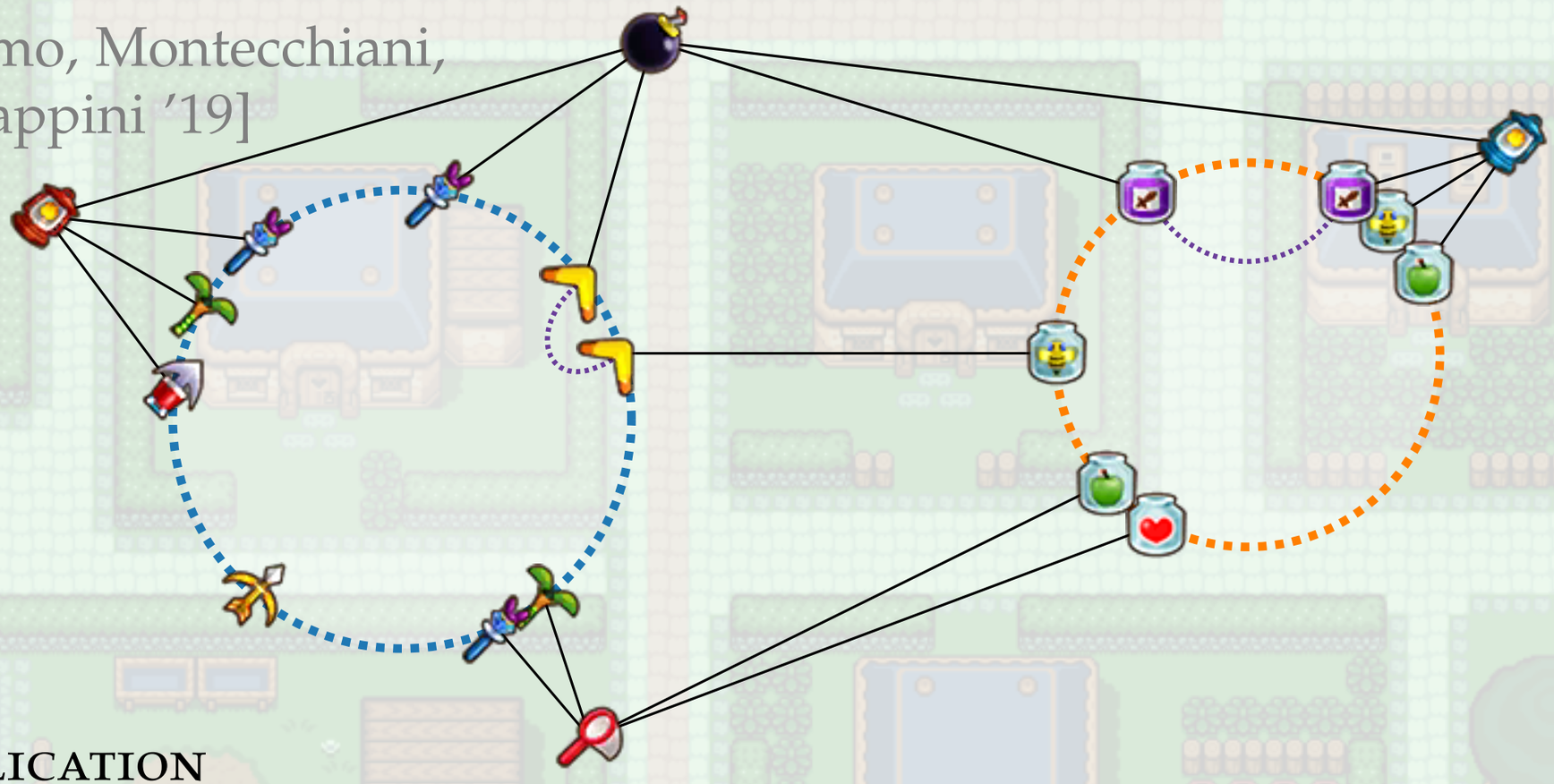
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



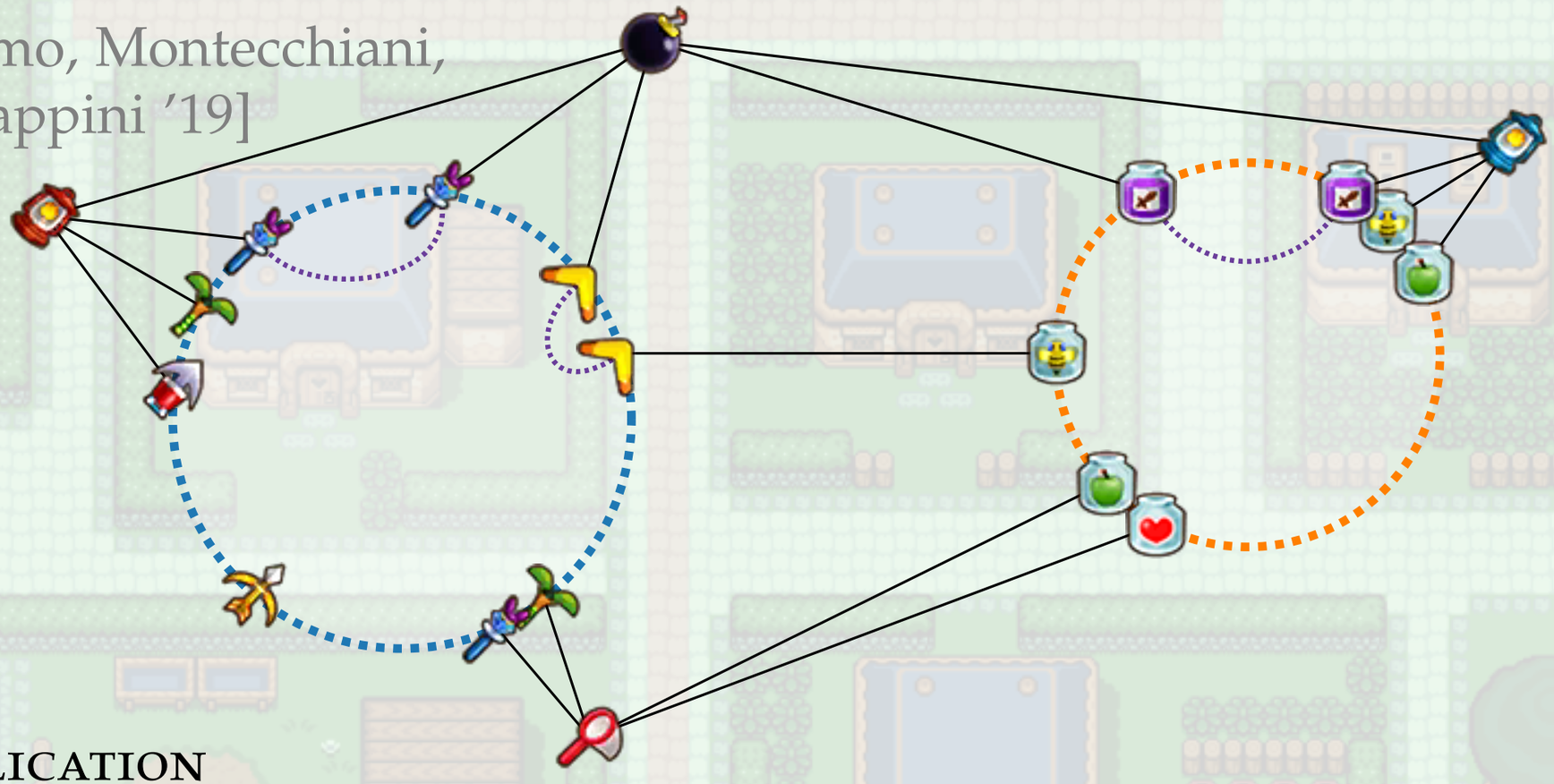
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



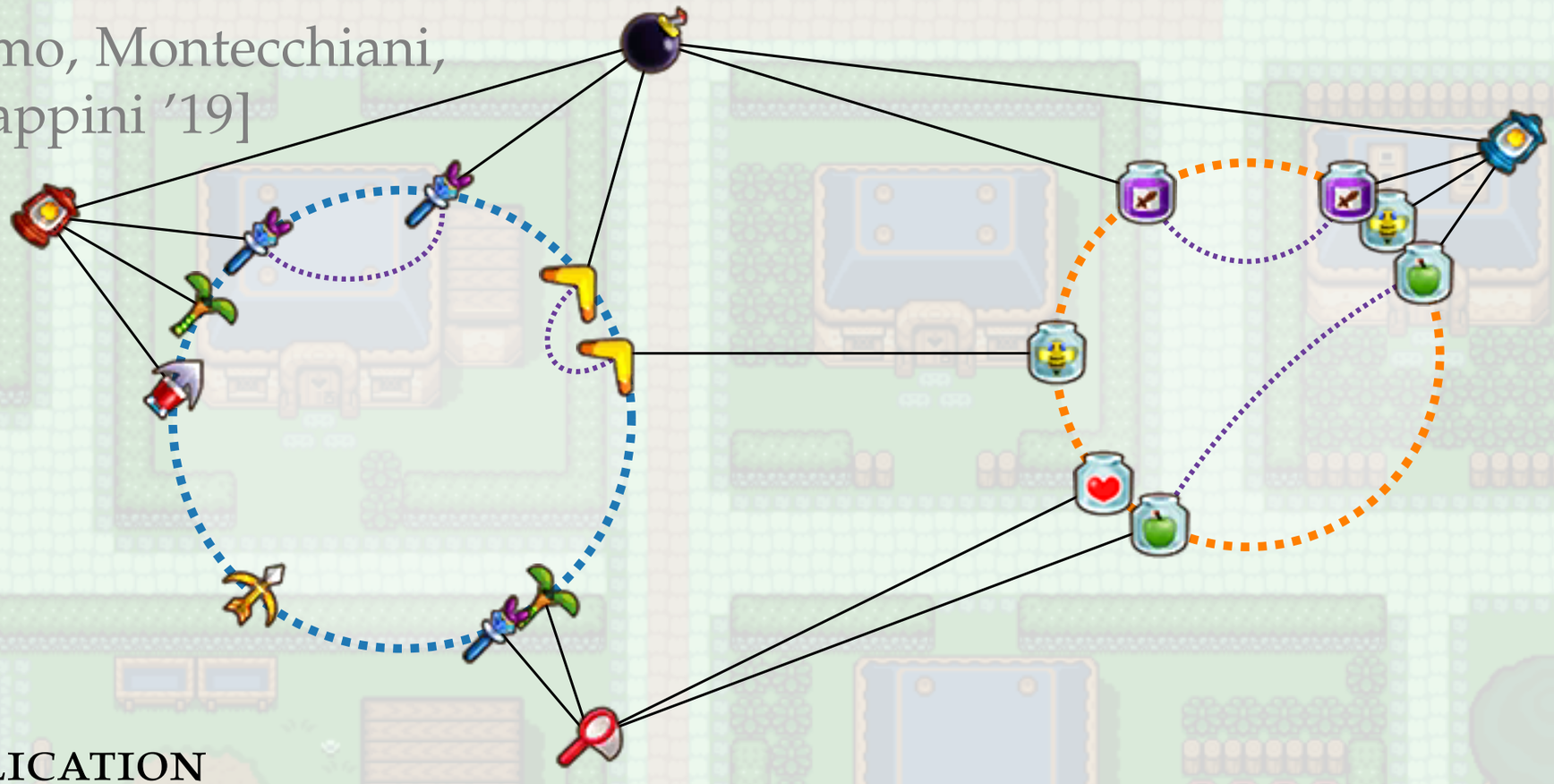
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



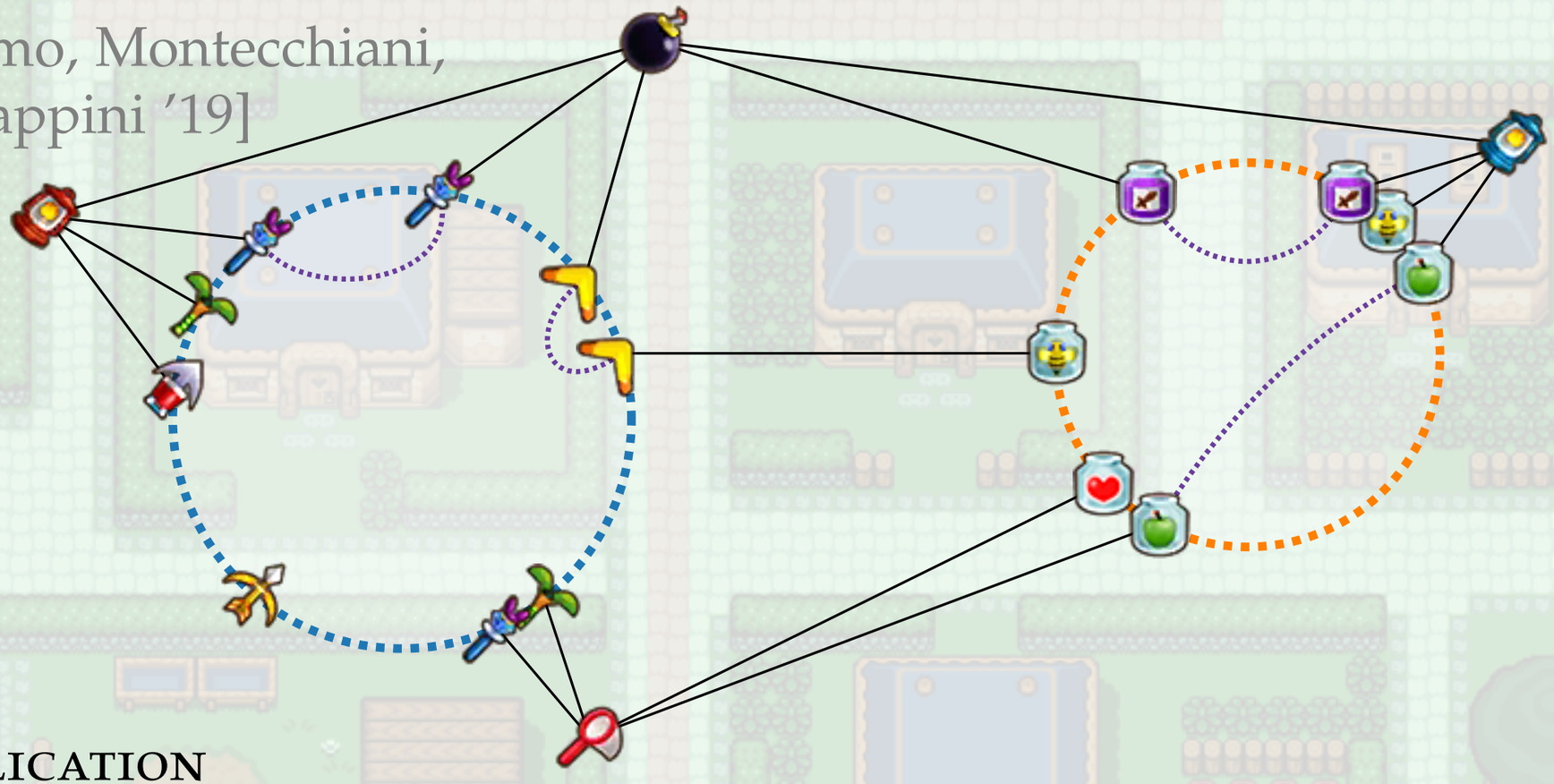
1. NodeReplication
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon

ChordLink

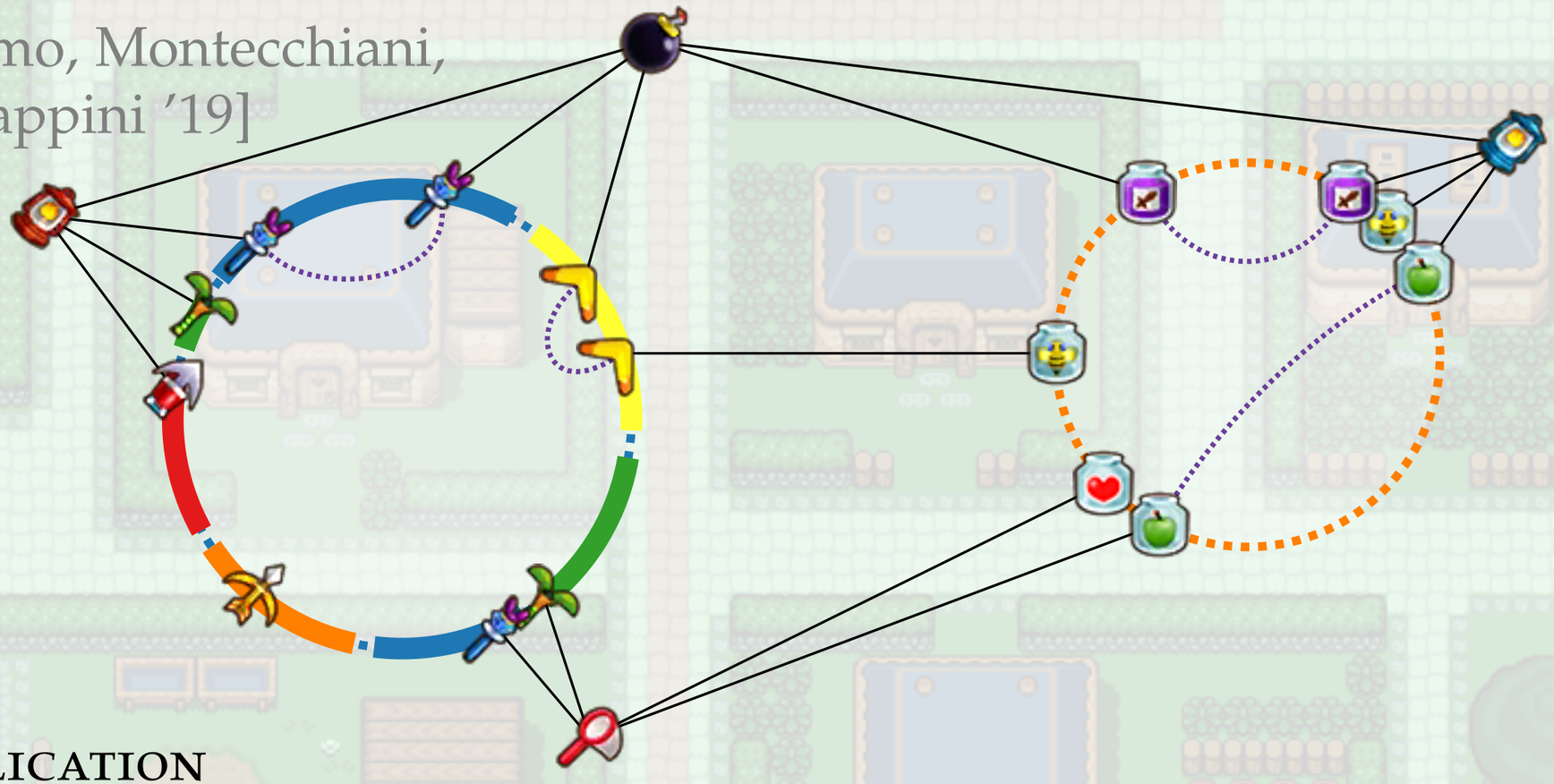
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging

ChordLink

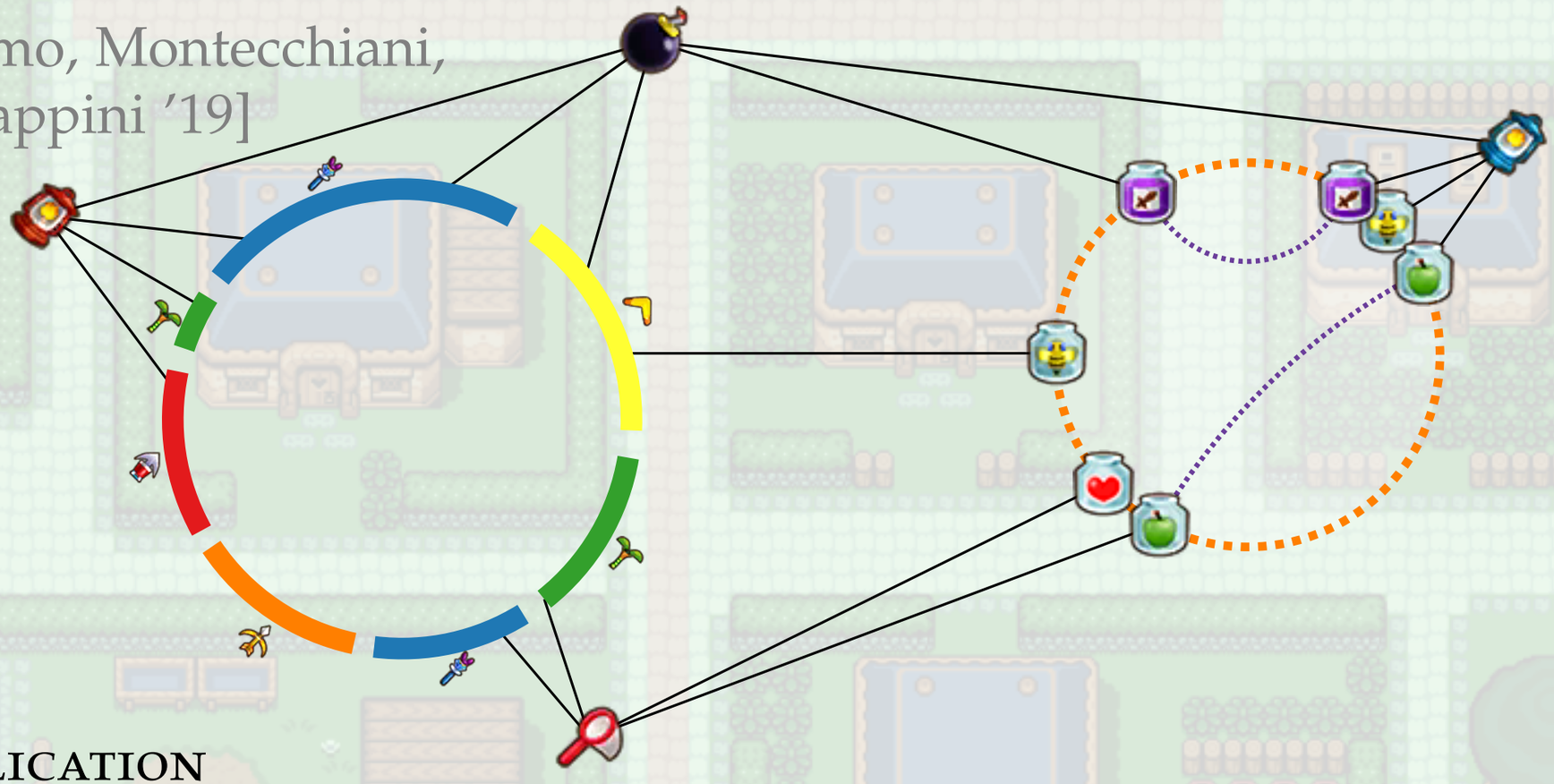
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging

ChordLink

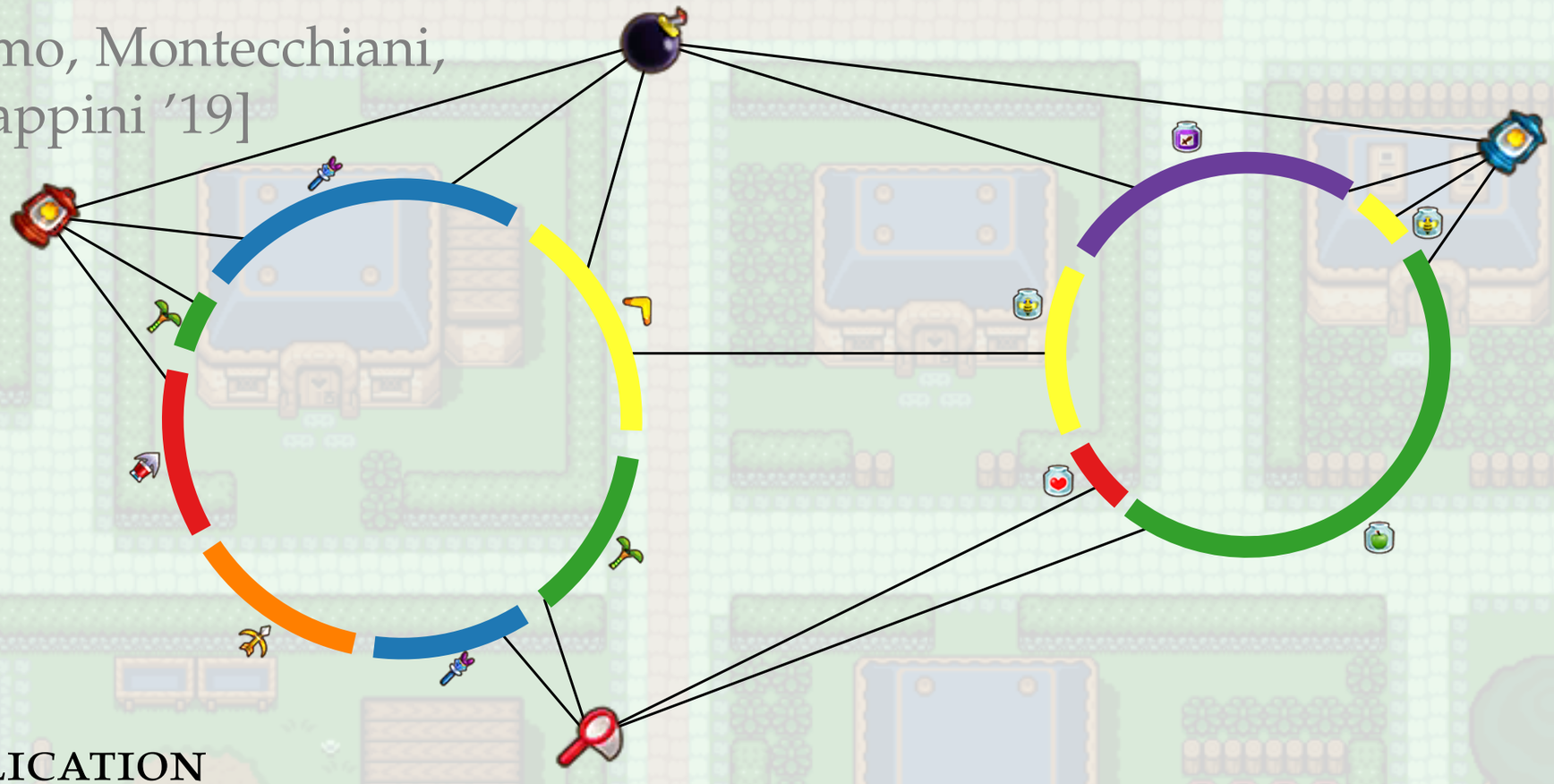
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging

ChordLink

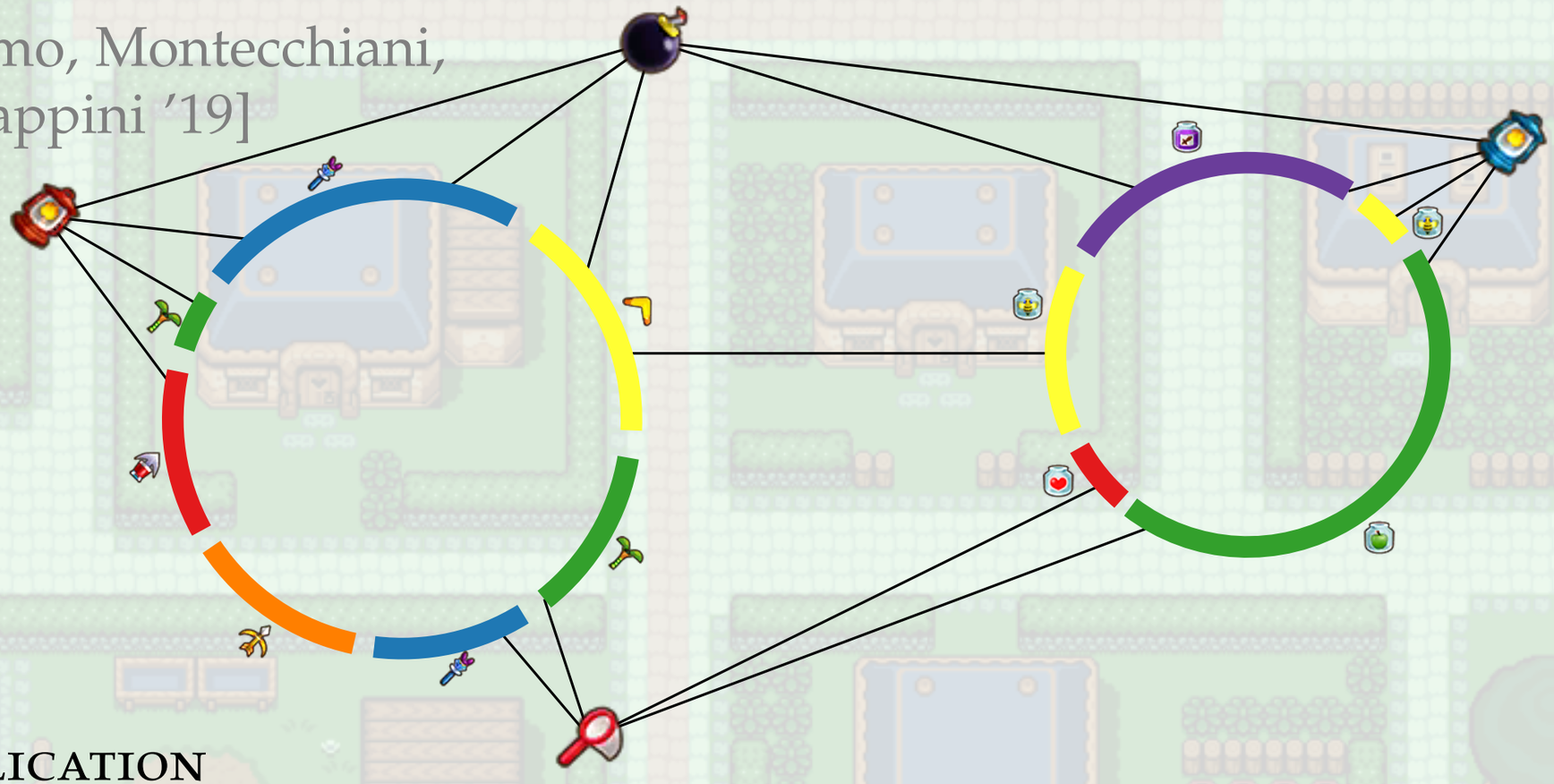
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging

ChordLink

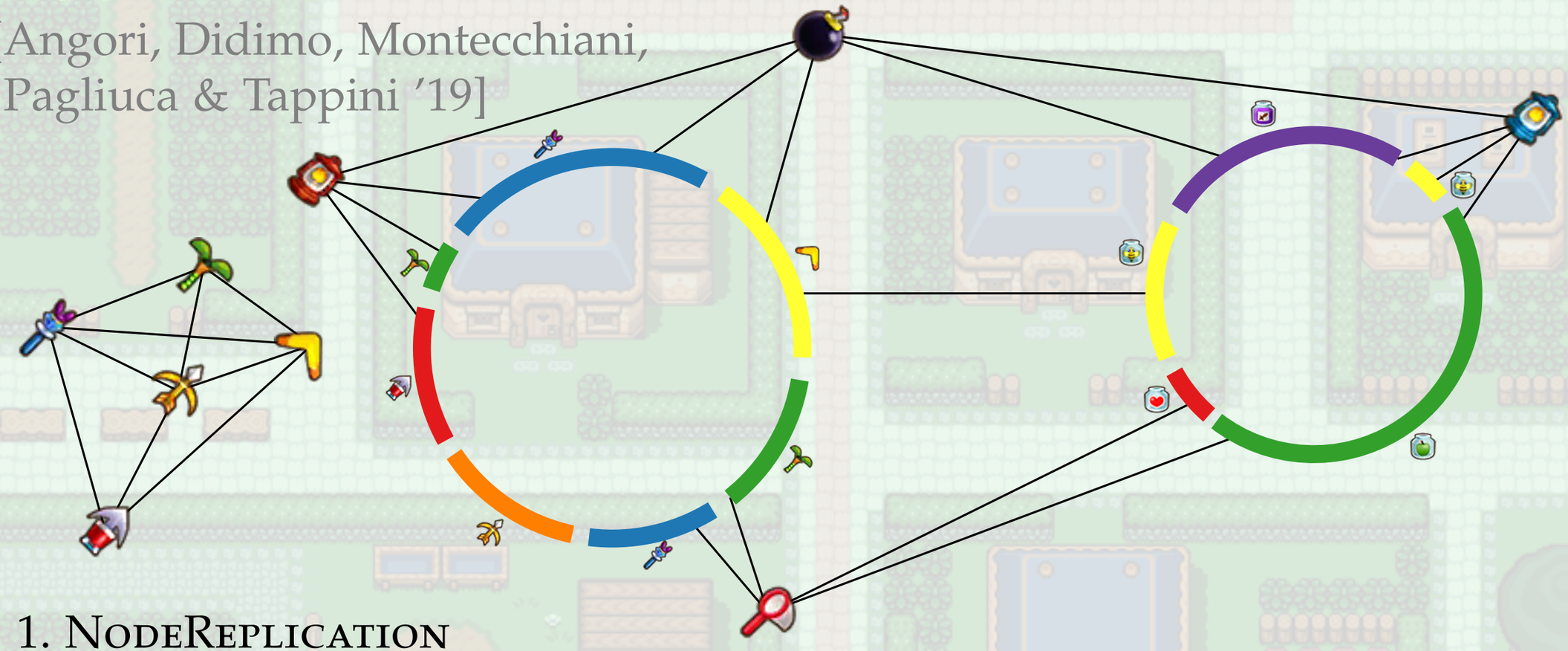
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion

ChordLink

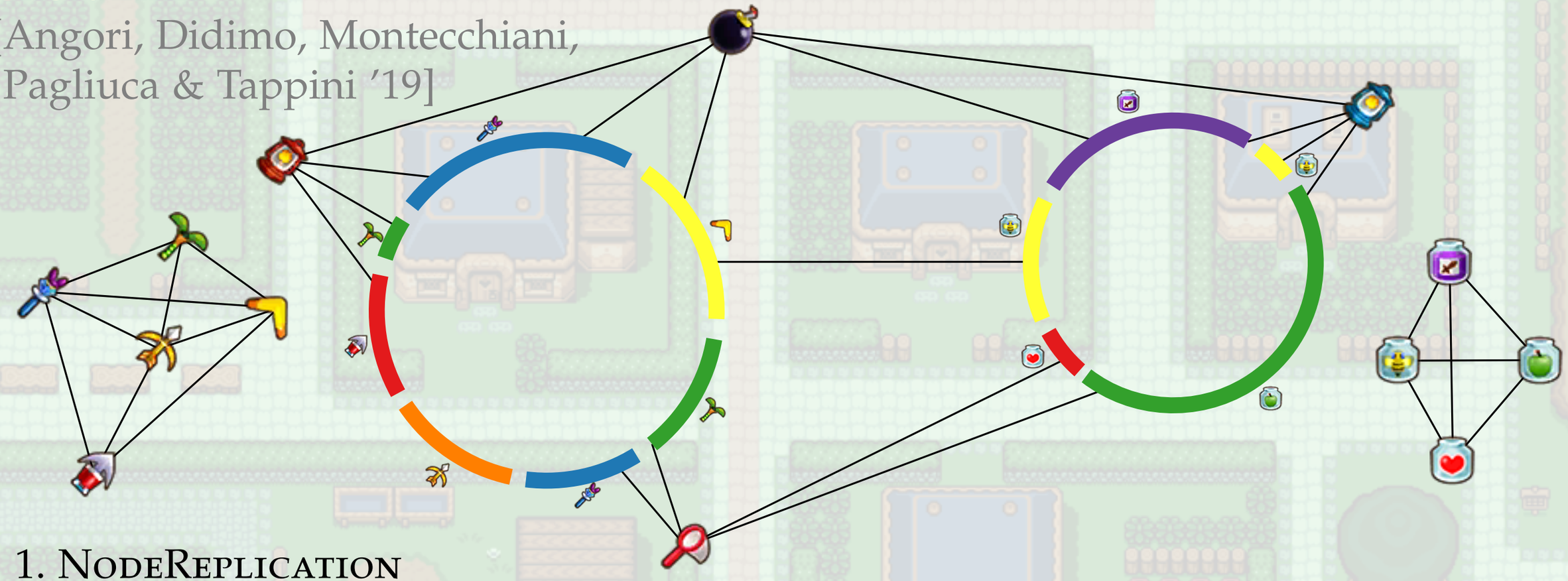
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion

ChordLink

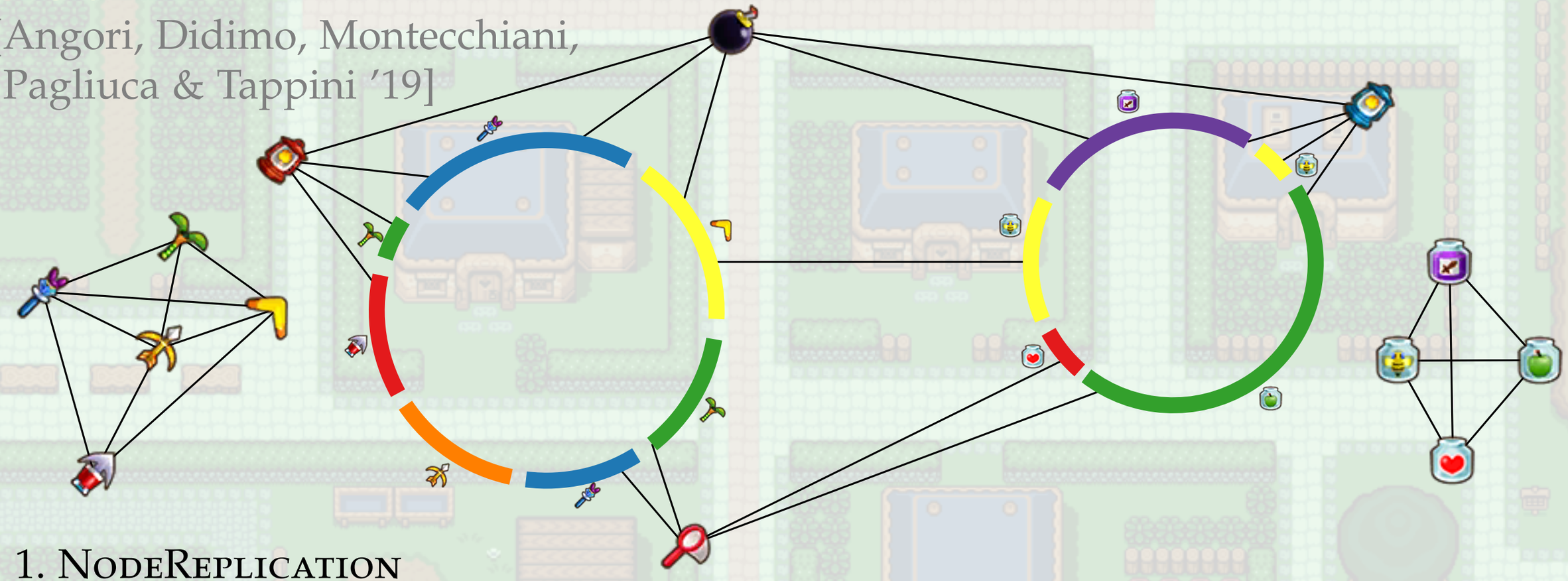
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion

ChordLink

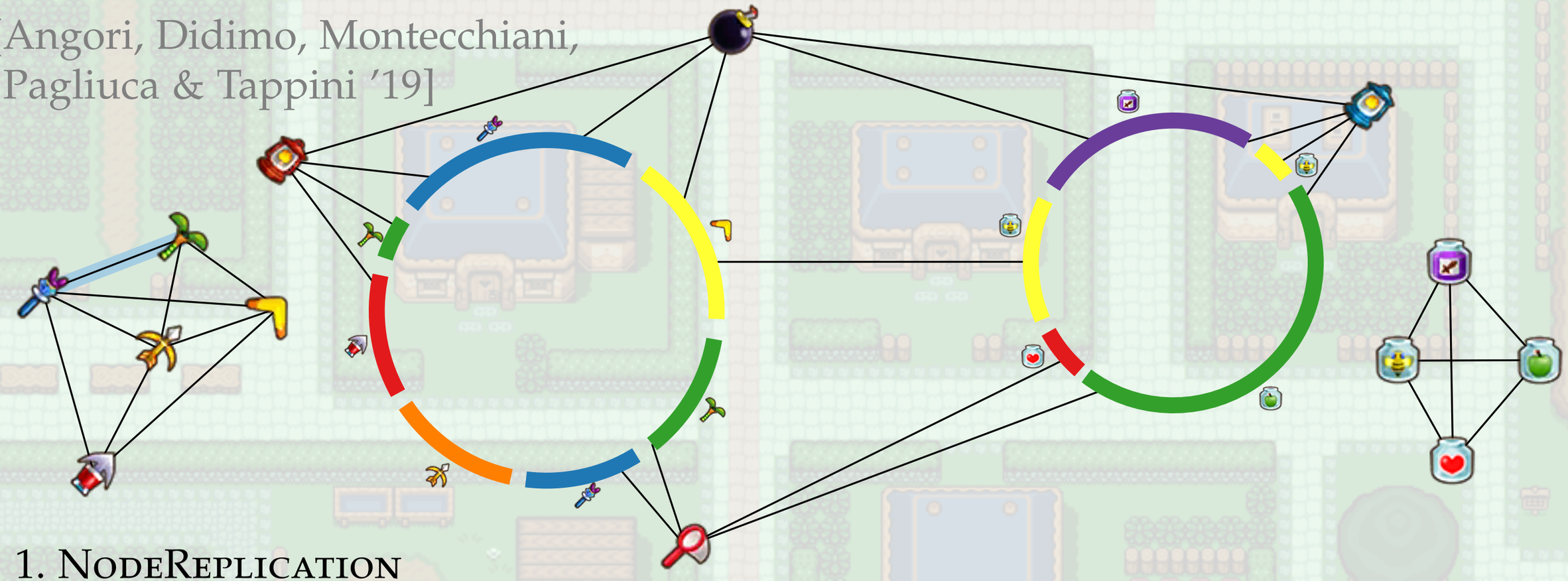
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

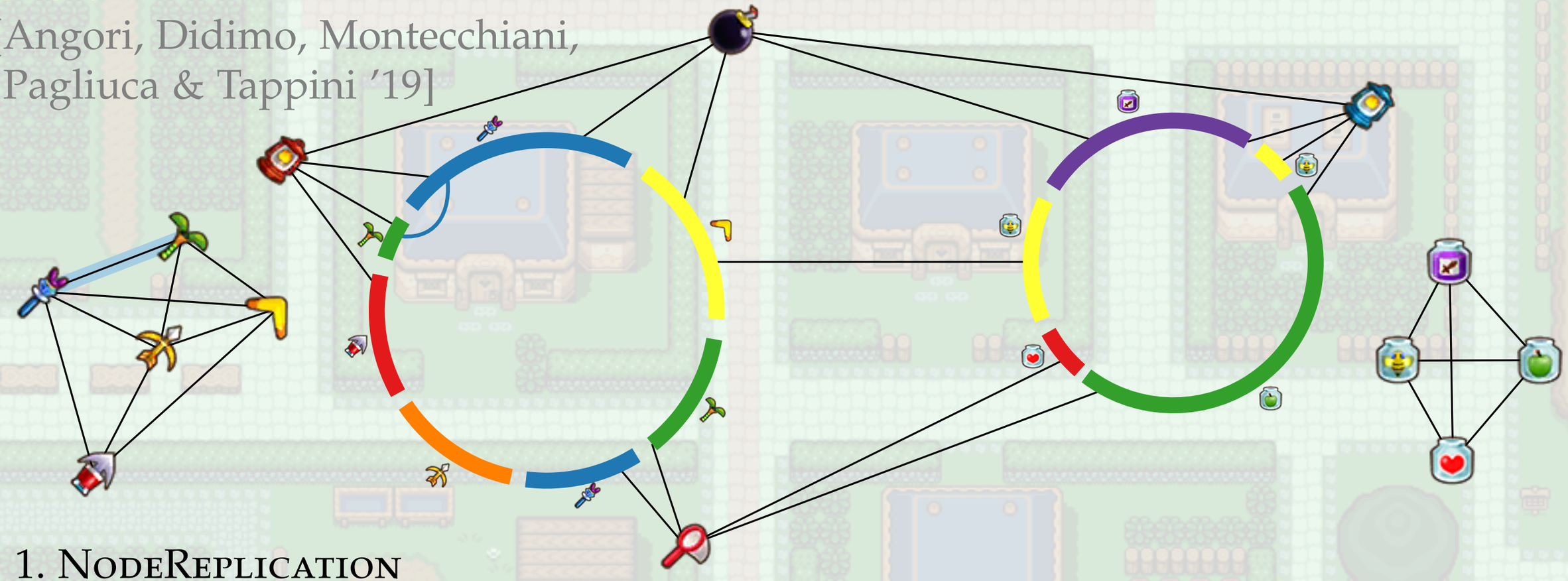
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

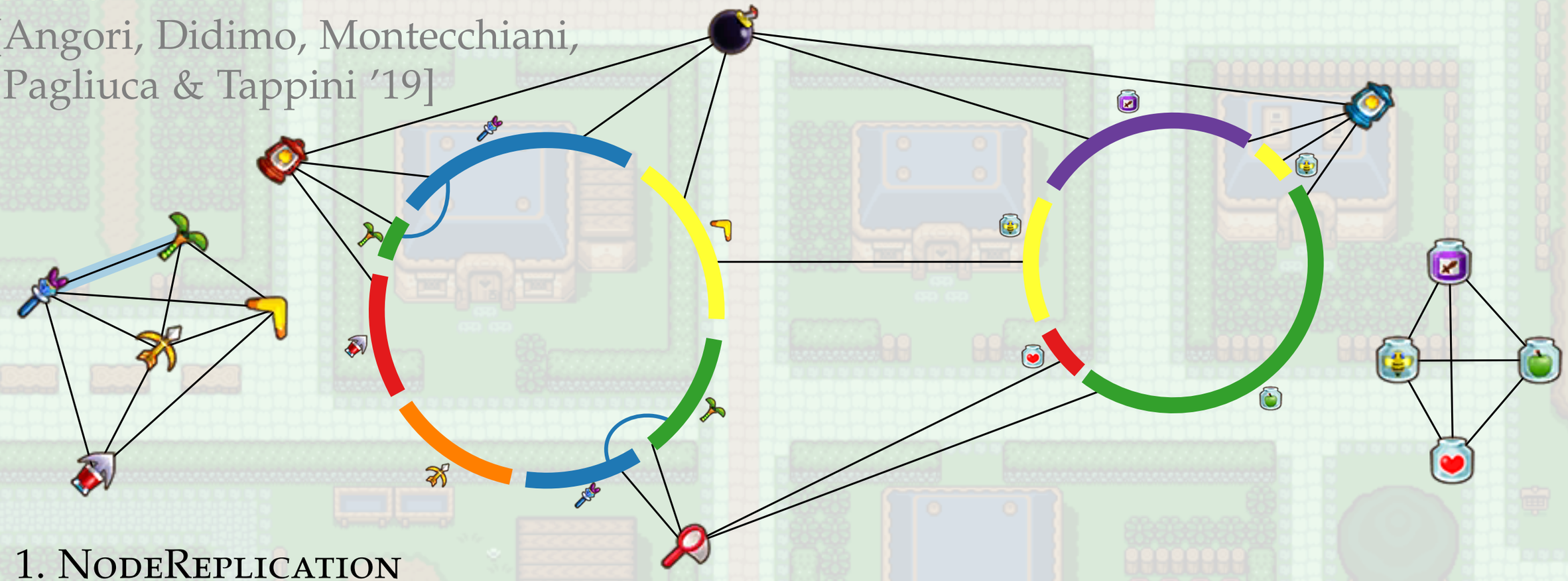
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

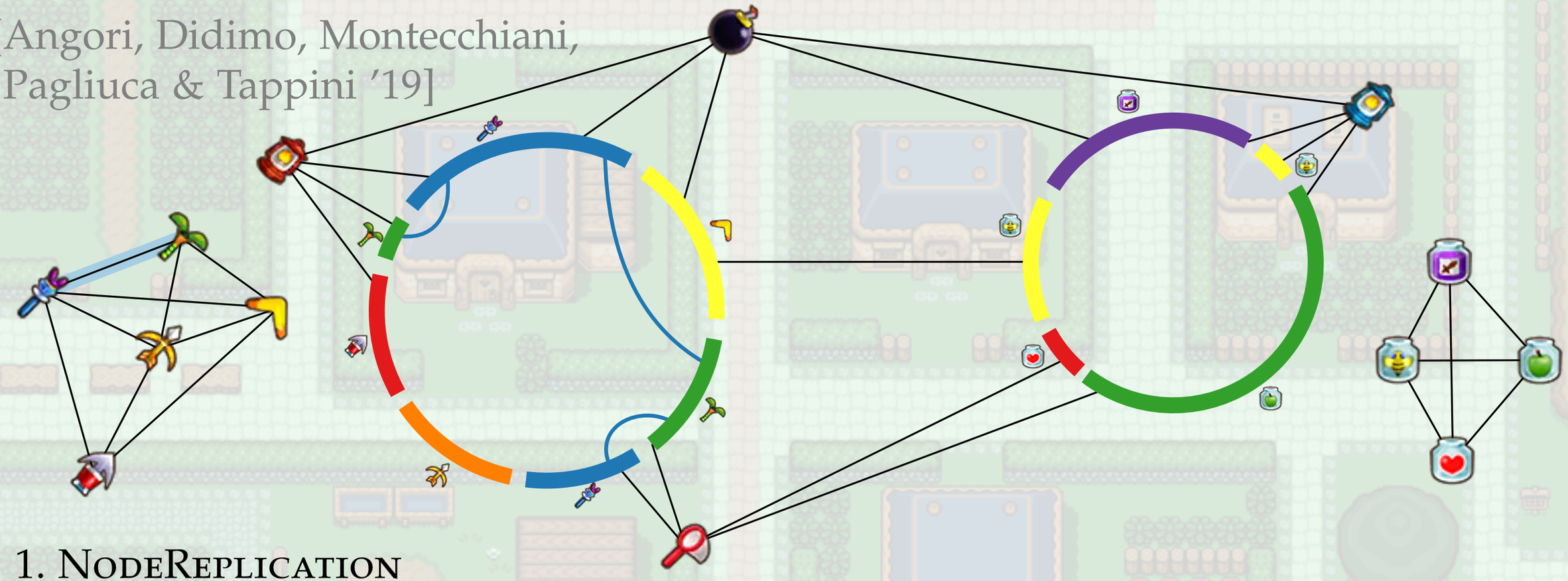
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

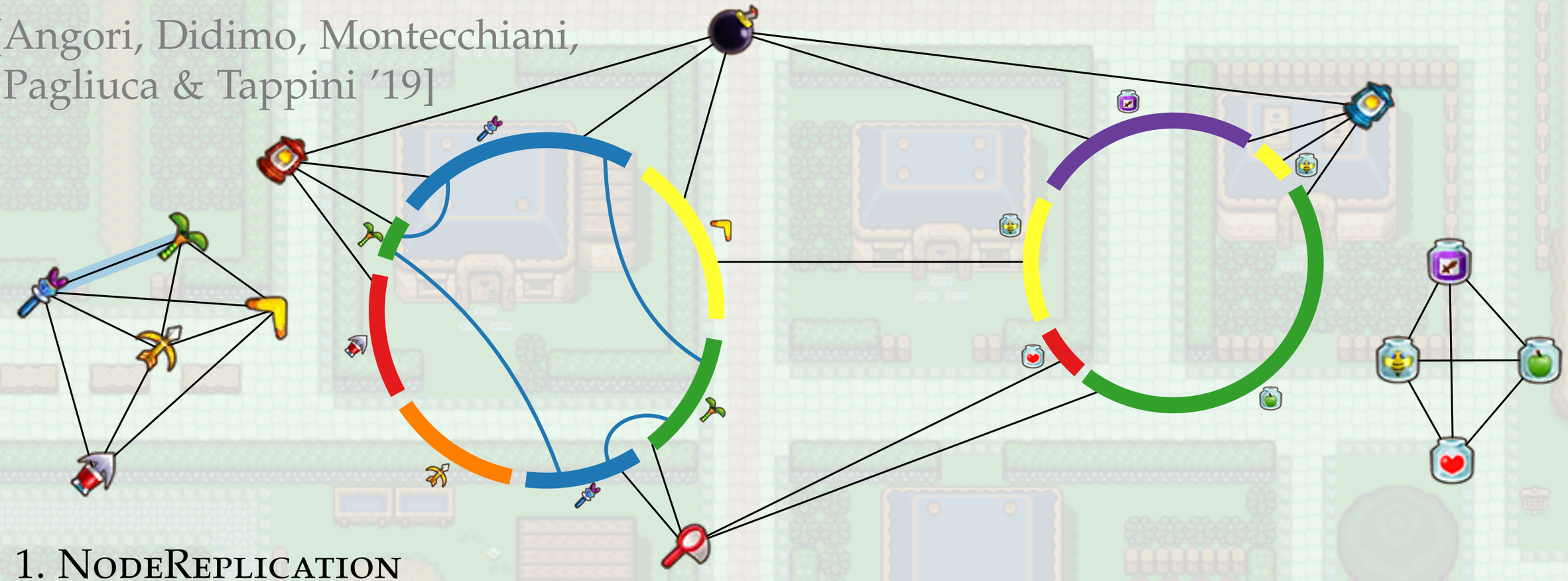
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

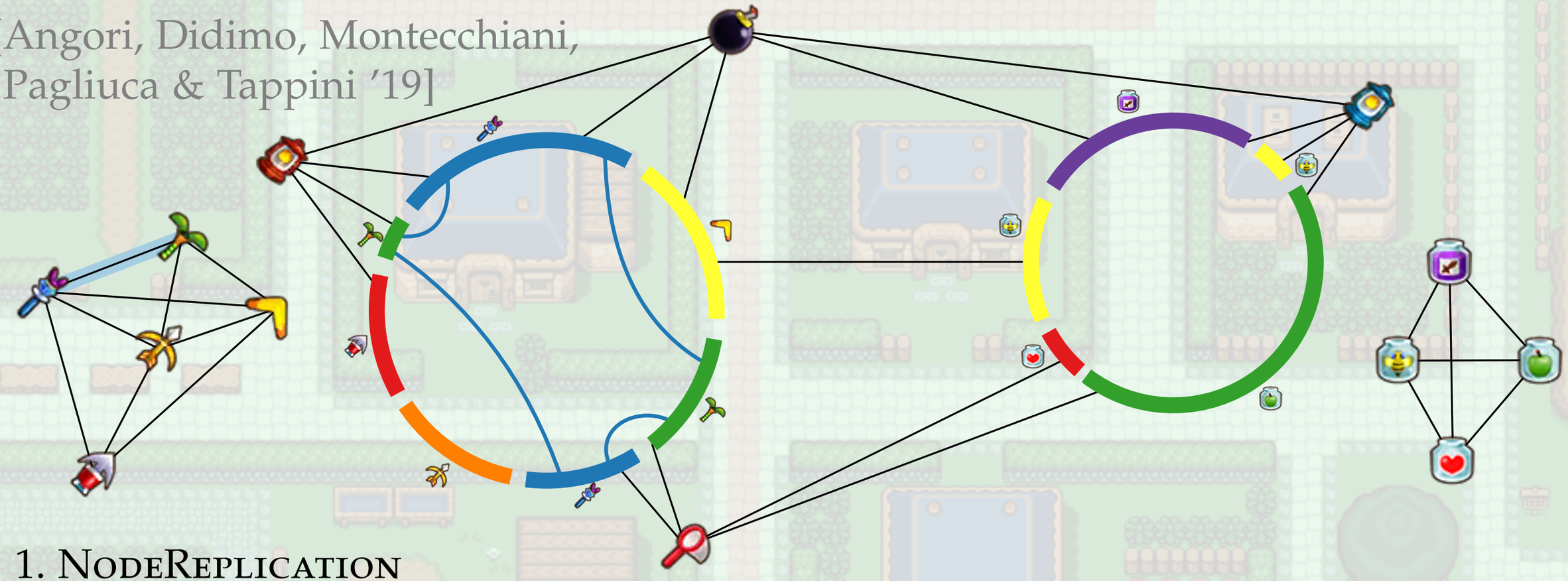
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster

ChordLink

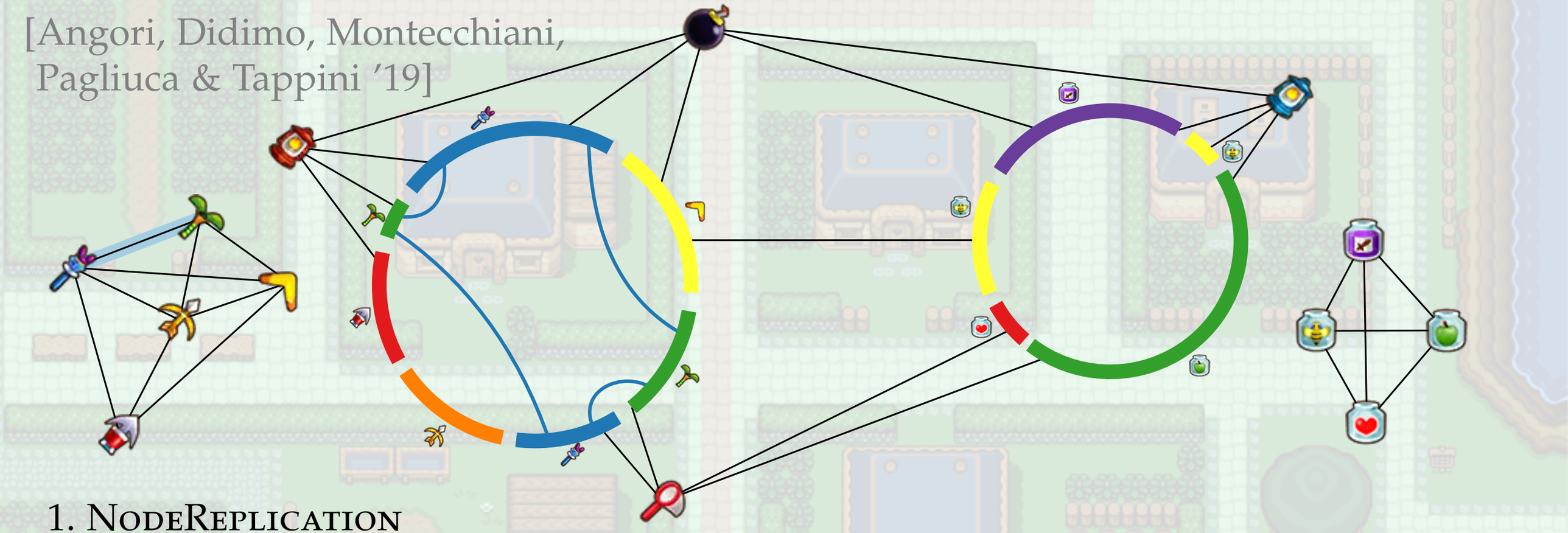
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices

ChordLink

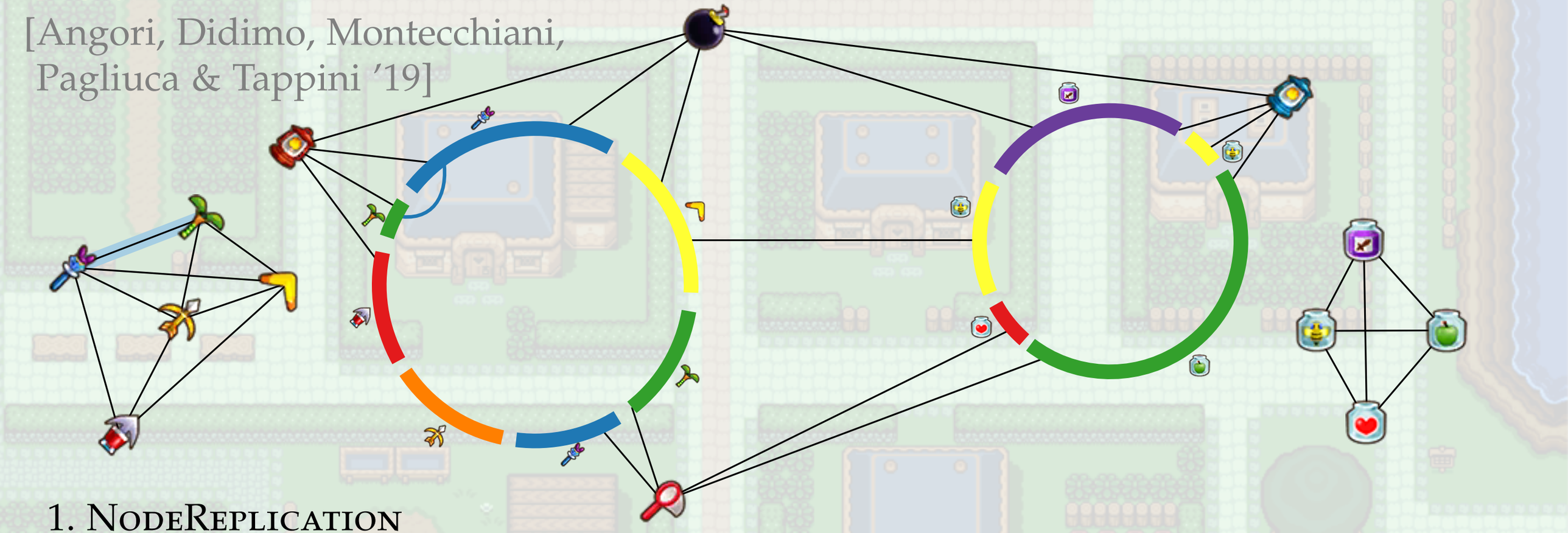
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

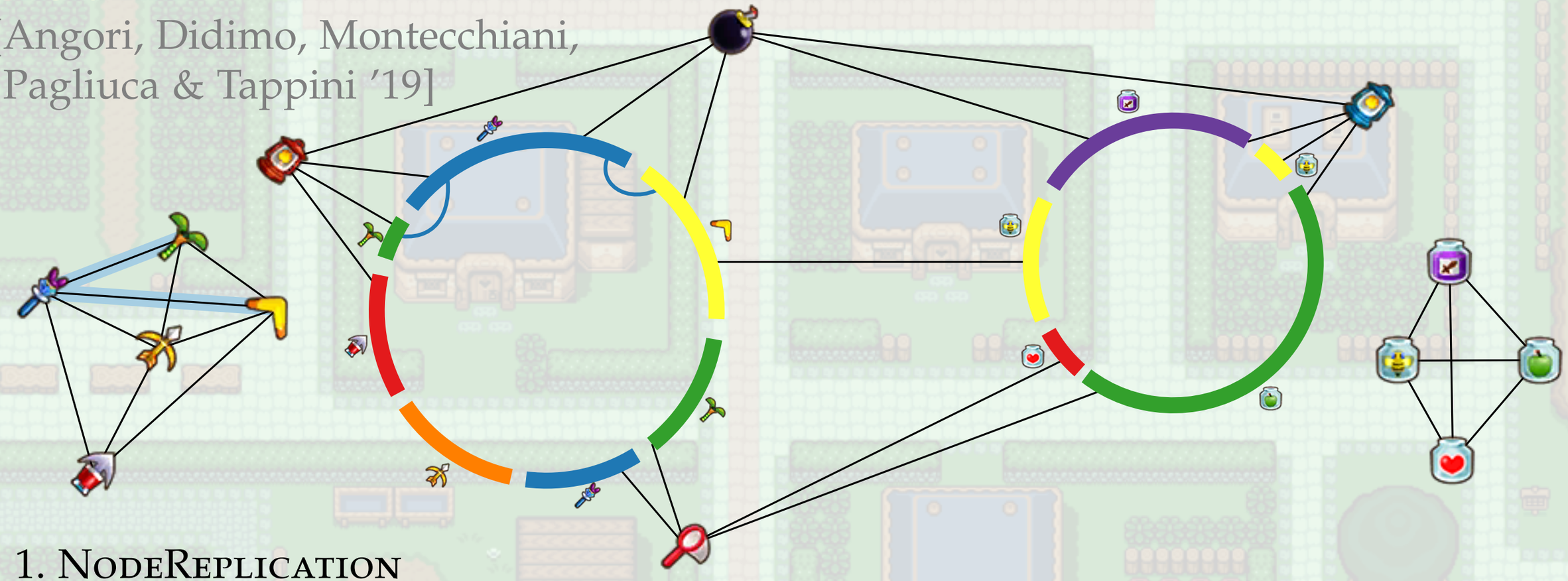
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

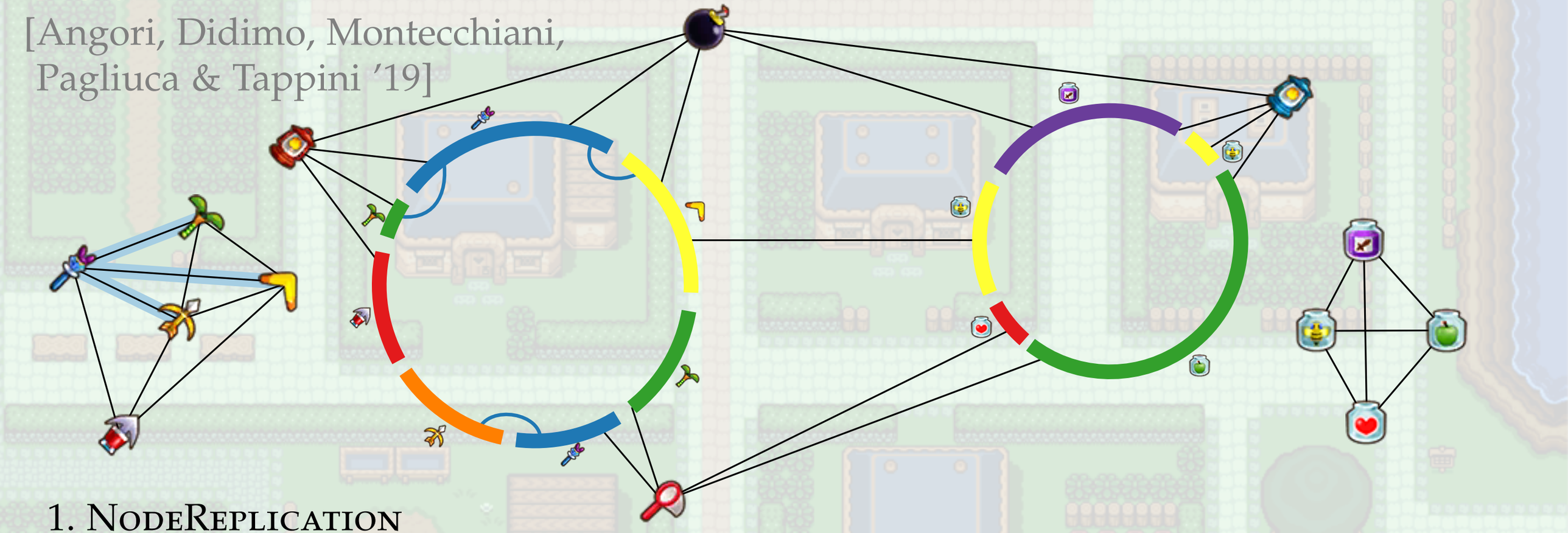
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

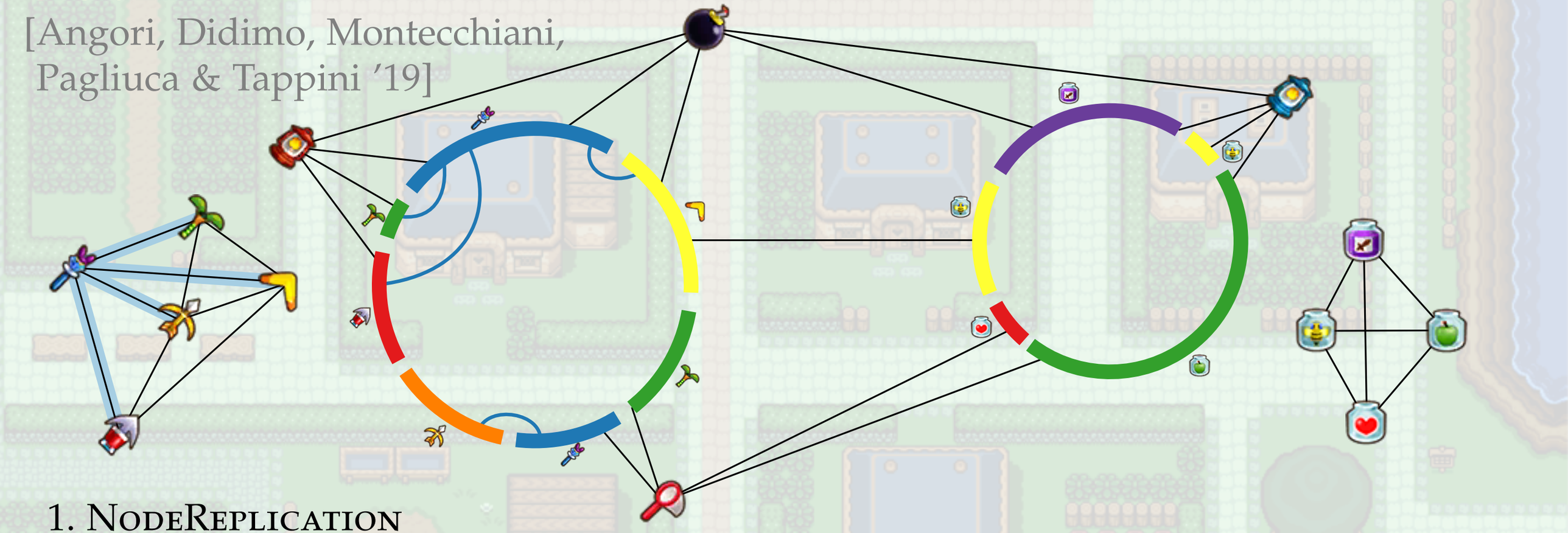
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

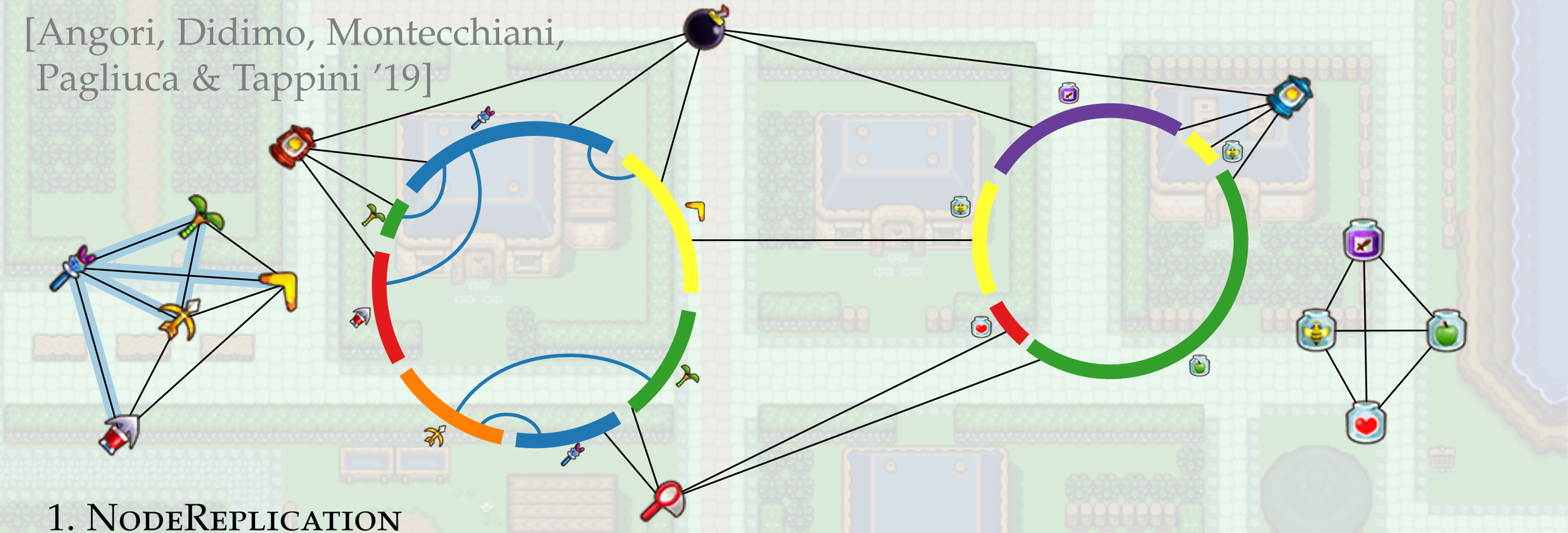
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

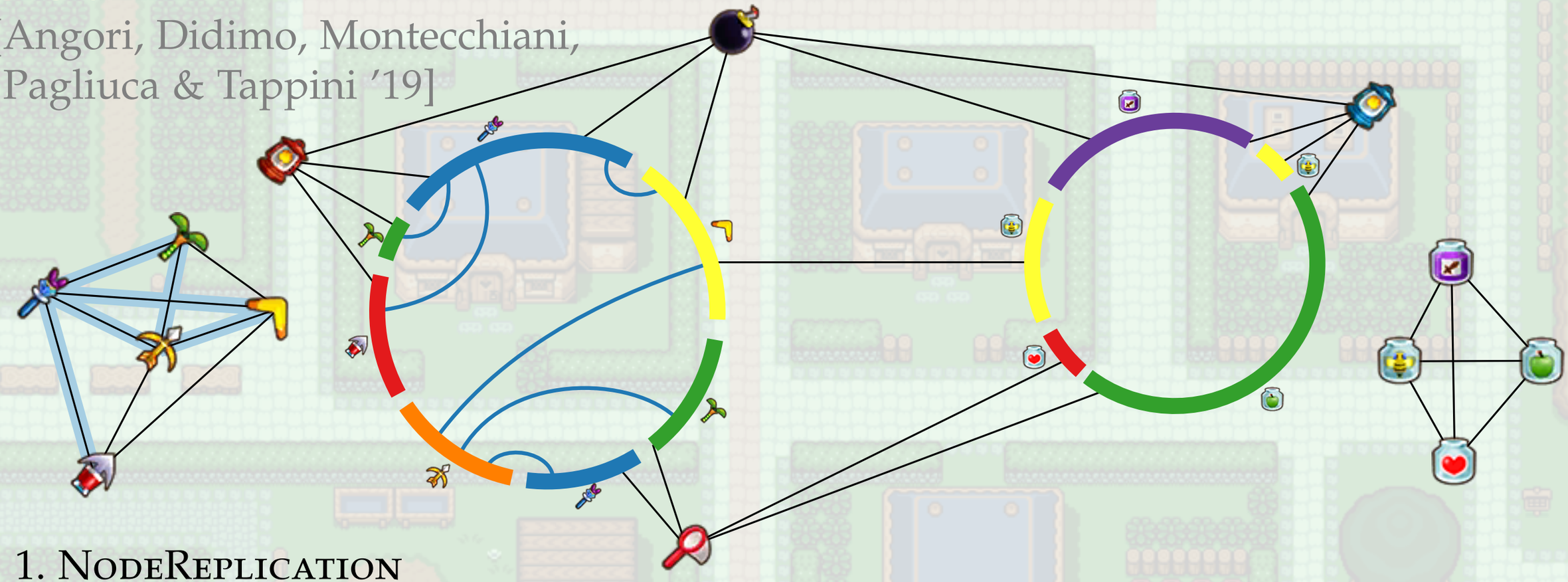
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

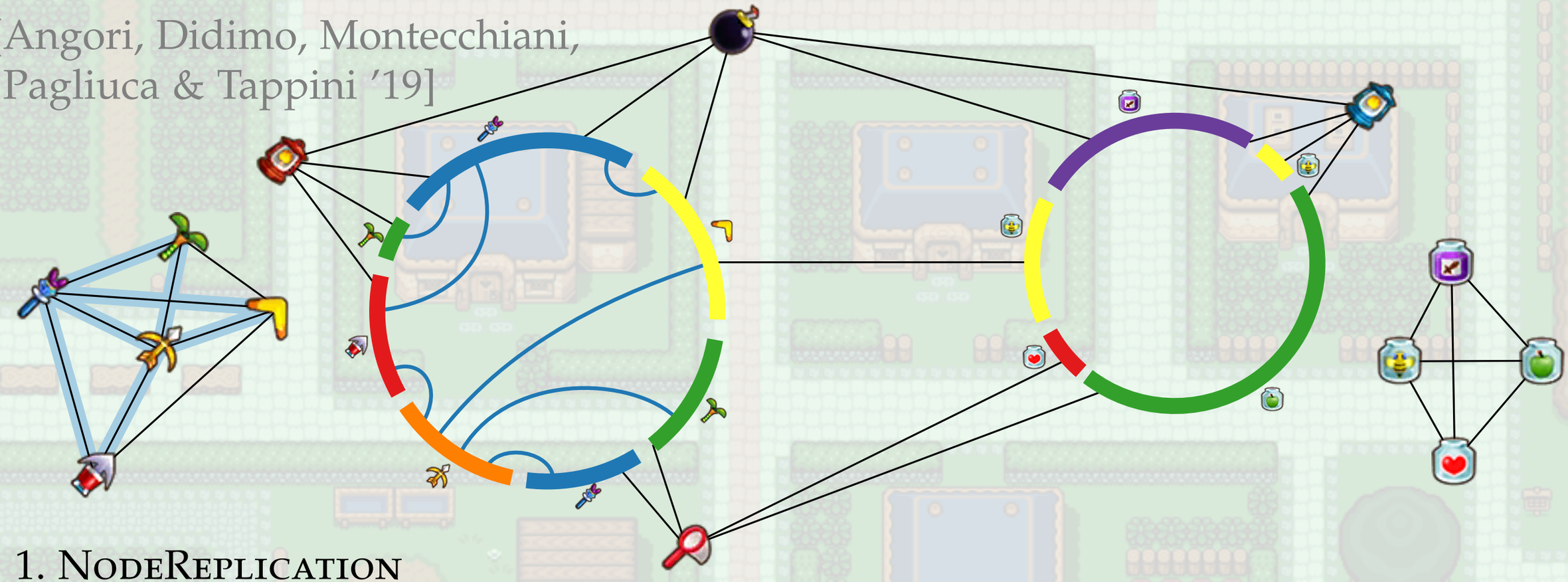
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

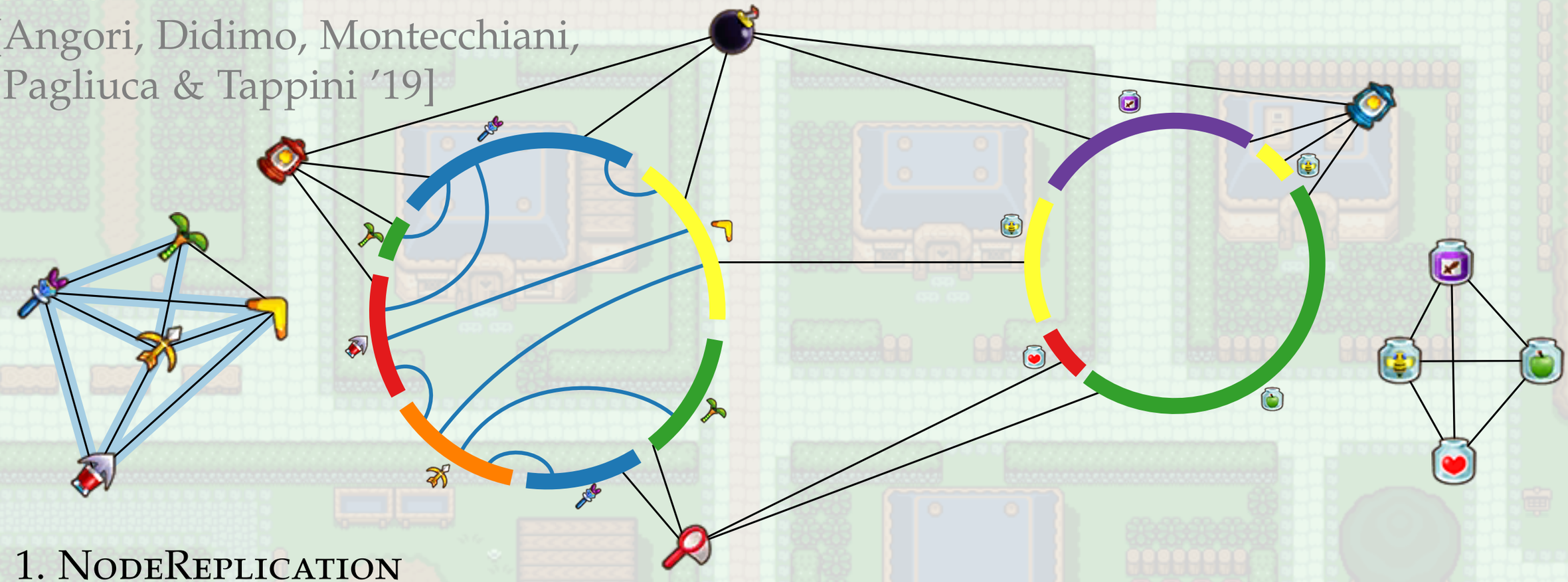
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

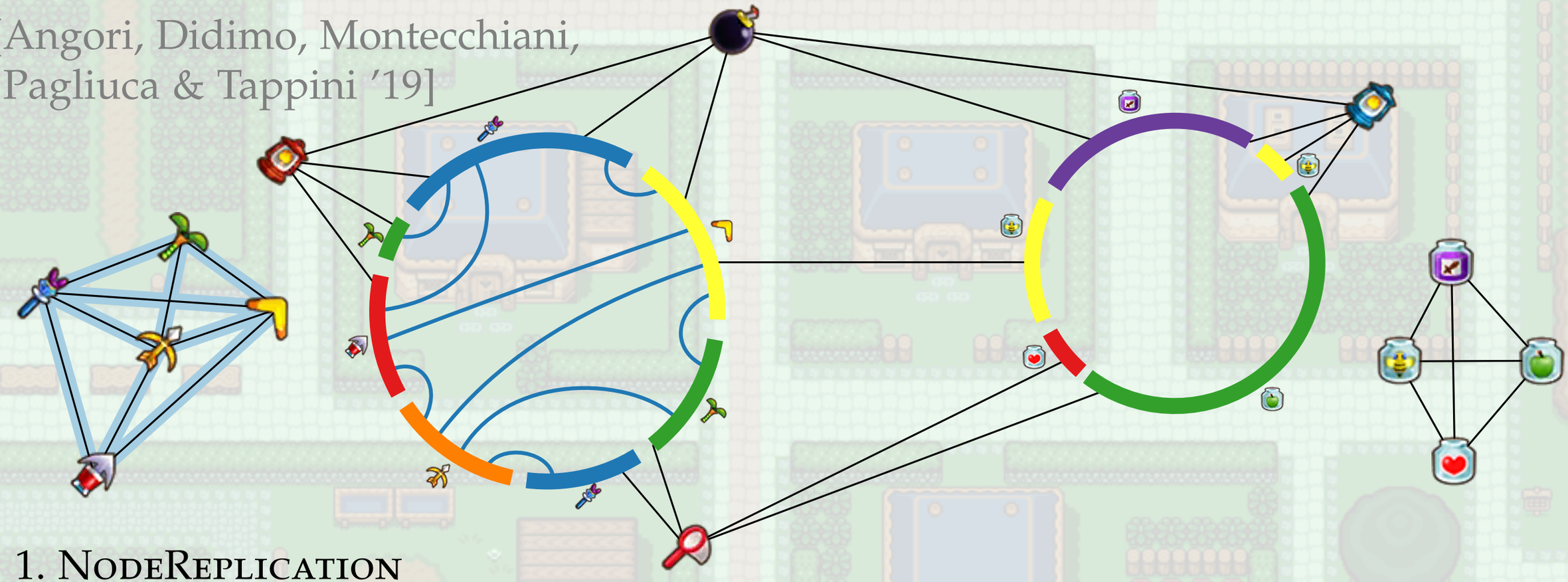
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

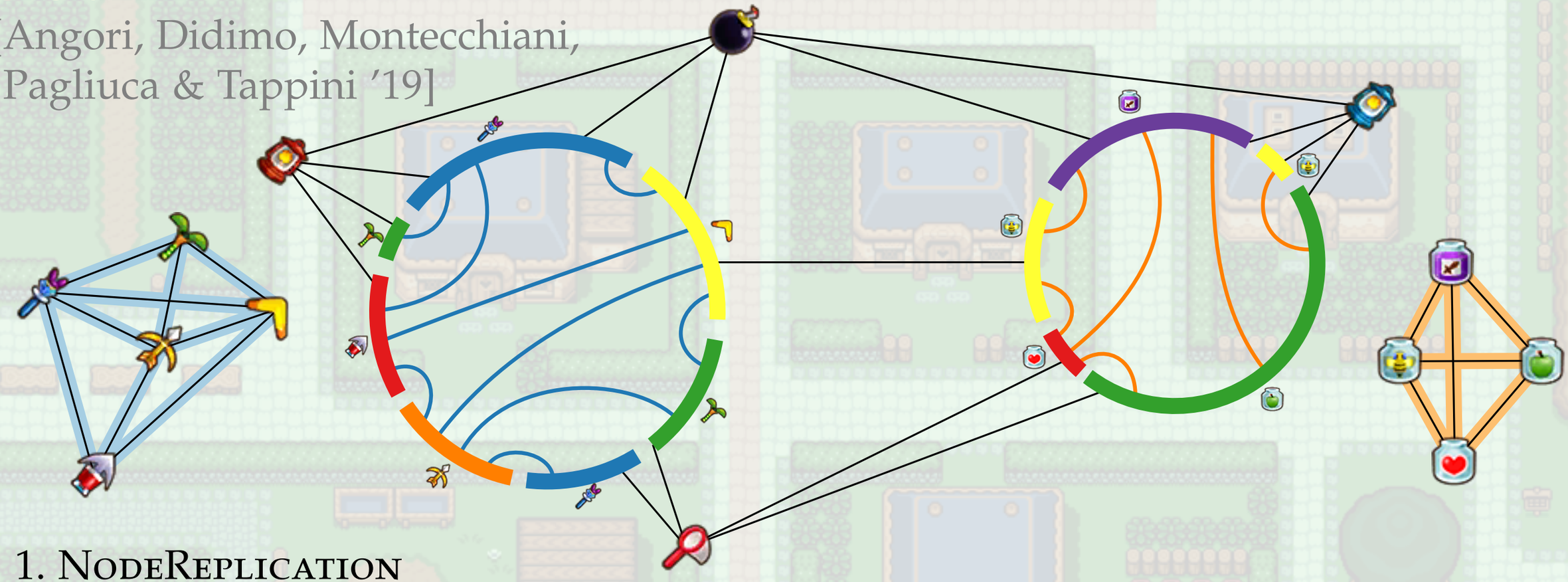
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

ChordLink

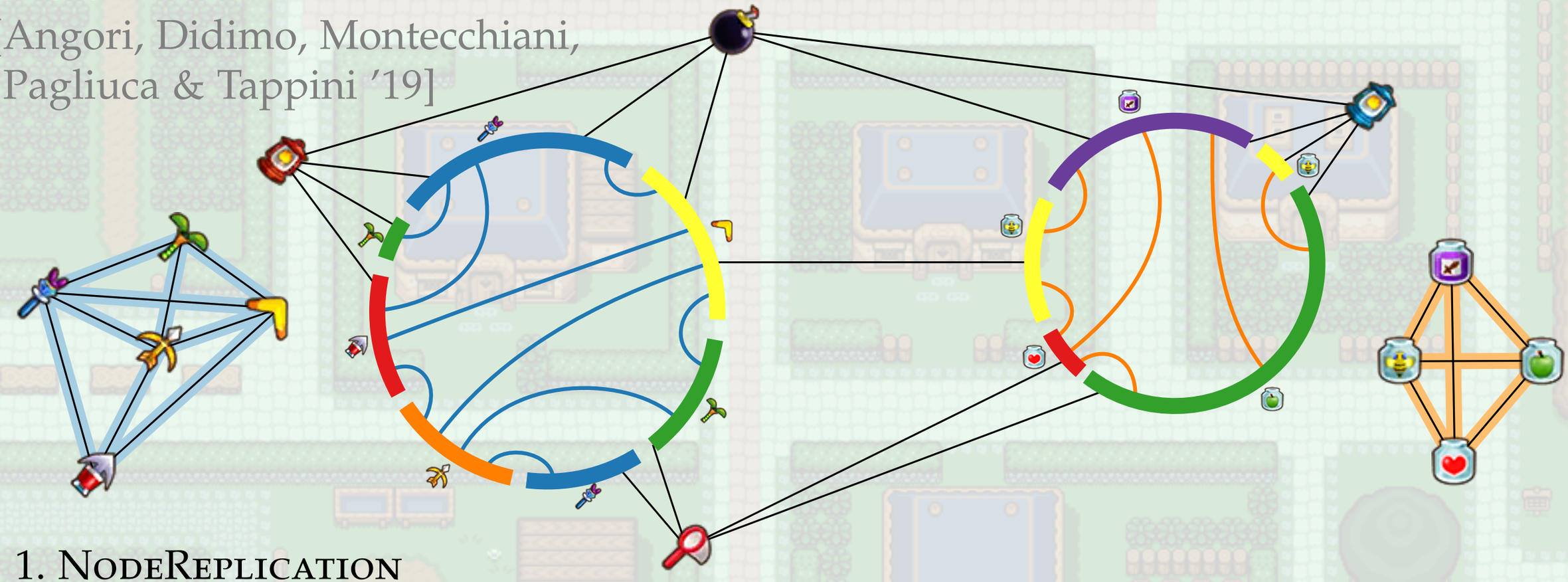
[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

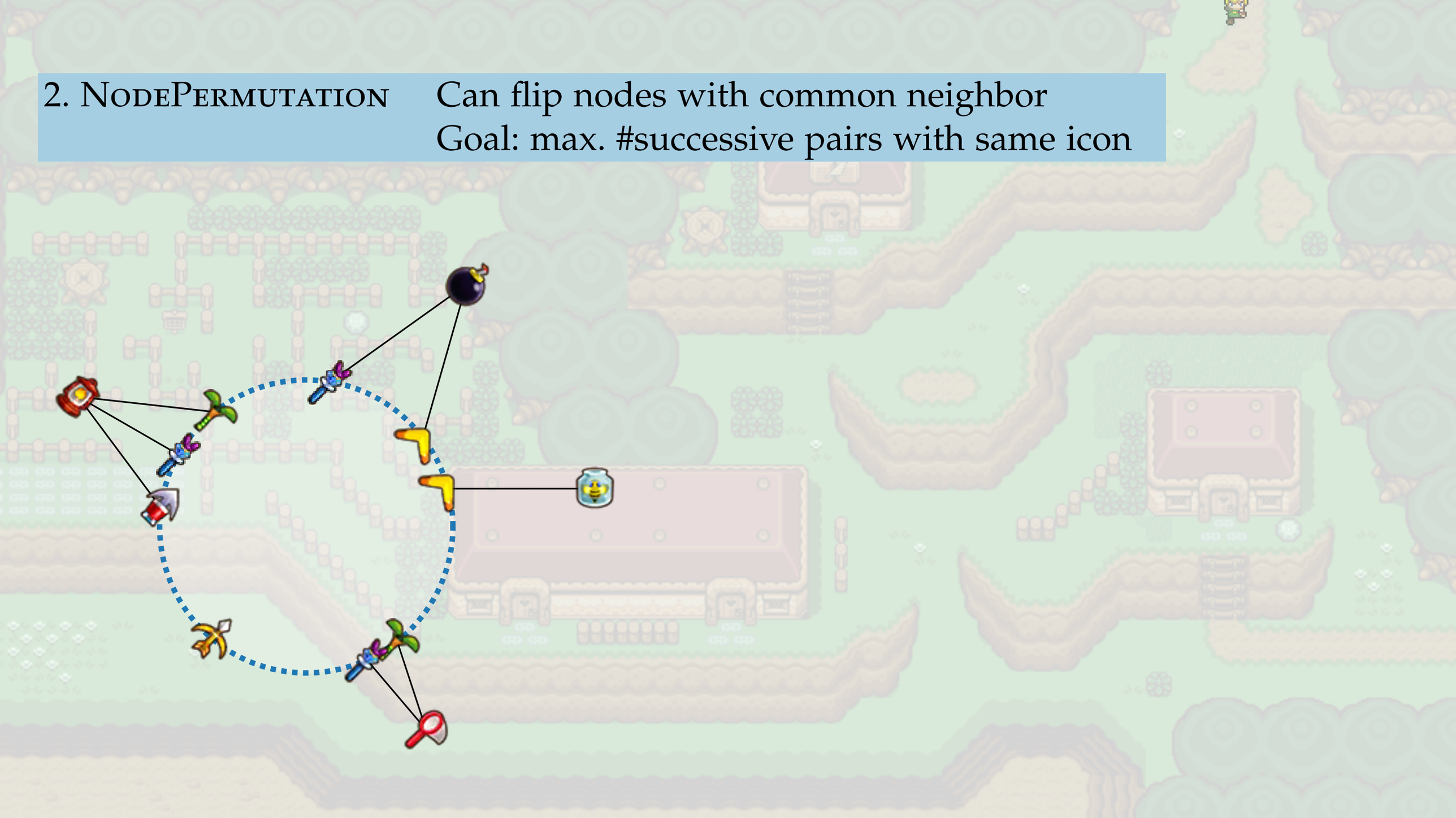
ChordLink

[Angori, Didimo, Montecchiani, Pagliuca & Tappini '19]



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings

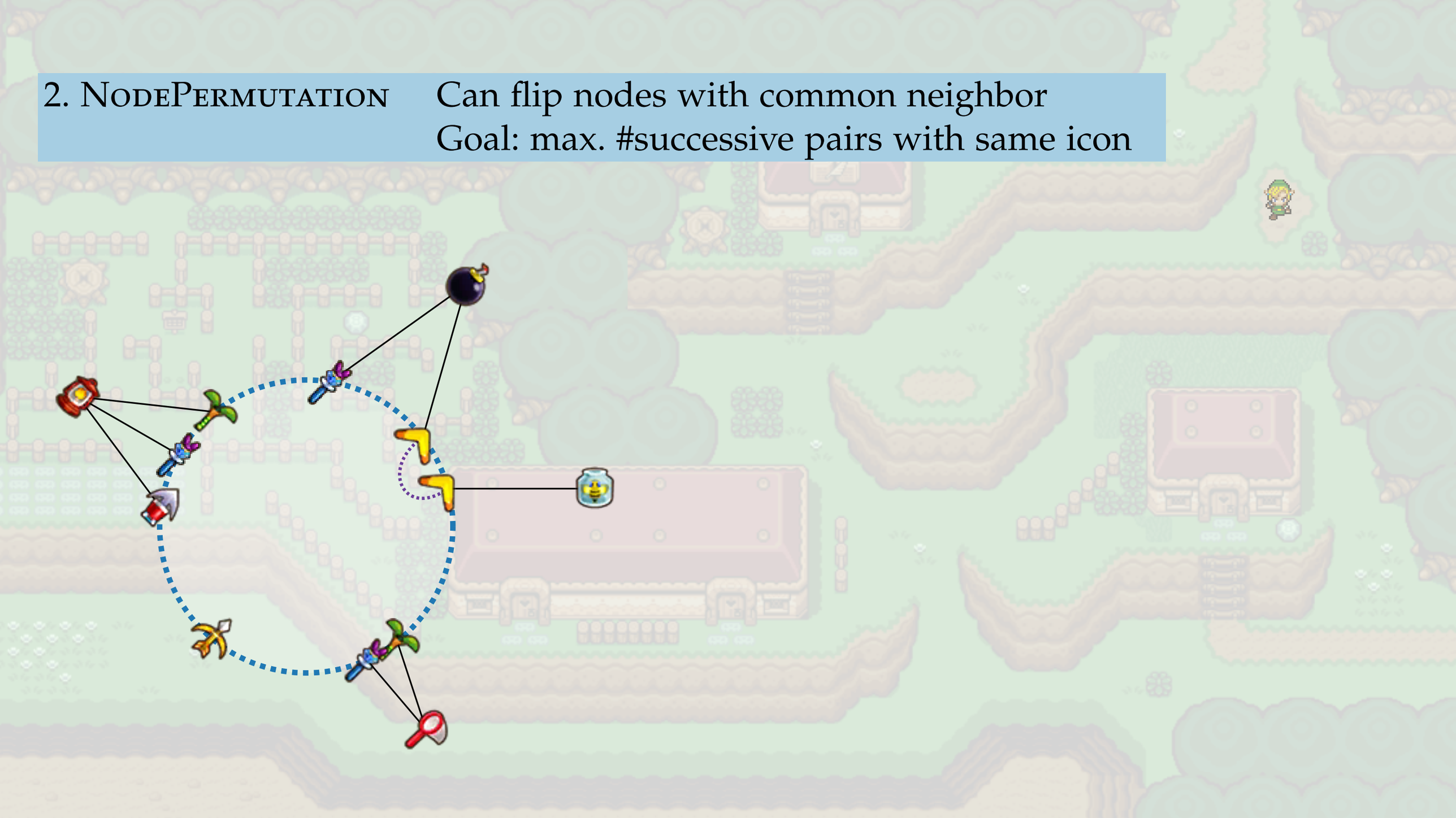
2. NodePermutation Can flip nodes with common neighbor
Goal: max. #successive pairs with same icon



2. NodePermutation

Can flip nodes with common neighbor

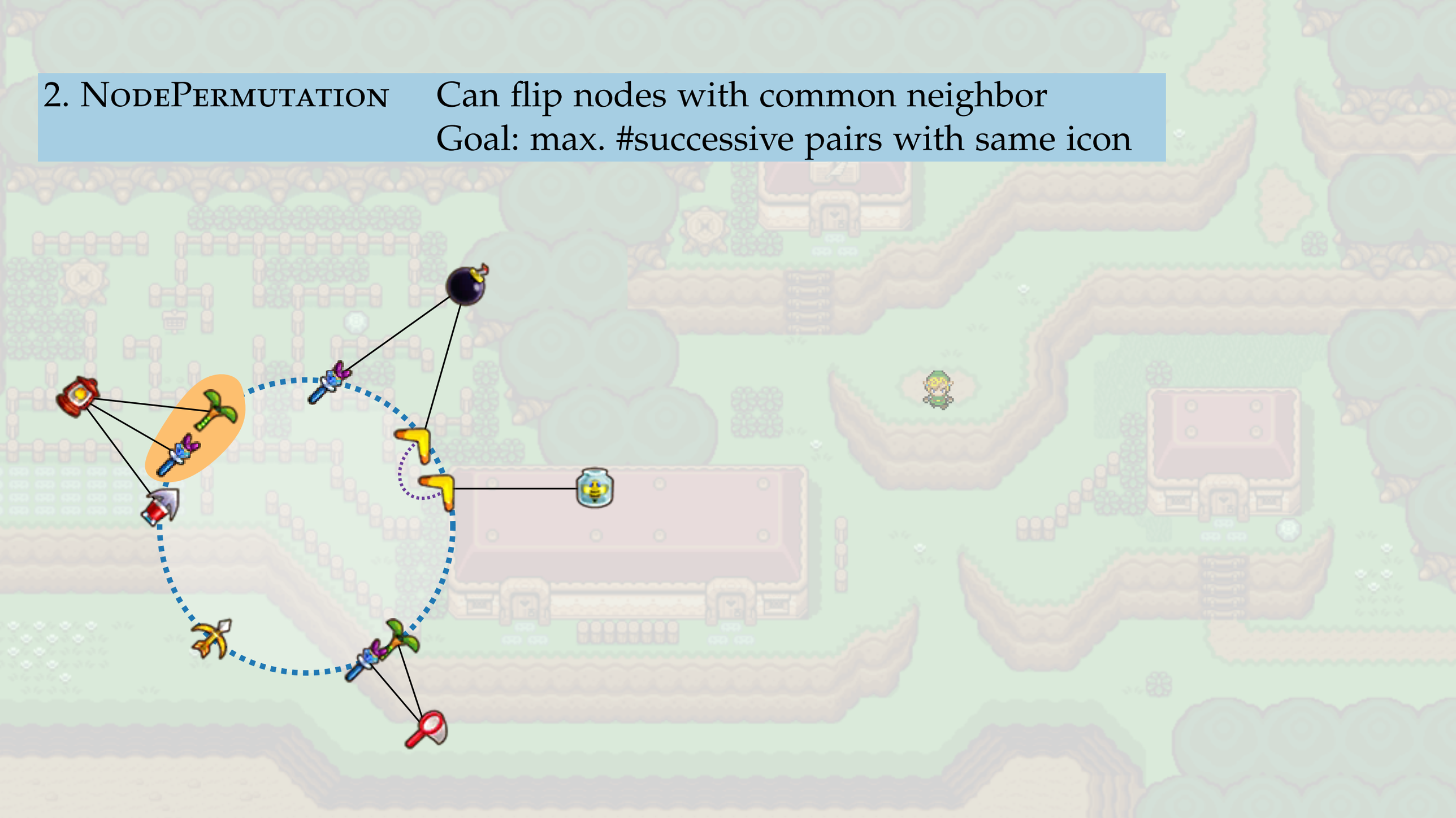
Goal: max. #successive pairs with same icon



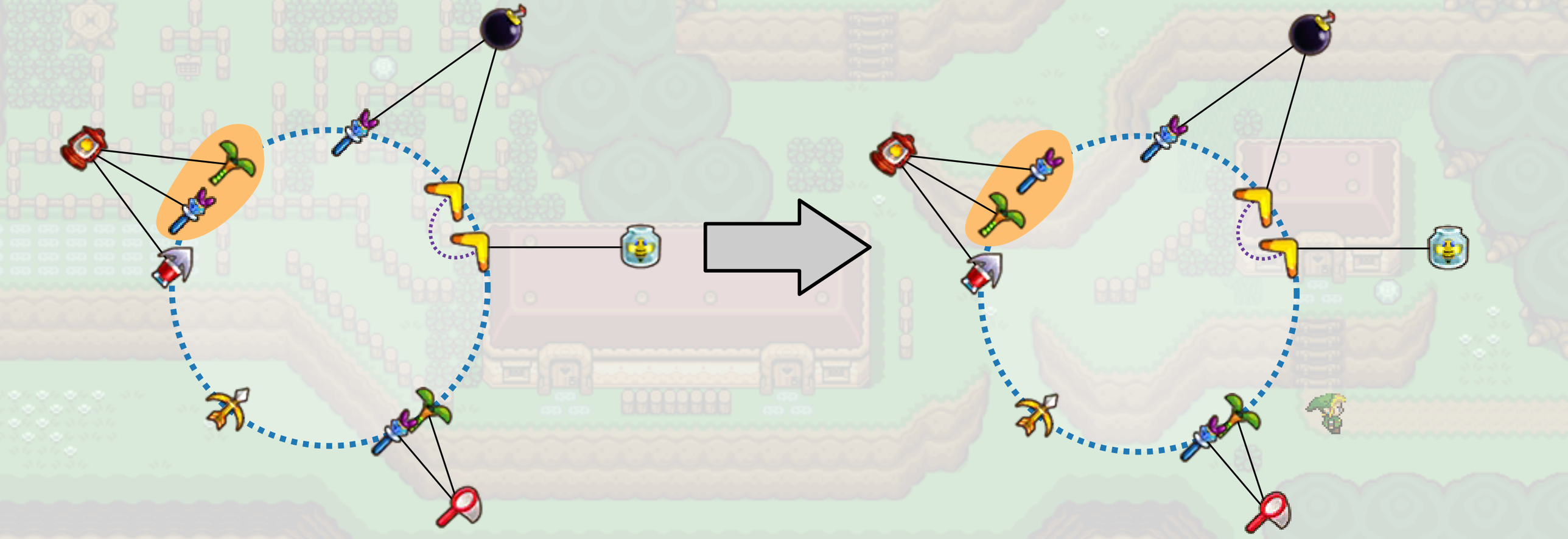
2. NodePermutation

Can flip nodes with common neighbor

Goal: max. #successive pairs with same icon



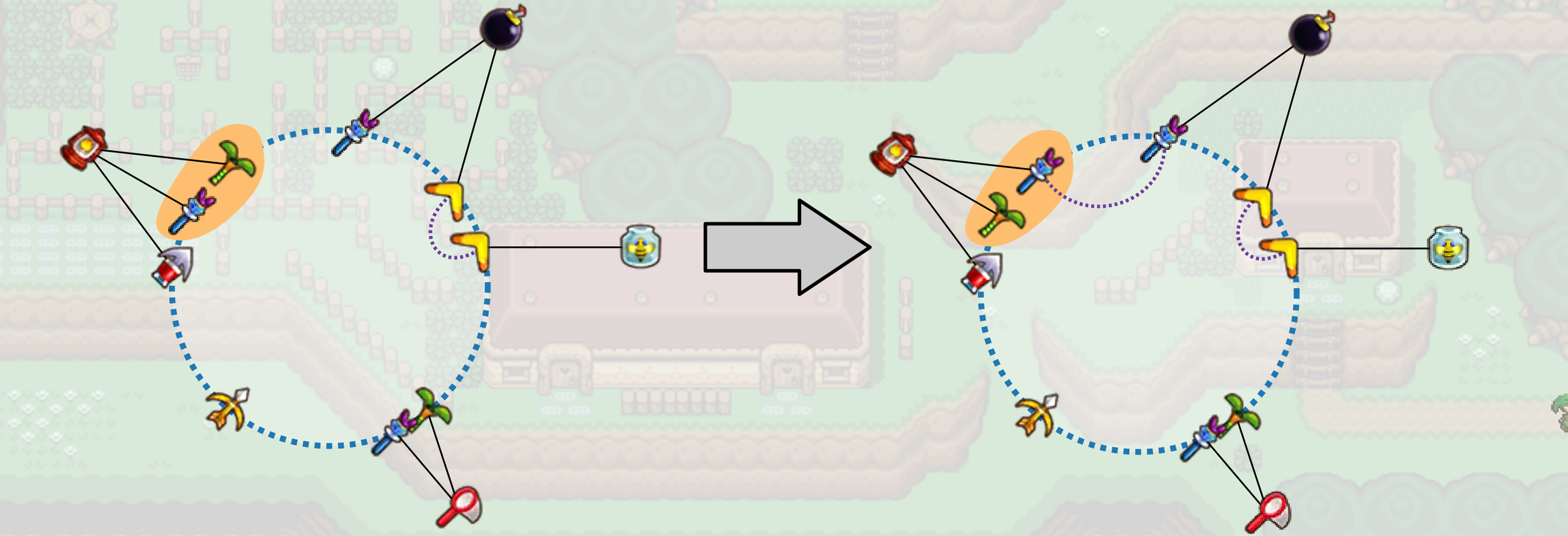
2. NodePermutation Can flip nodes with common neighbor
Goal: max. #successive pairs with same icon

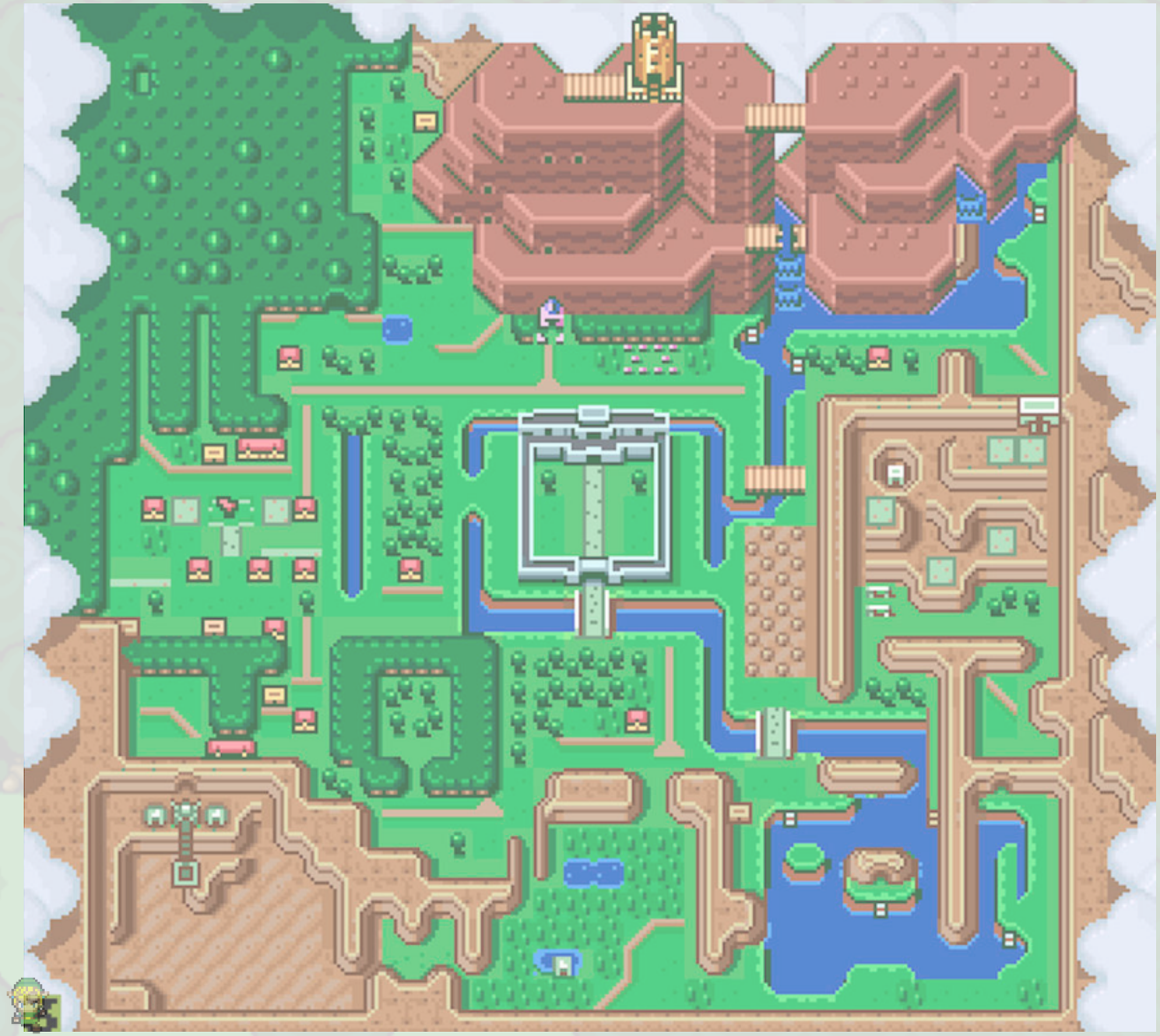


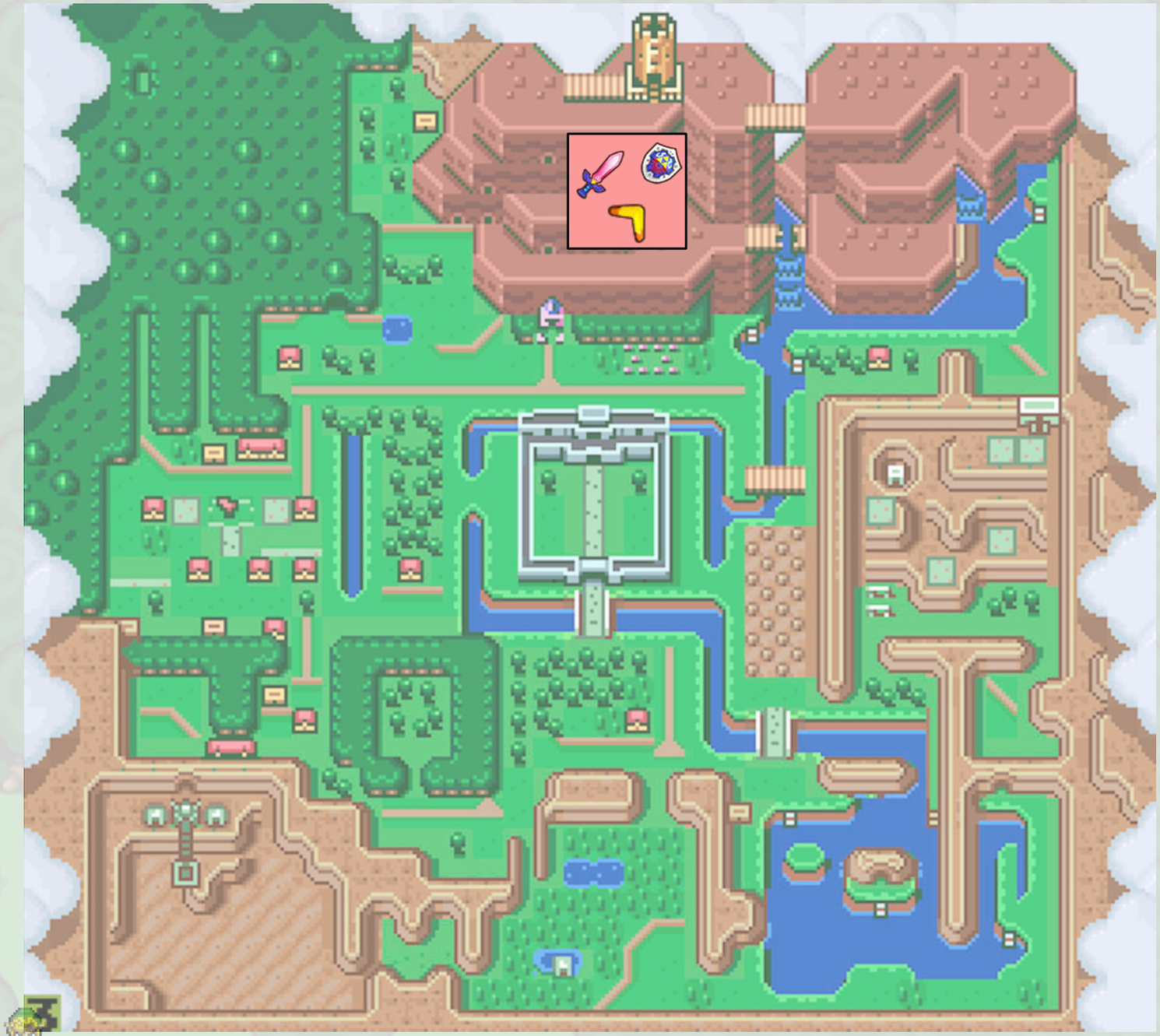
2. NodePermutation

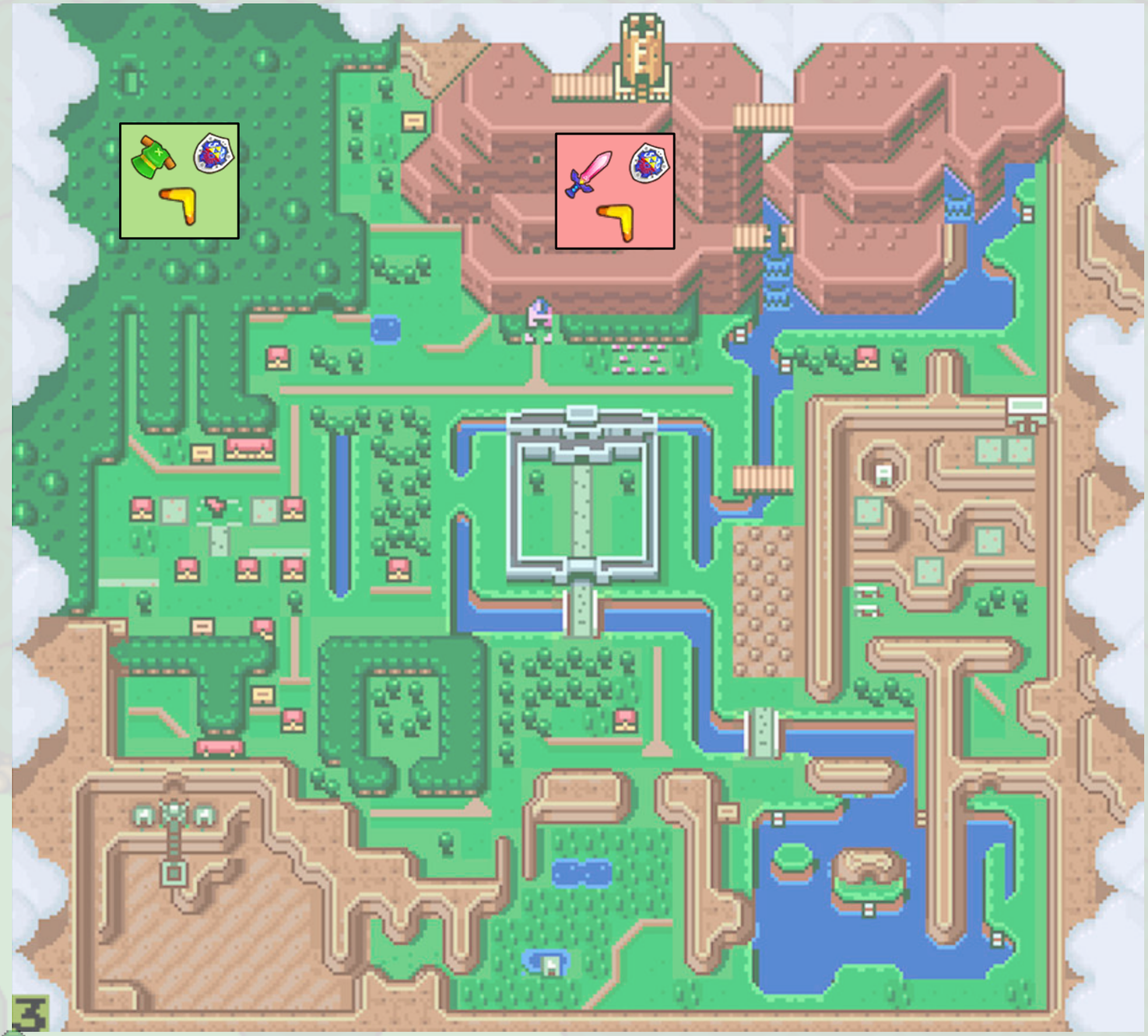
Can flip nodes with common neighbor

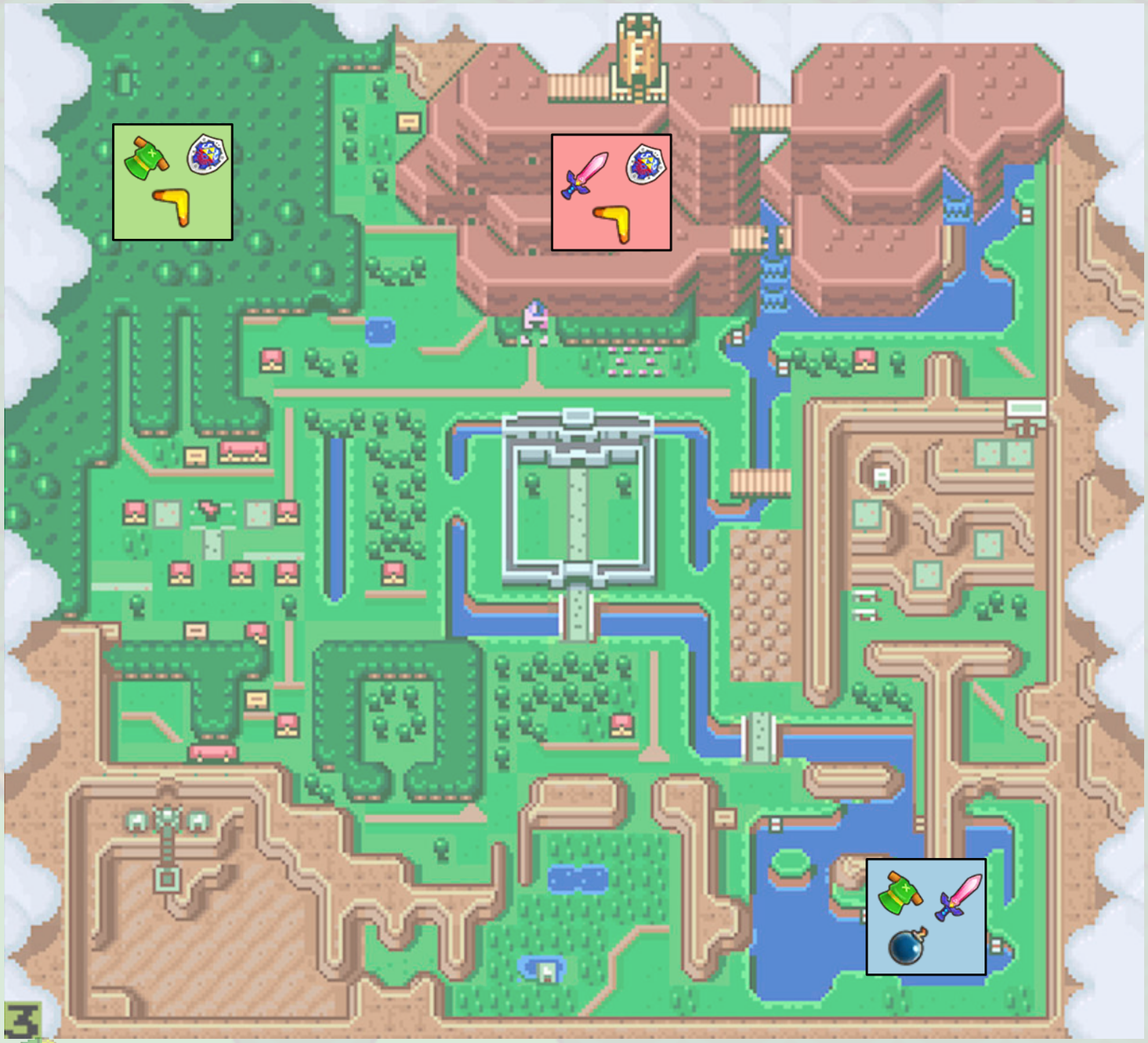
Goal: max. #successive pairs with same icon

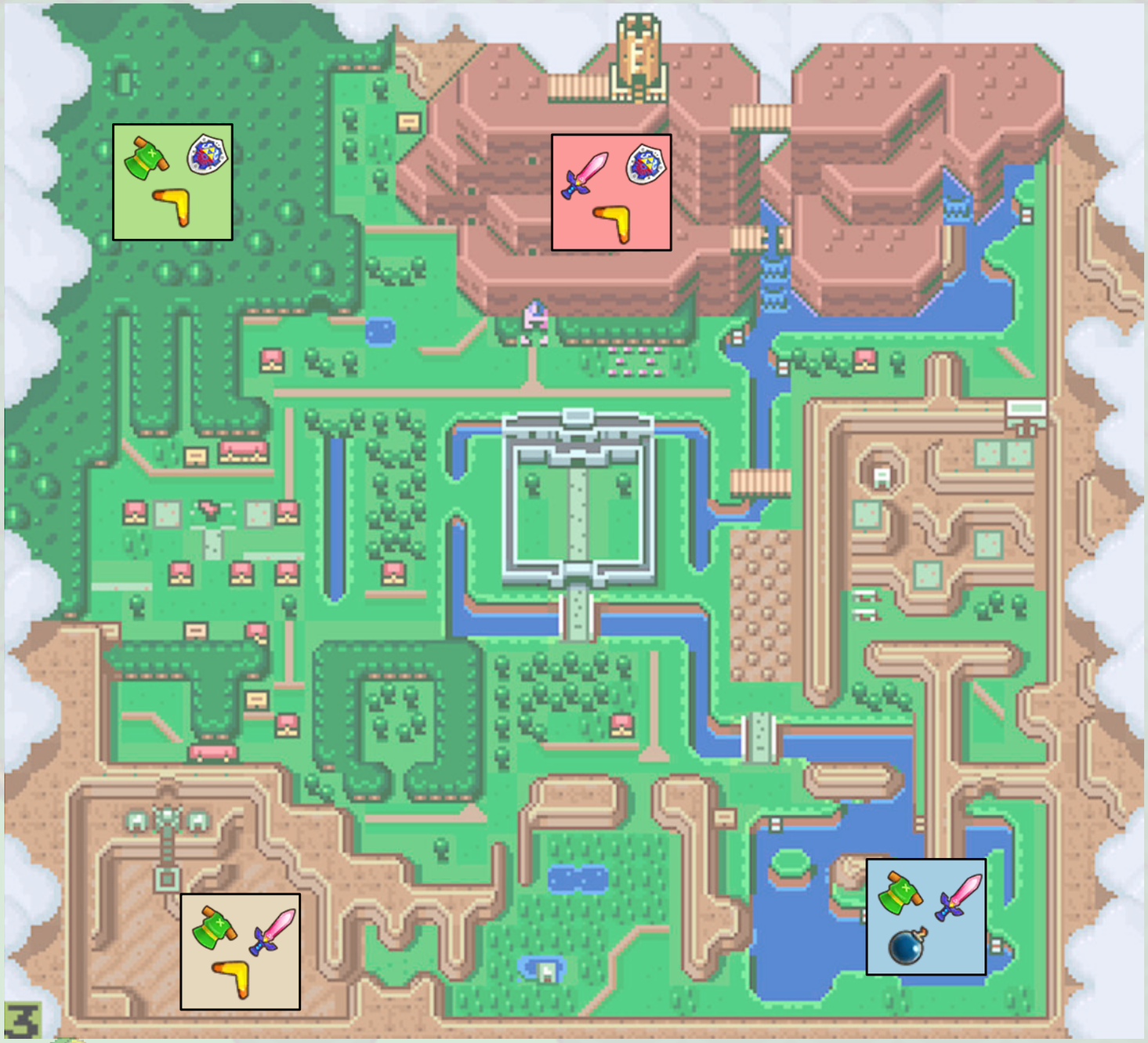


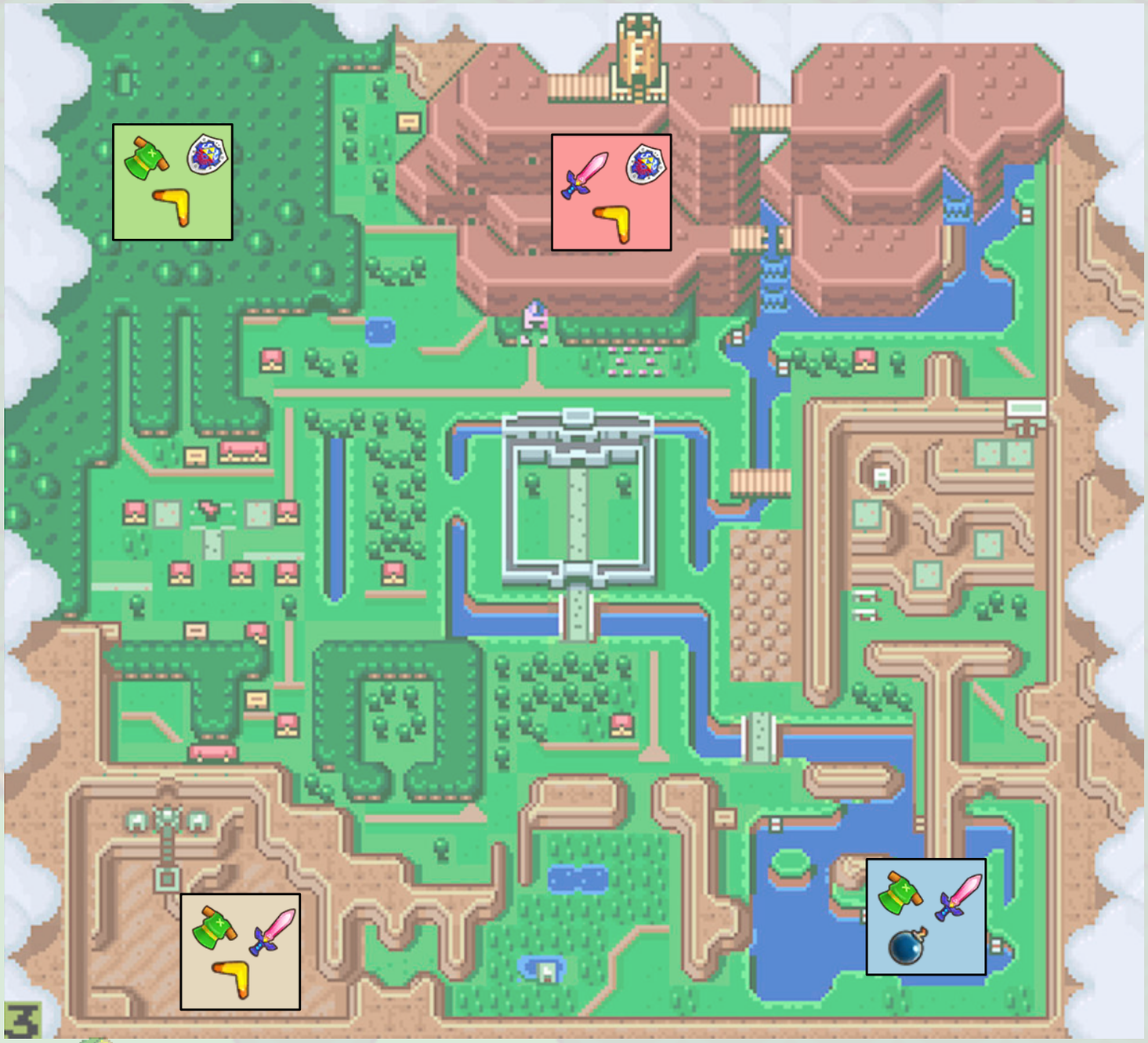




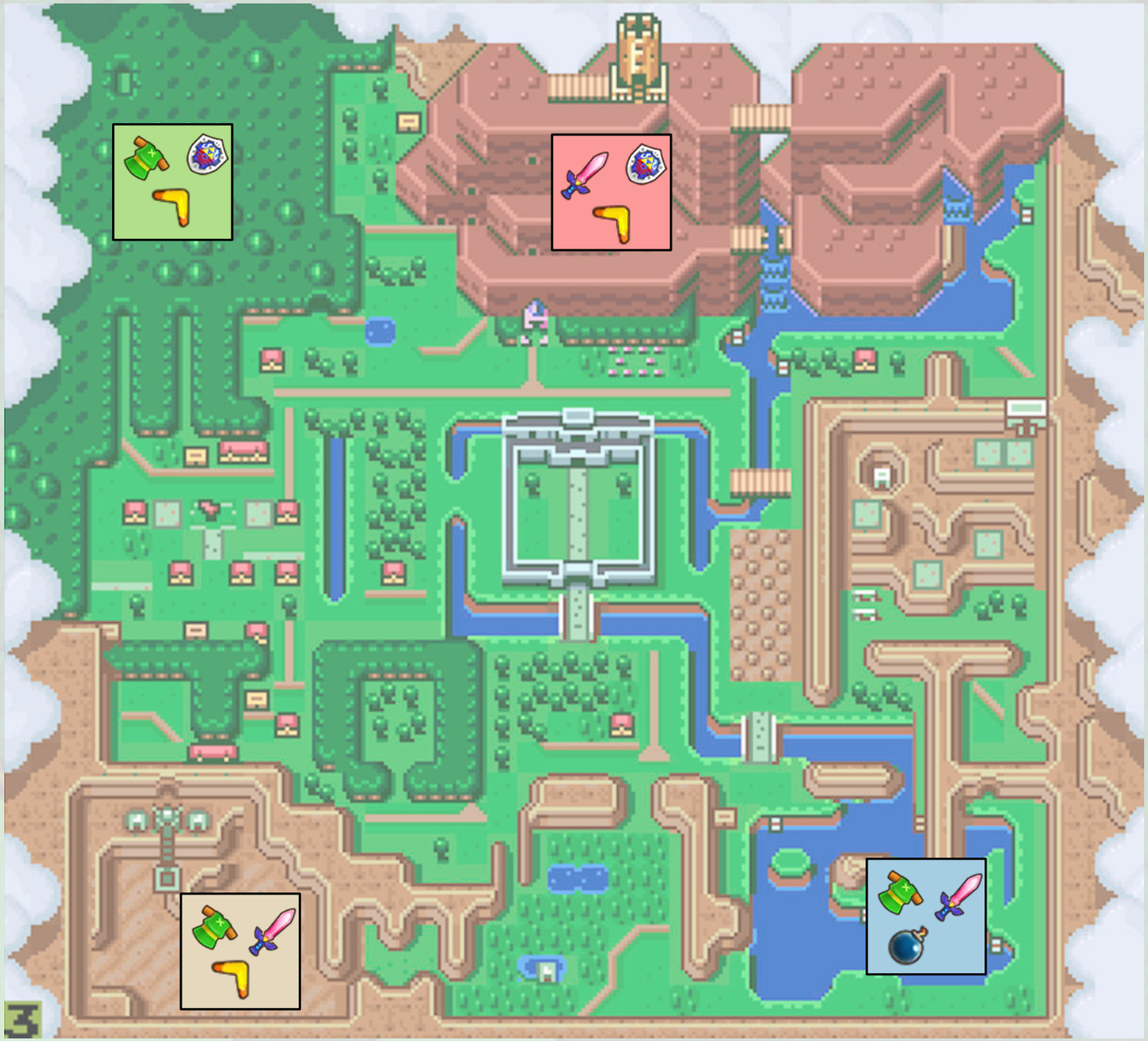




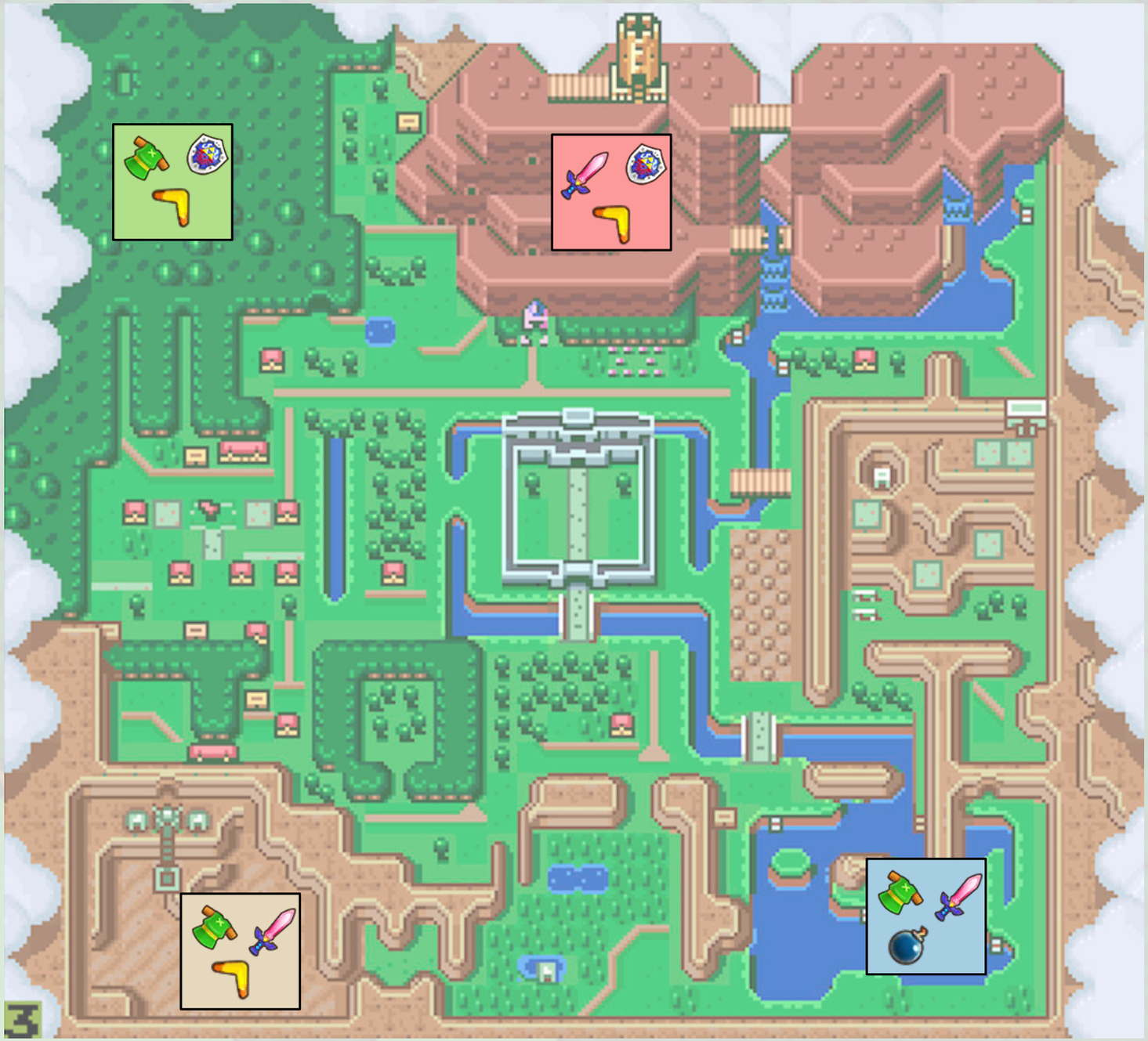




Where should I go to get all items with minimum effort?

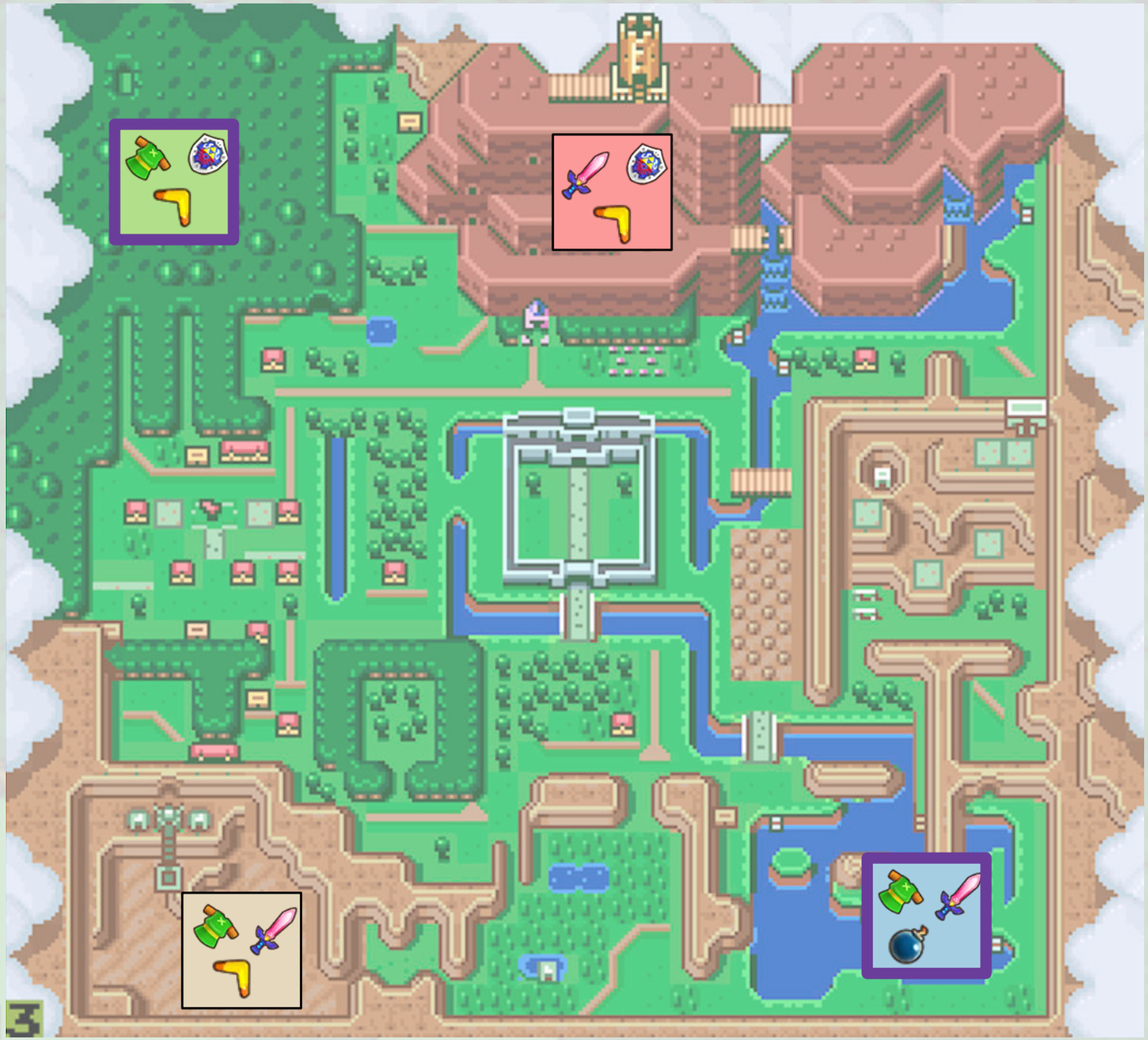


$n = 5, k = 2$



$n = 5, k = 2$

Is there a size-k SetCover?



$n = 5, k = 2$

Is there a size-k SetCover?

$n = 5, k = 2$

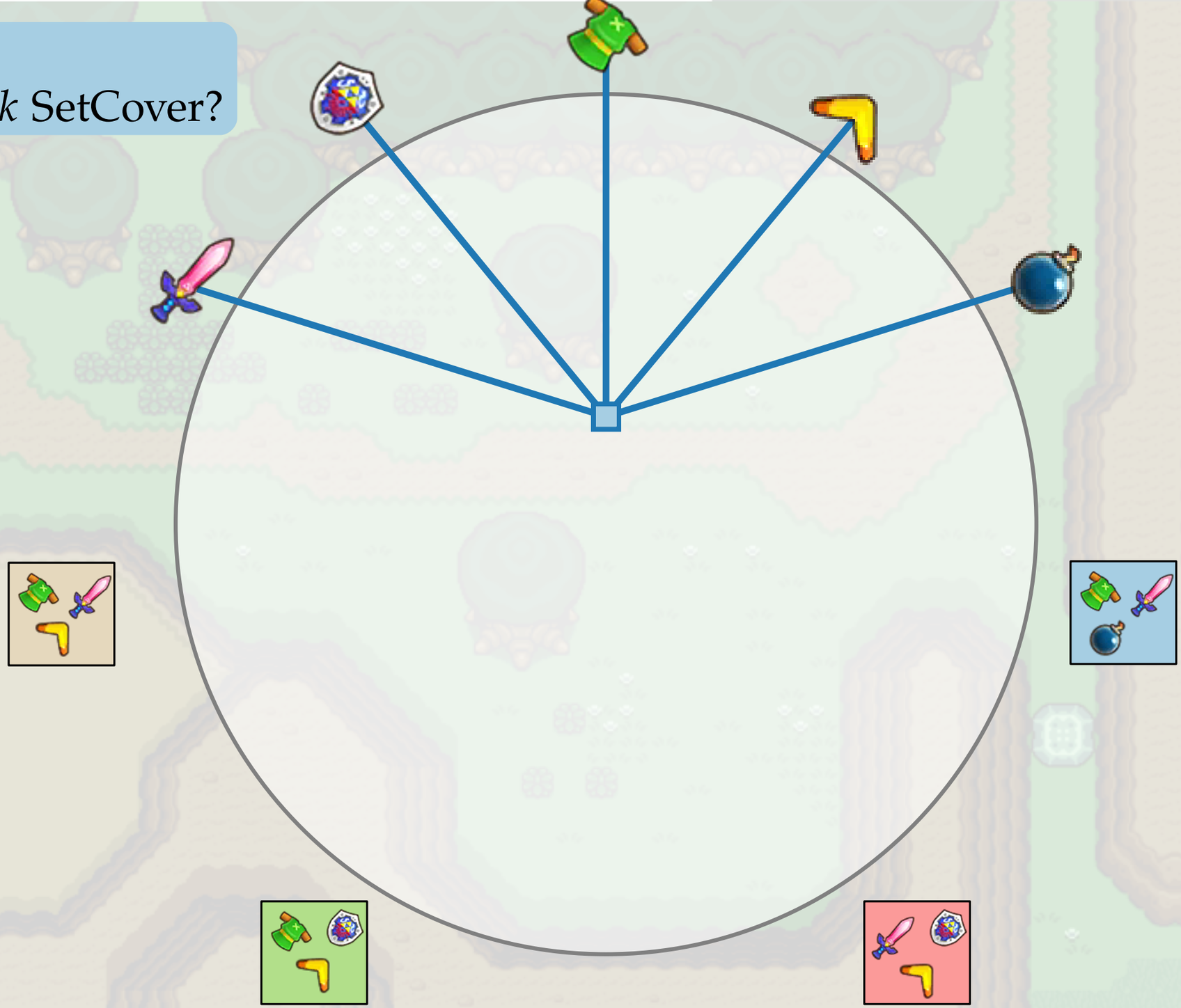
Is there a size-k SetCover?

$n = 5, k = 2$

Is there a size-k SetCover?

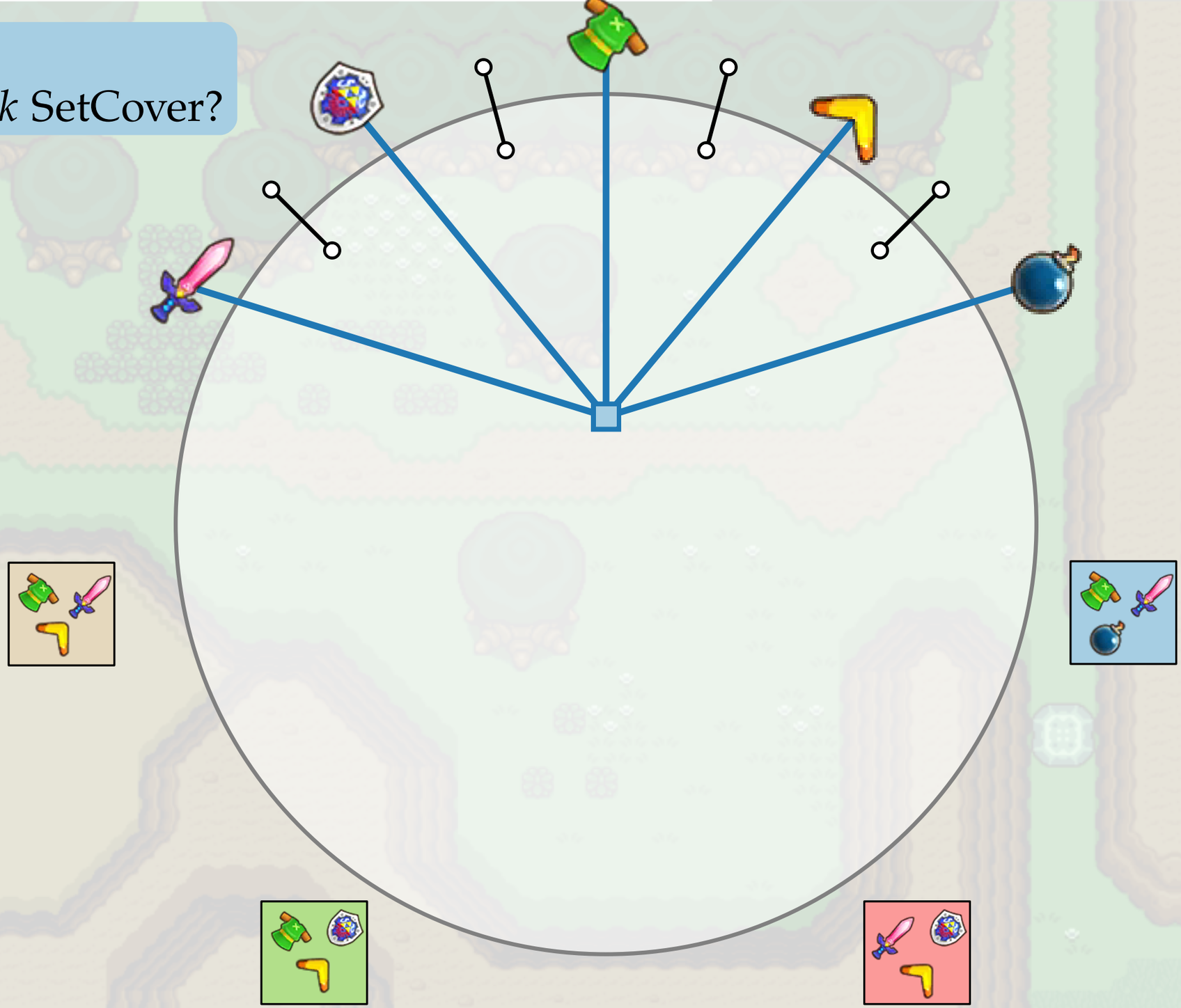
$n = 5, k = 2$

Is there a size-k SetCover?



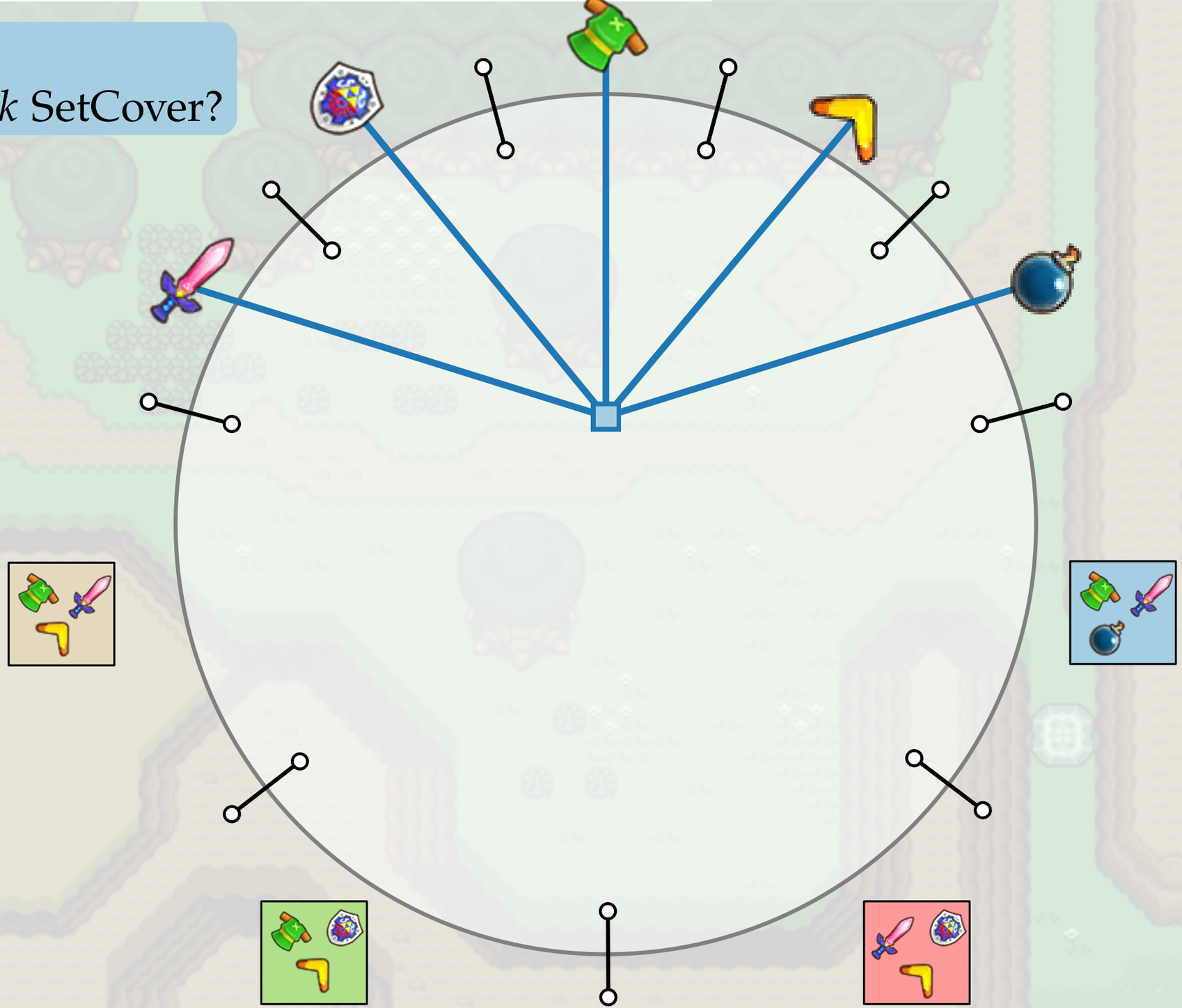
$n = 5, k = 2$

Is there a size-k SetCover?



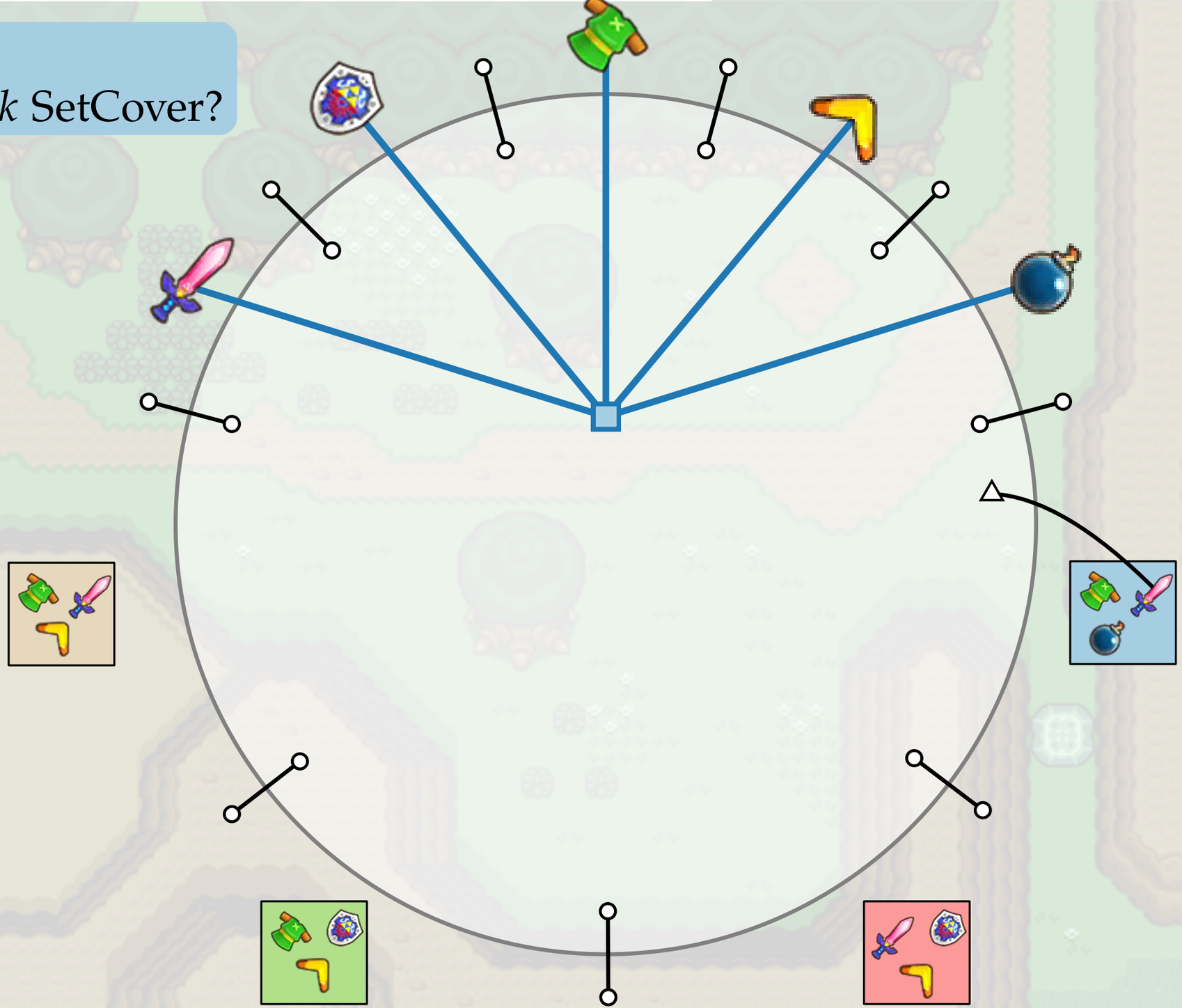
$n = 5, k = 2$

Is there a size-k SetCover?



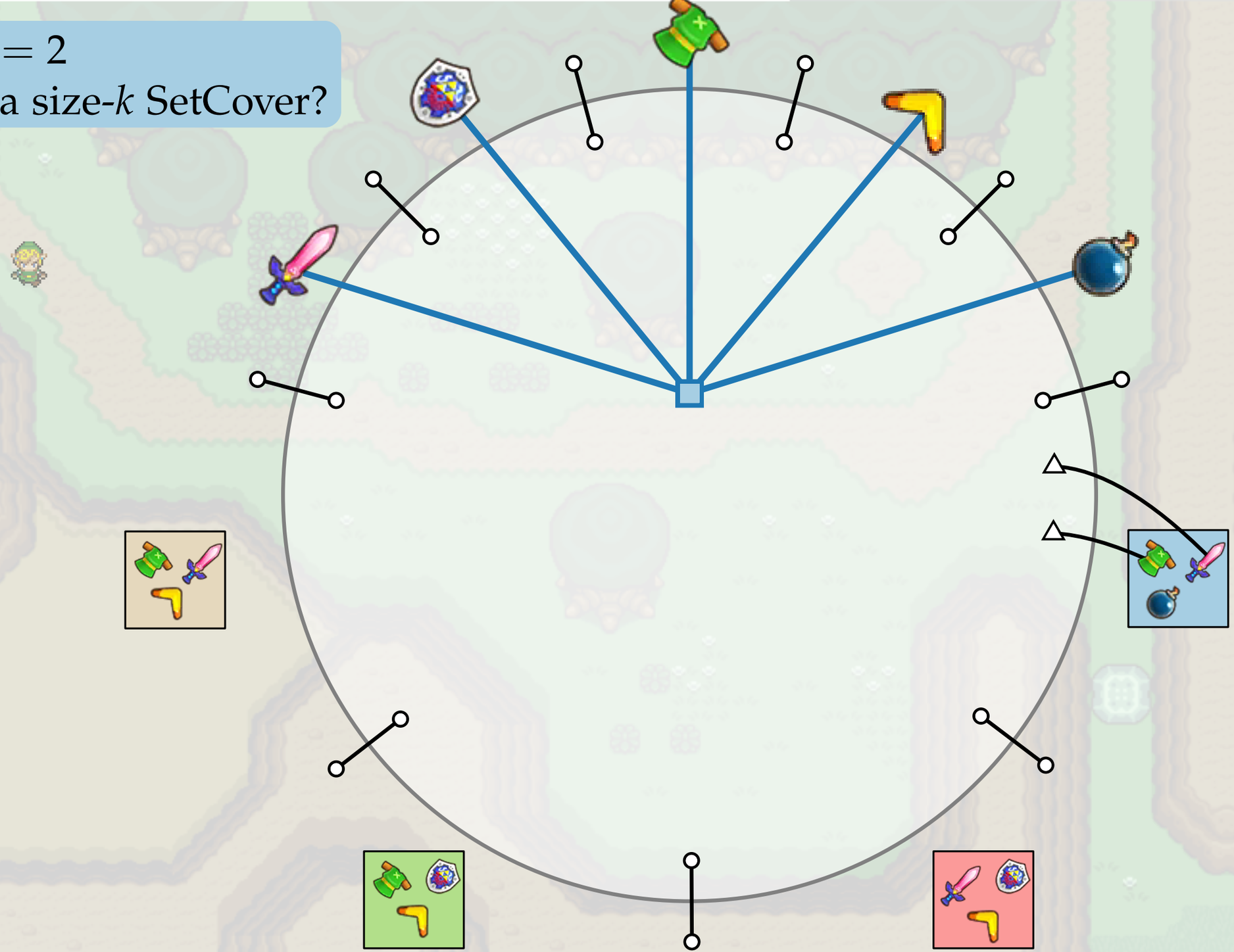
$n = 5, k = 2$

Is there a size-k SetCover?



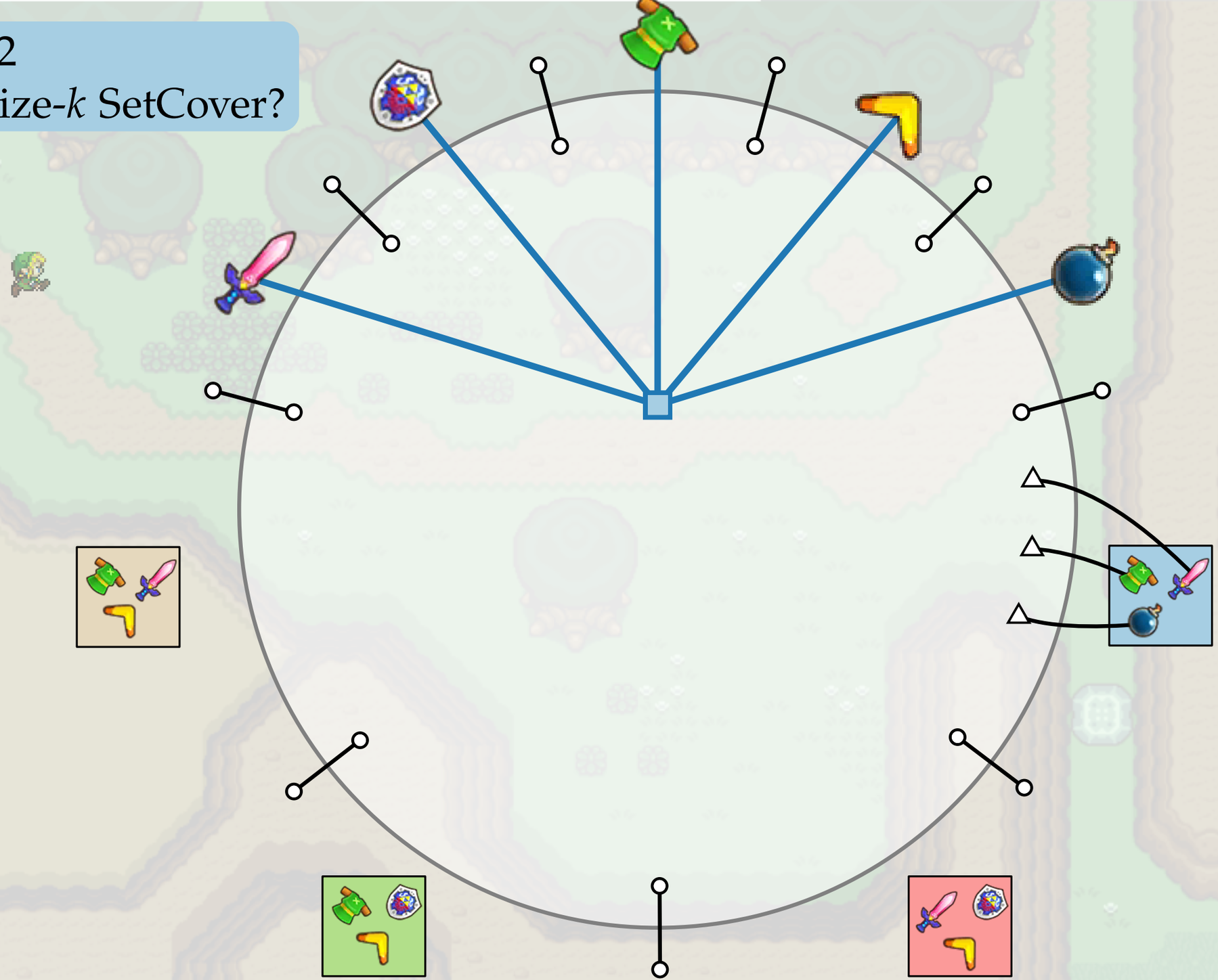
$n = 5, k = 2$

Is there a size-k SetCover?



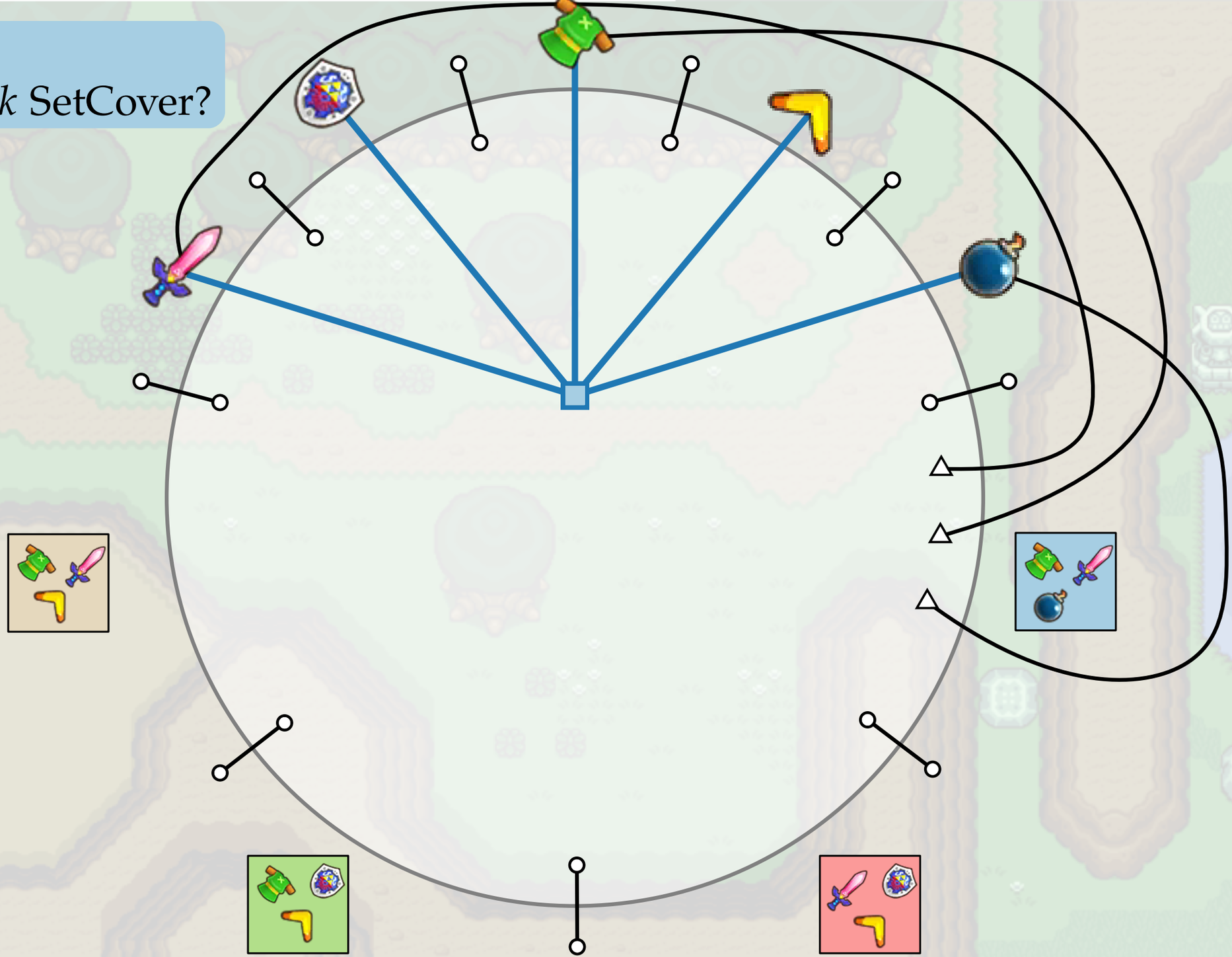
$n = 5, k = 2$

Is there a size-k SetCover?



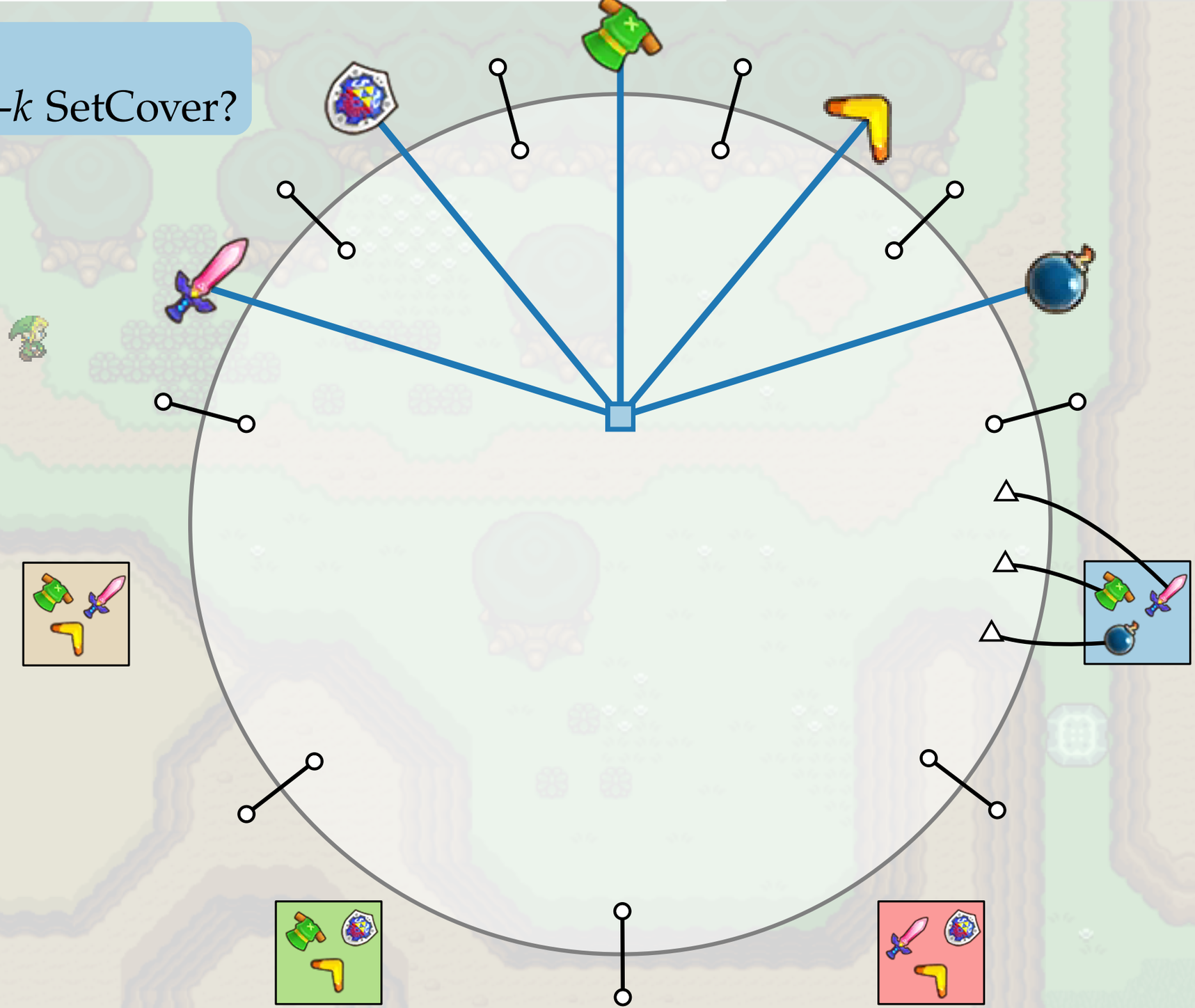
$n = 5, k = 2$

Is there a size-k SetCover?



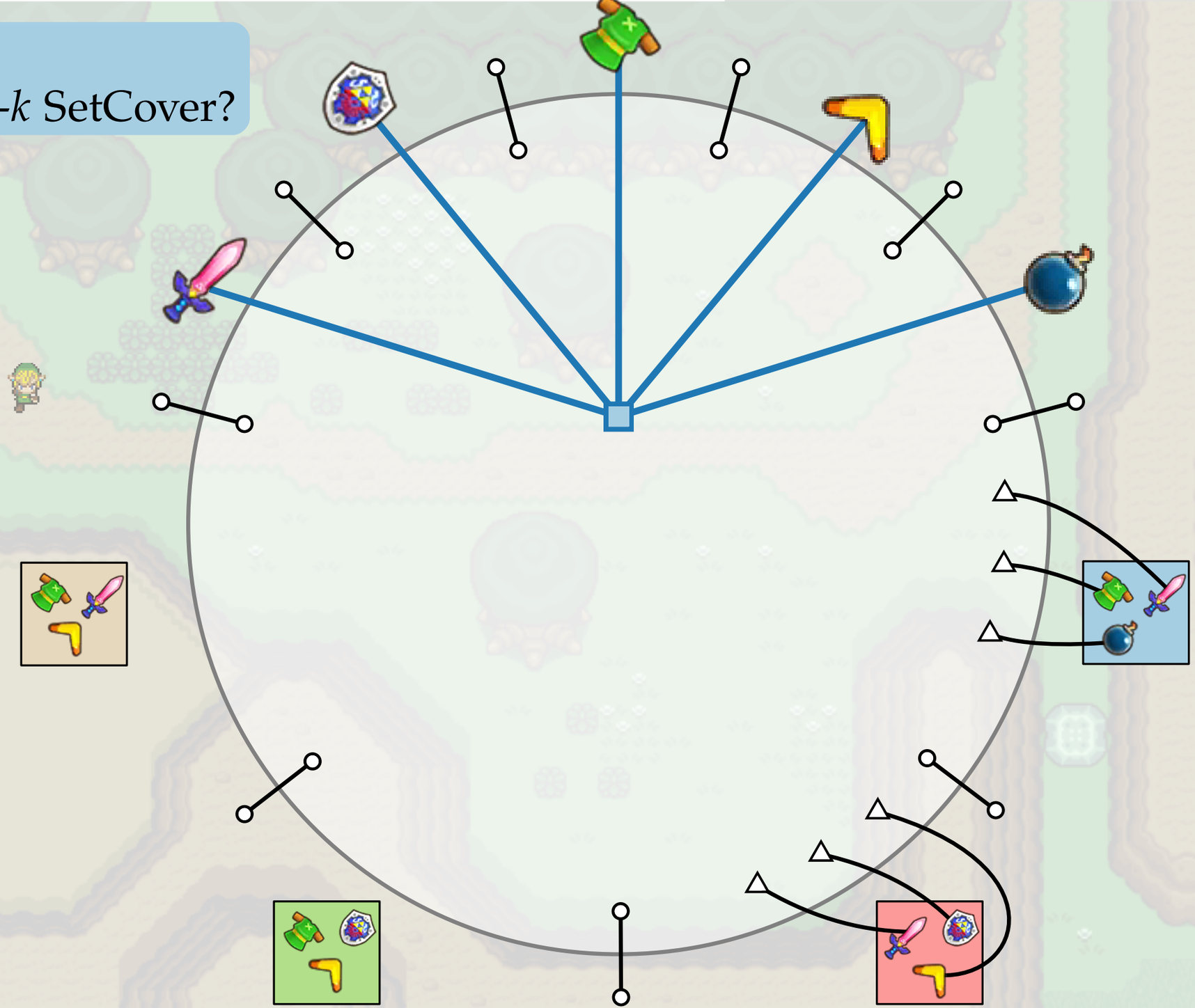
$n = 5, k = 2$

Is there a size-k SetCover?



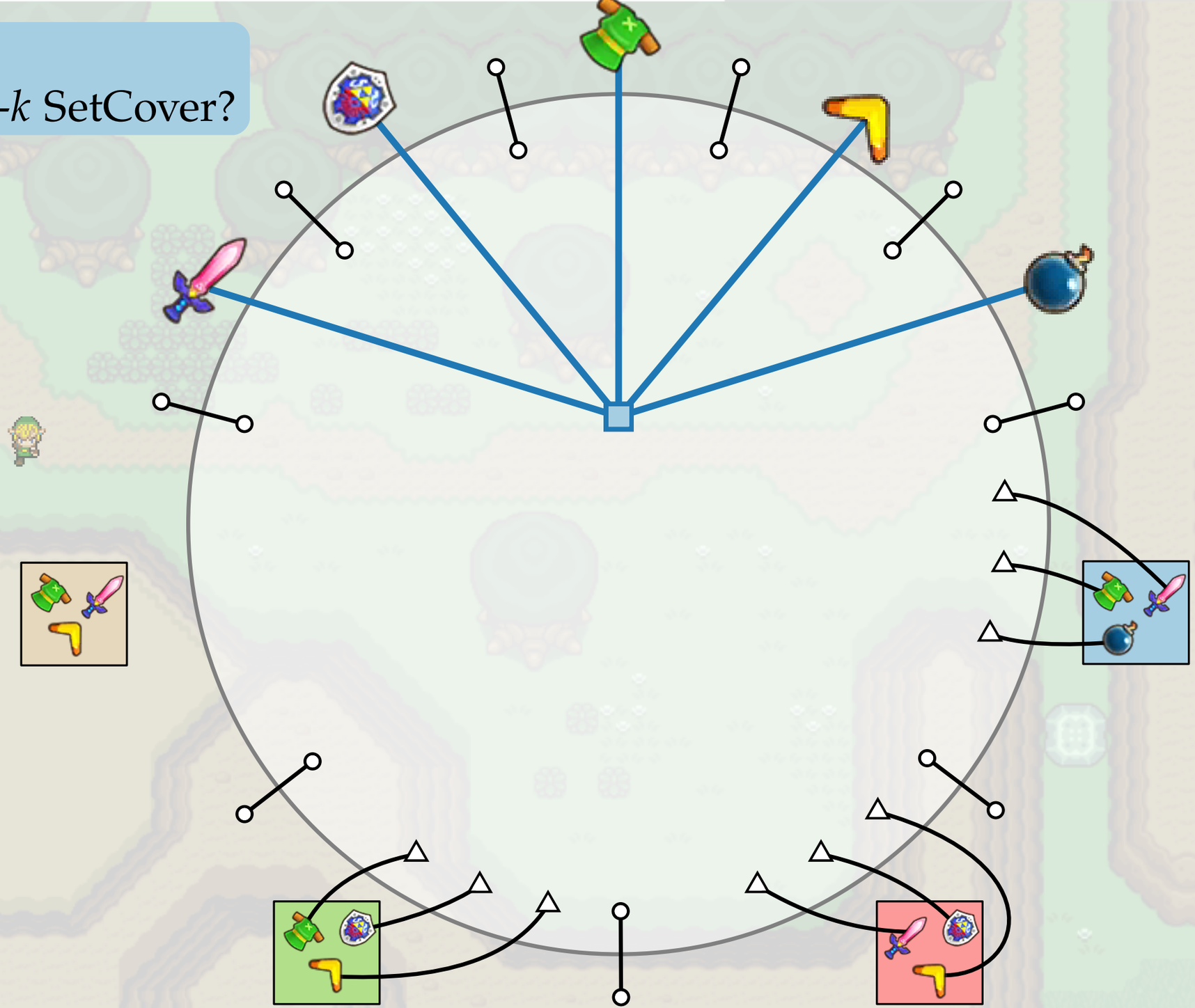
$n = 5, k = 2$

Is there a size-k SetCover?



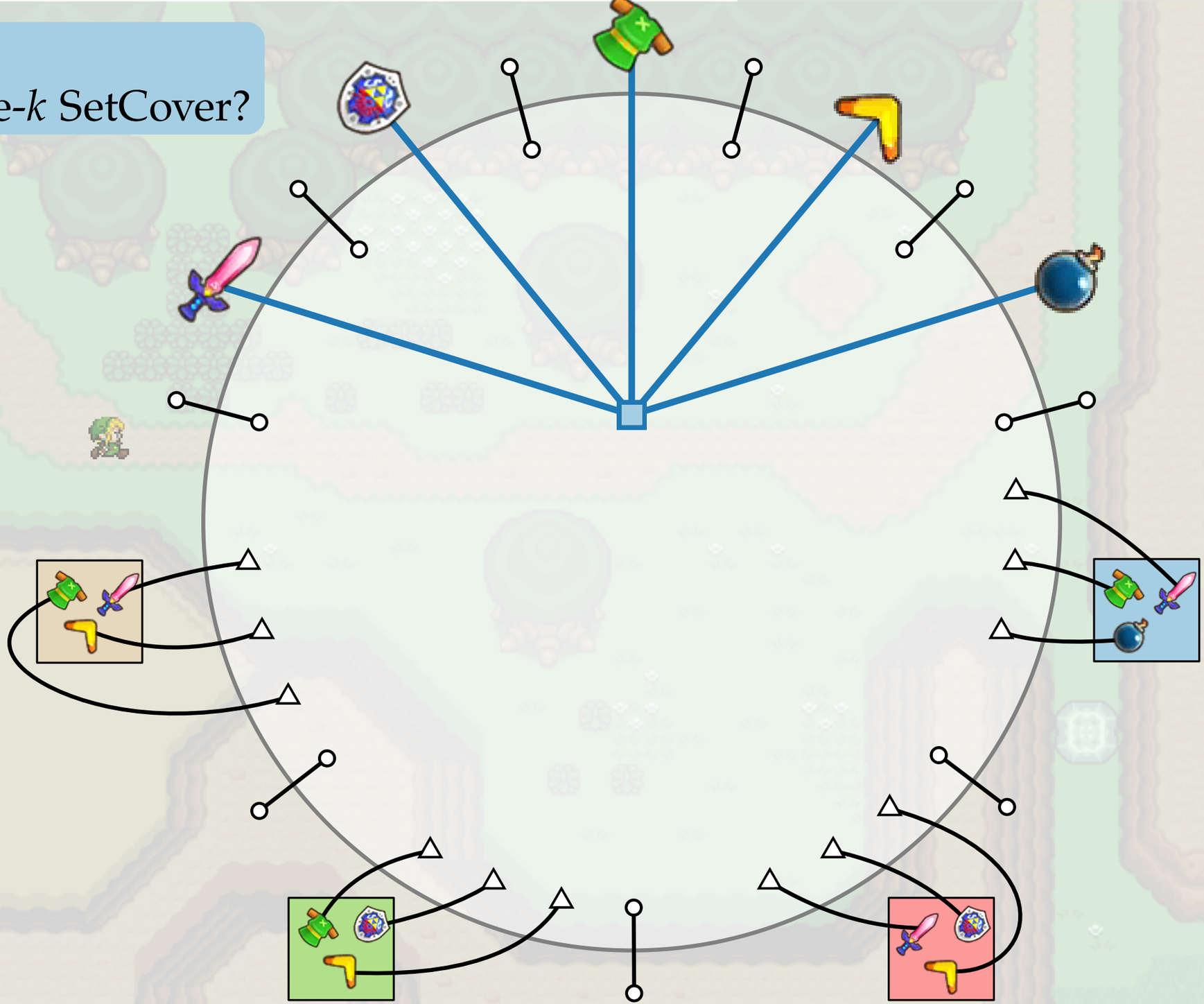
$n = 5, k = 2$

Is there a size-k SetCover?



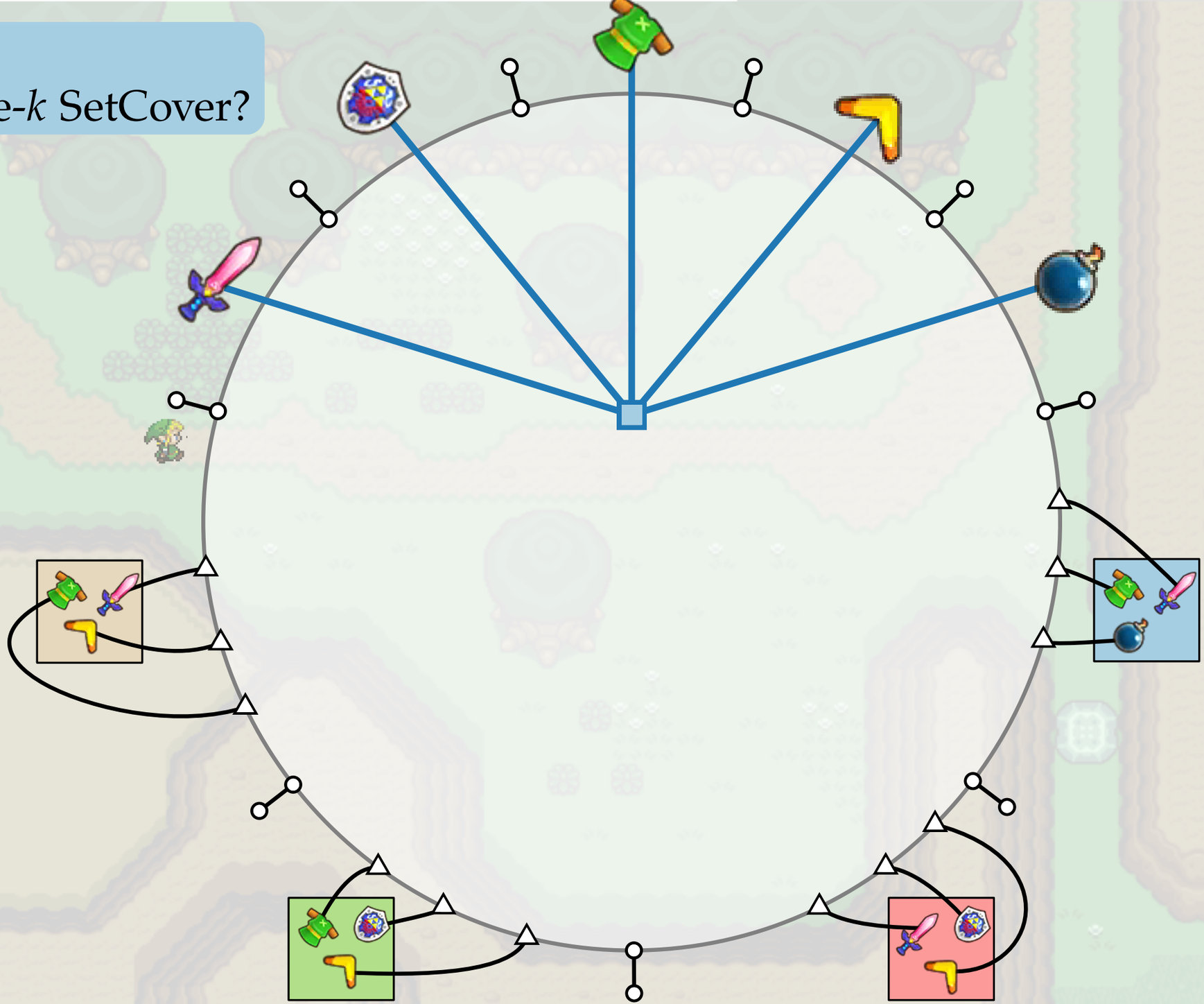
$n = 5, k = 2$

Is there a size-k SetCover?



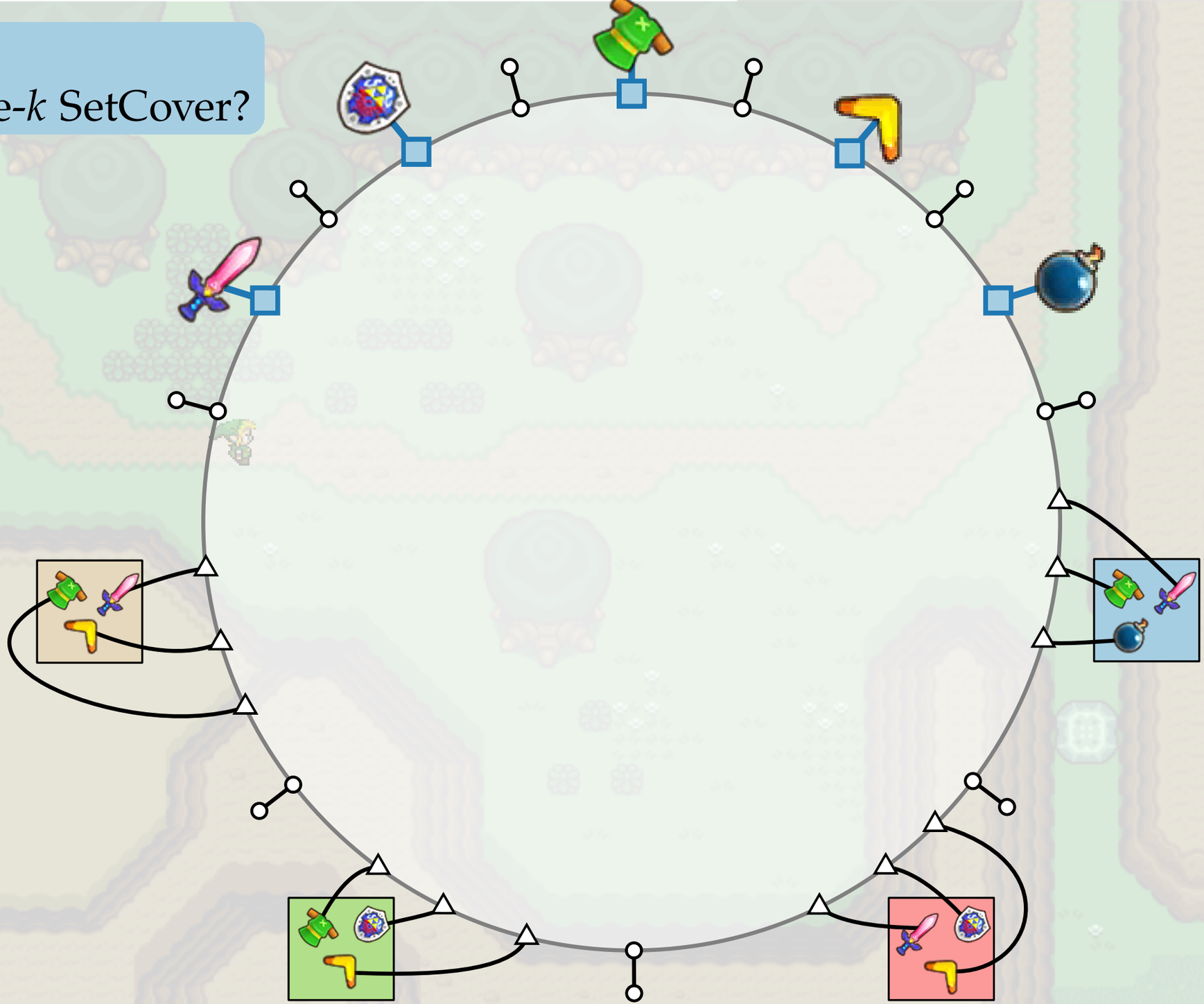
$n = 5, k = 2$

Is there a size-k SetCover?

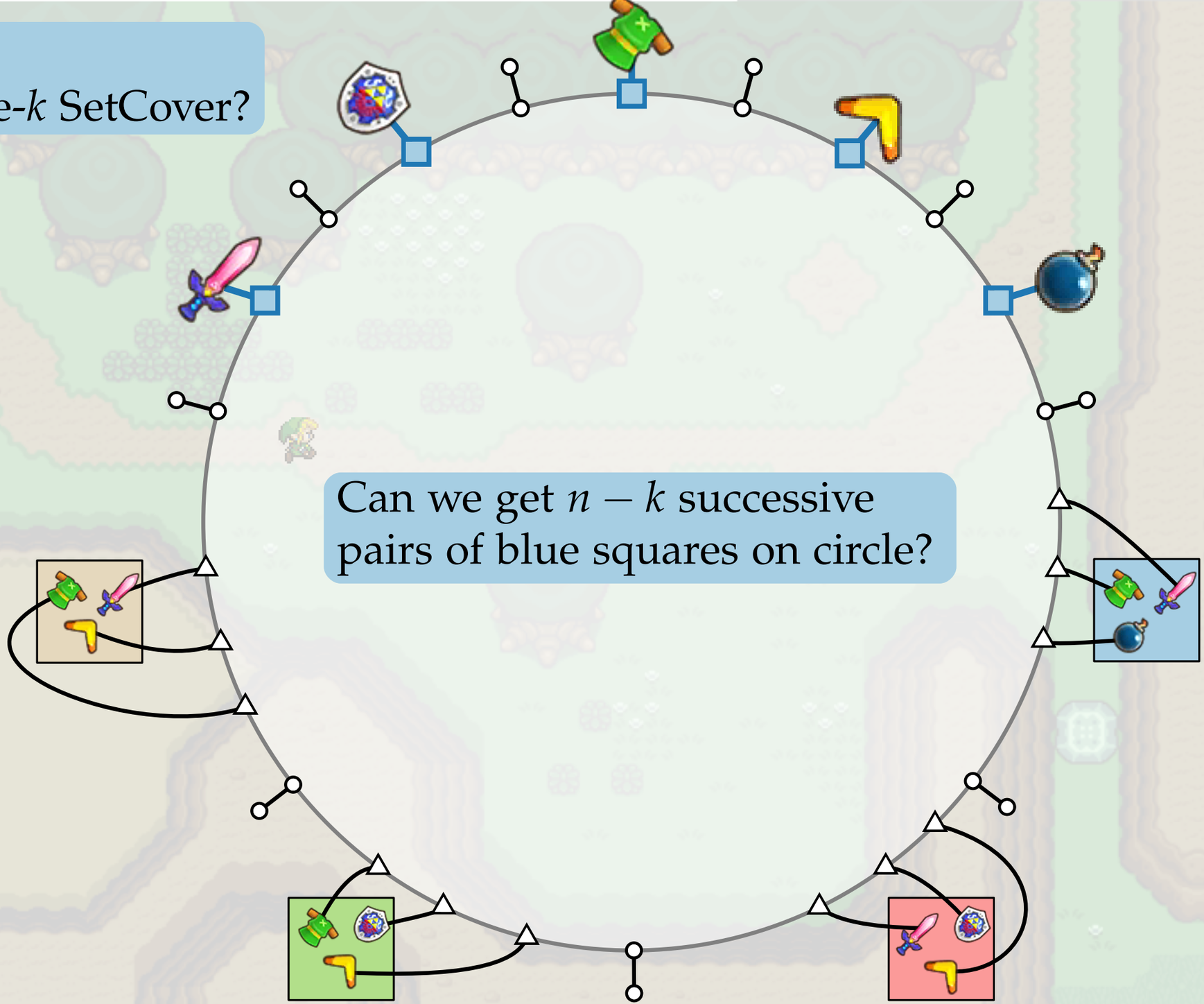


$n = 5, k = 2$

Is there a size-k SetCover?

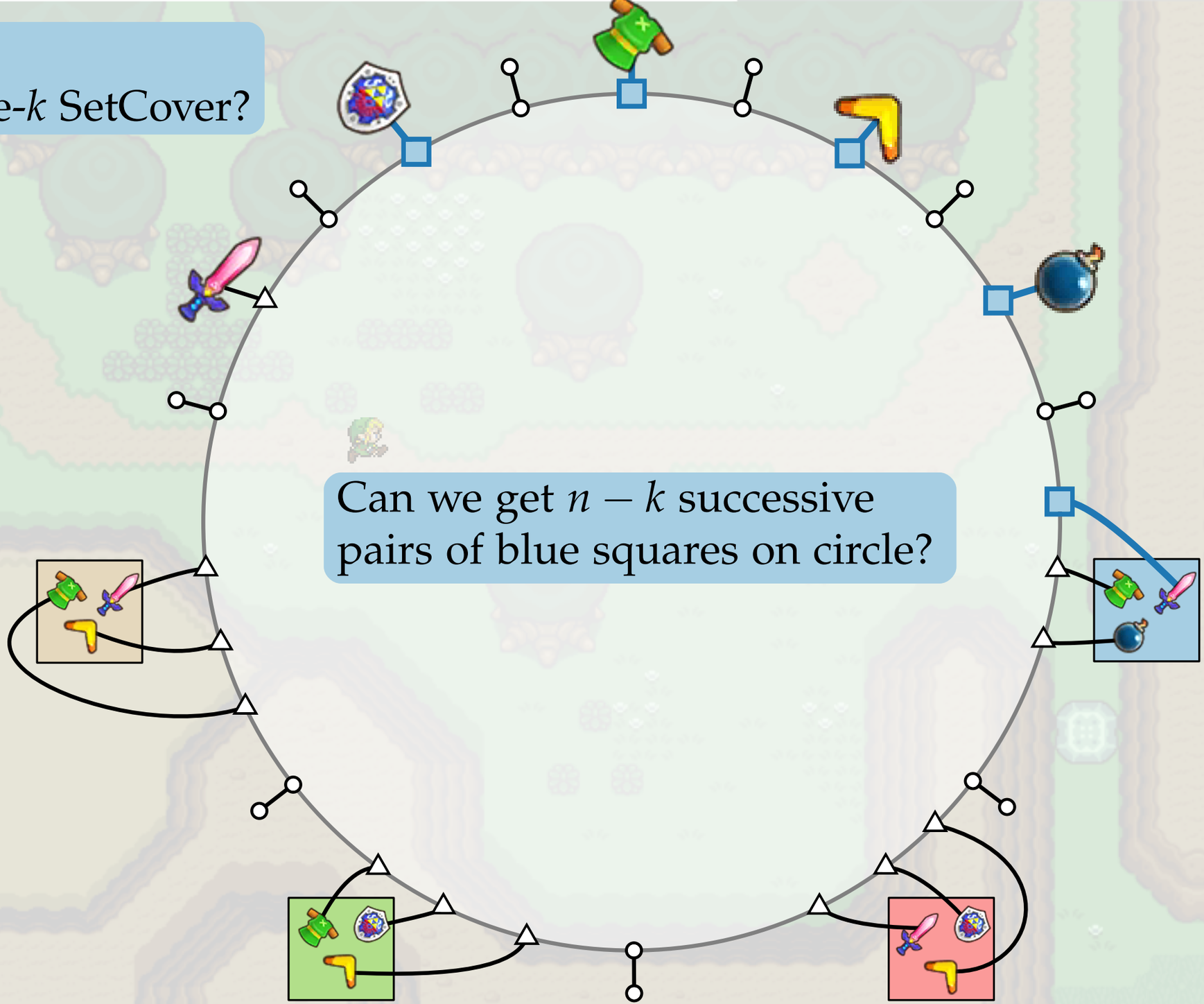


$n = 5, k = 2$
Is there a size-k SetCover?



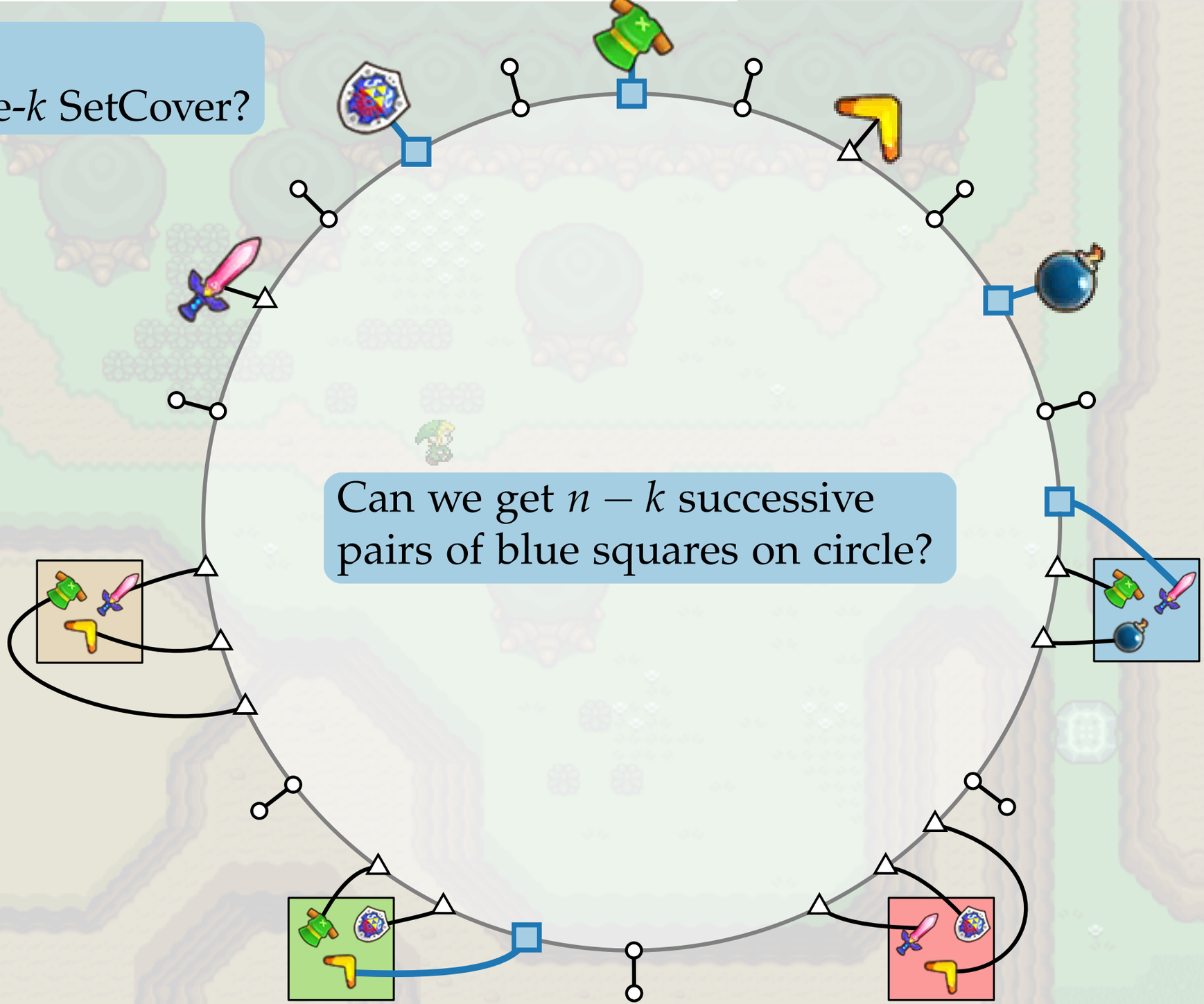
Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?



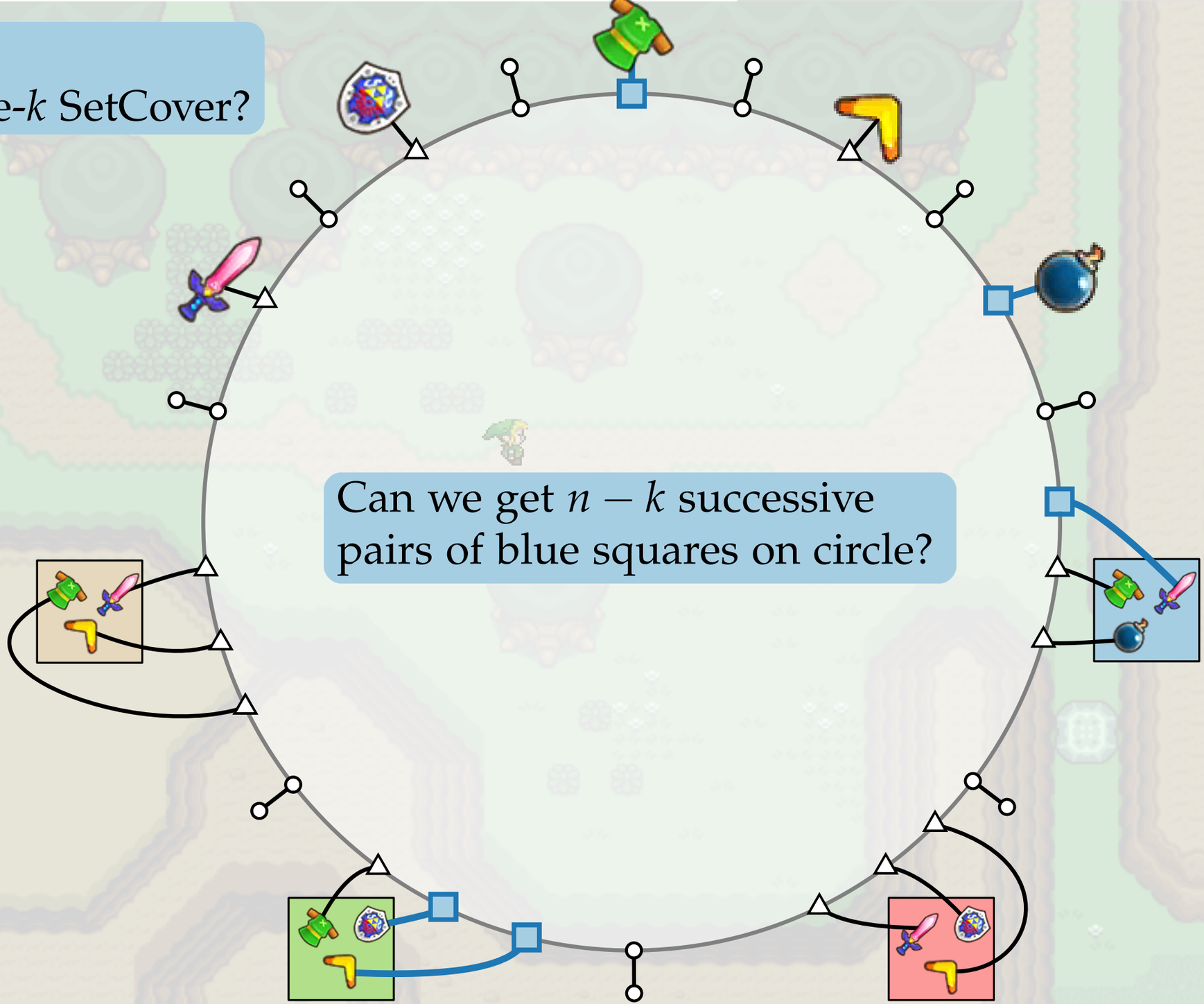
Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?



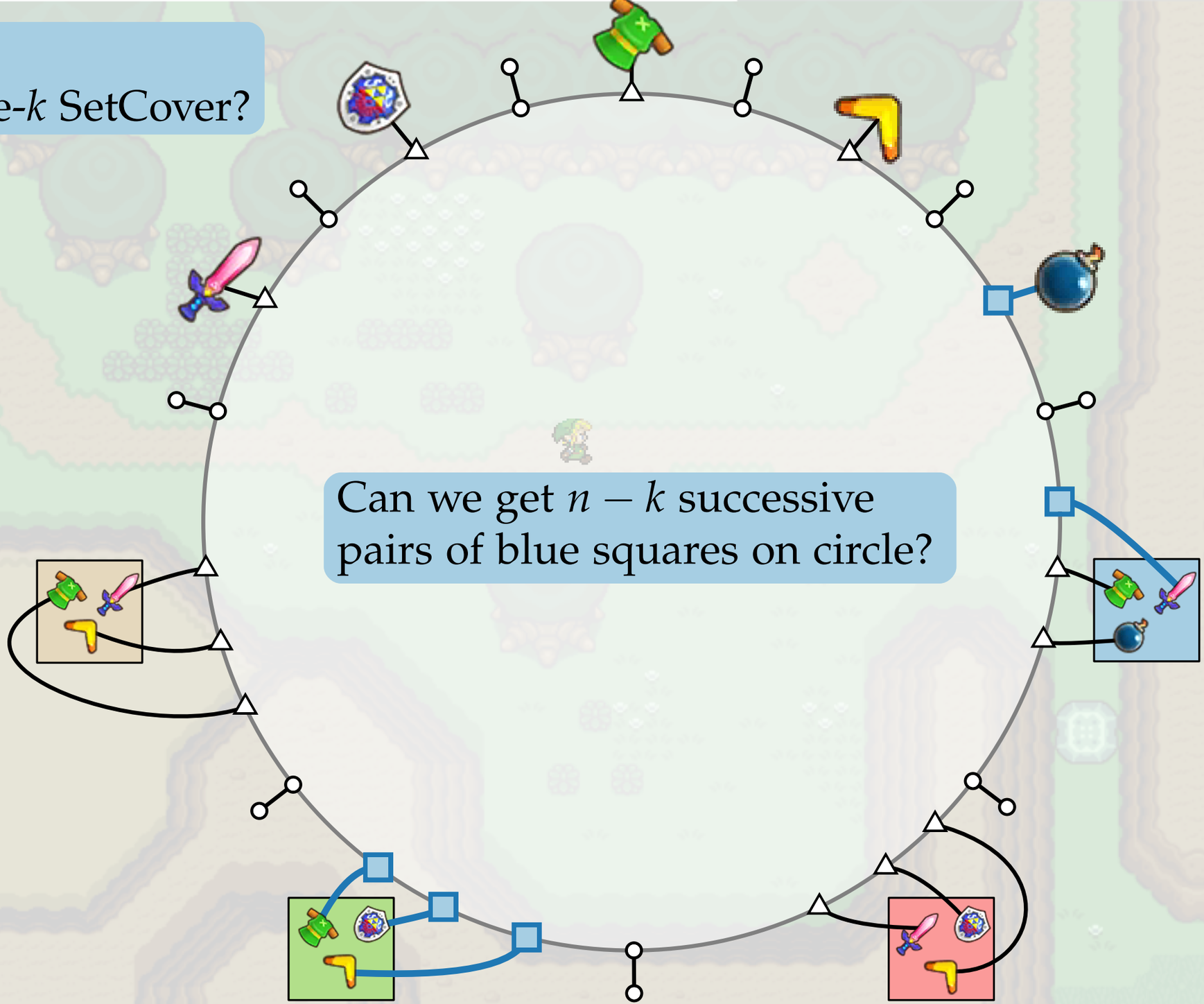
Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?



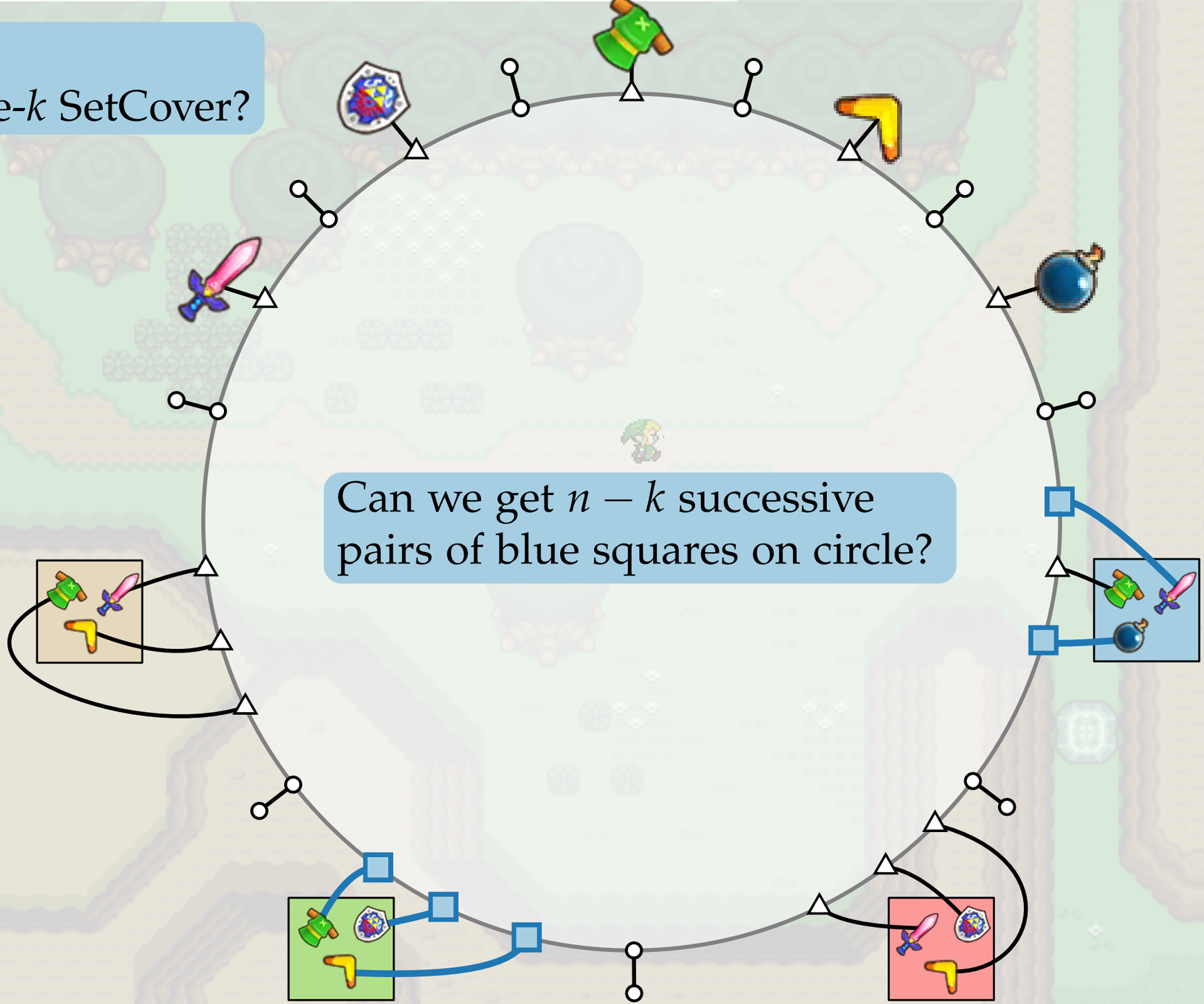
Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?



Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?



Can we get $n - k$ successive pairs of blue squares on circle?

$n = 5, k = 2$
Is there a size-k SetCover?

Can we get $n - k$ successive pairs of blue squares on circle?

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

Can we get $n - k$ successive pairs of blue squares on circle?

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

$n = 5, k = 2$
Is there a size-k SetCover?

$n = 5, k = 2$
Is there a size-k SetCover?

$n = 5, k = 2$
Is there a size-k SetCover?

$n = 5, k = 2$
Is there a size-k SetCover?

Can we get $n - k$ successive pairs of blue squares on circle?

2 successive blue pairs

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

Can we get $n - k$ successive pairs of blue squares on circle?

2 successive blue pairs

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

$n = 5, k = 2$
Is there a size-k SetCover?

Can we get $n - k$ successive pairs of blue squares on circle?

2 successive blue pairs

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

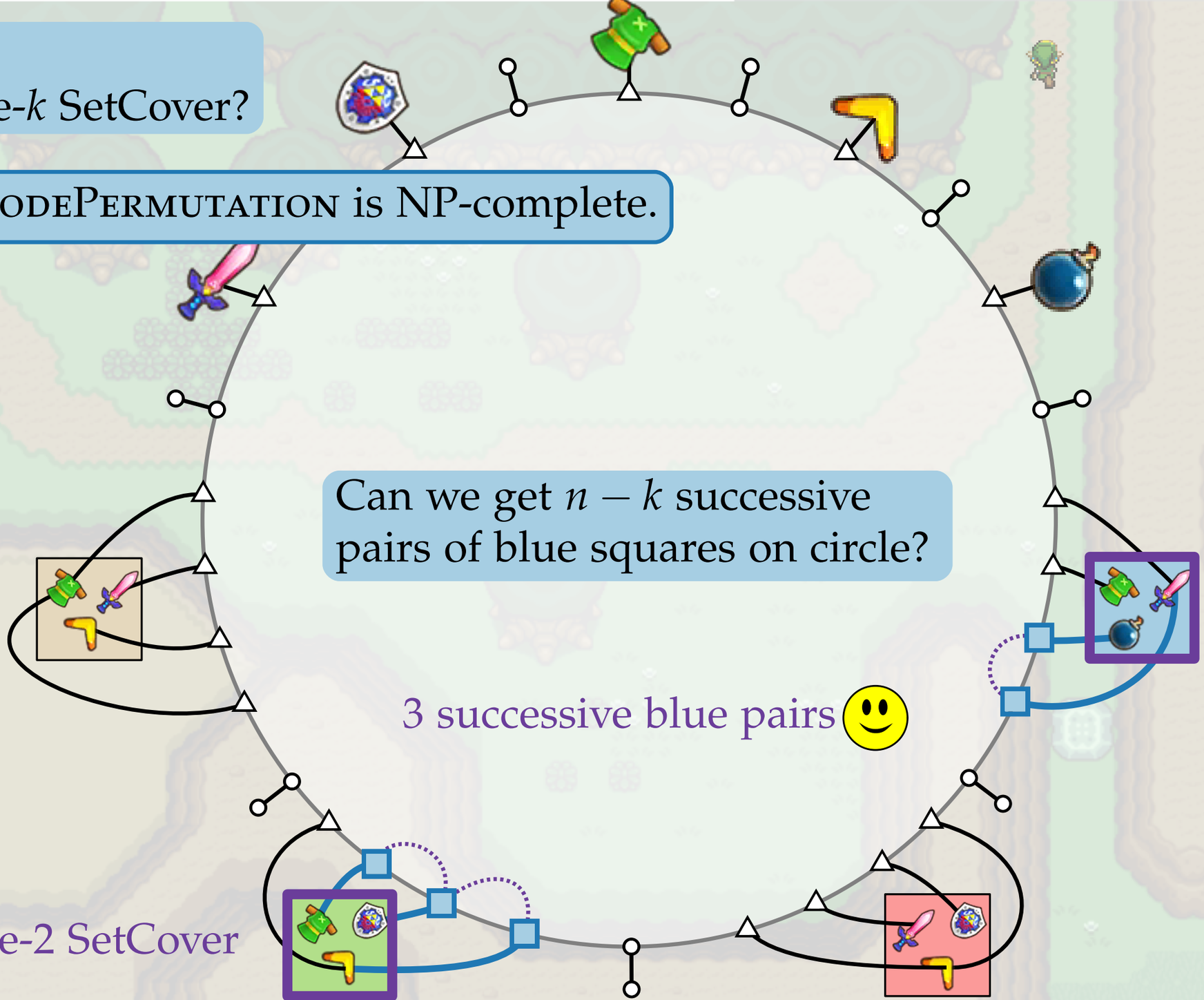
Can we get $n - k$ successive pairs of blue squares on circle?

3 successive blue pairs

size-2 SetCover

$n = 5, k = 2$
Is there a size-k SetCover?

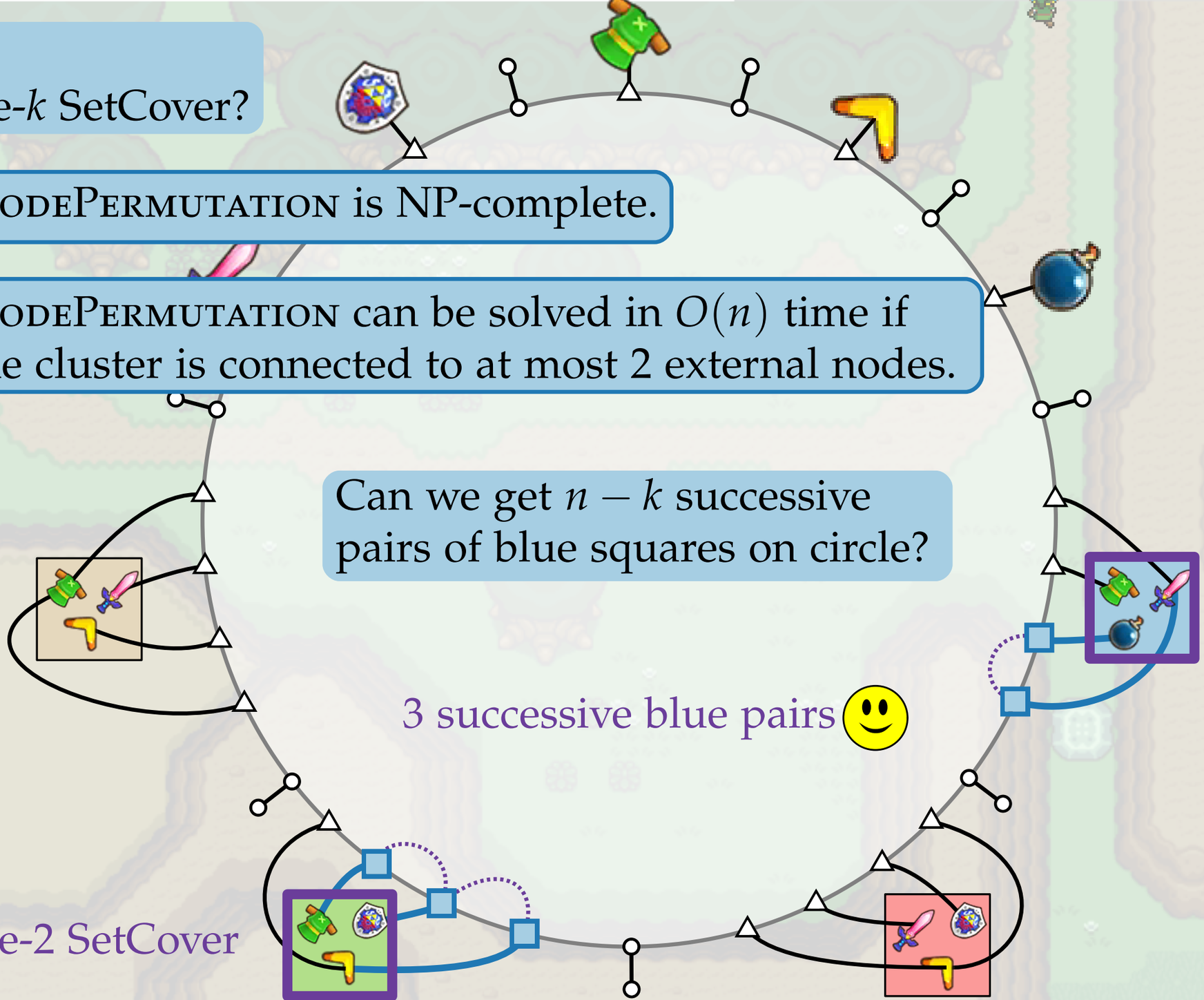
Theorem. NodePermutation is NP-complete.

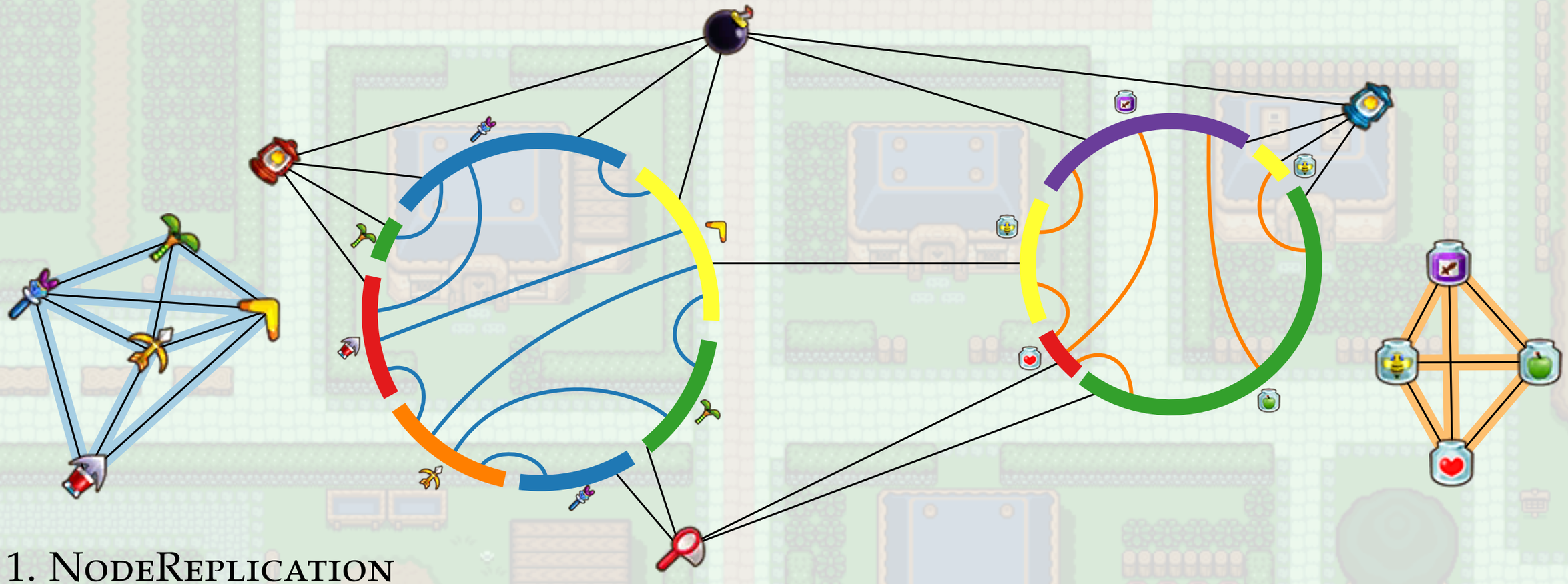


$n = 5, k = 2$
Is there a size-k SetCover?

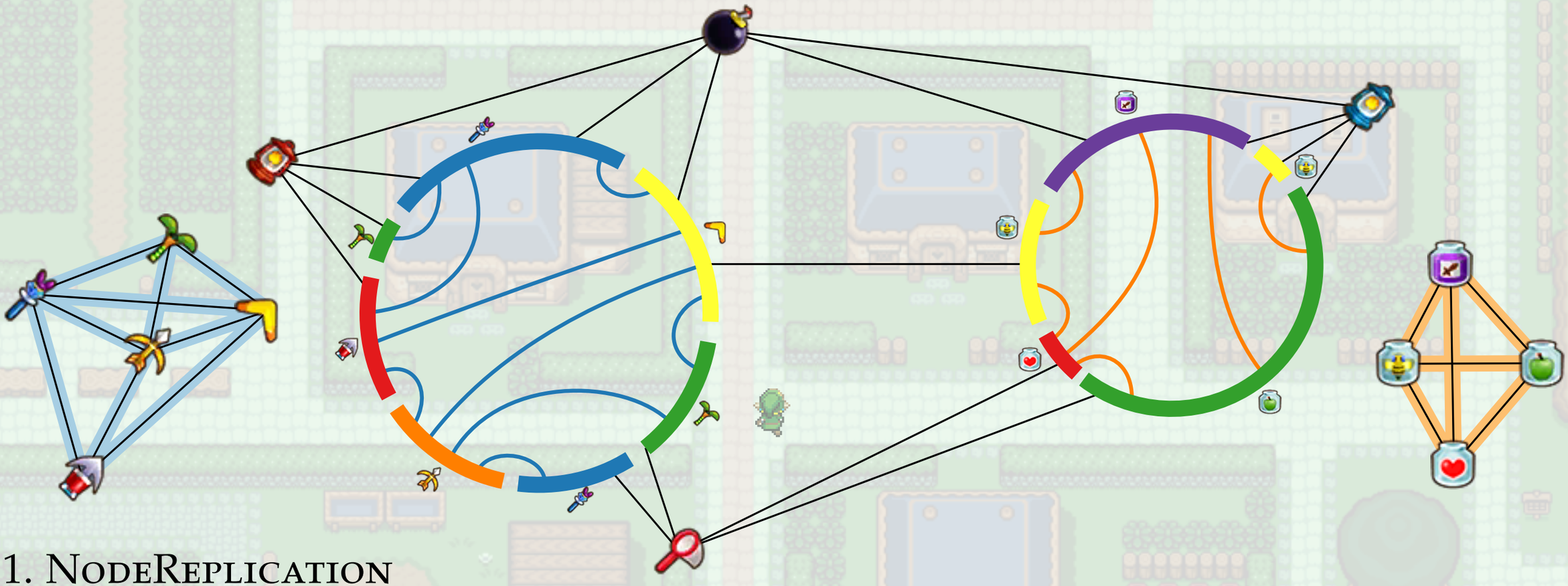
Theorem. NodePermutation is NP-complete.

Theorem. NodePermutation can be solved in $O(n)$ time if the cluster is connected to at most 2 external nodes.

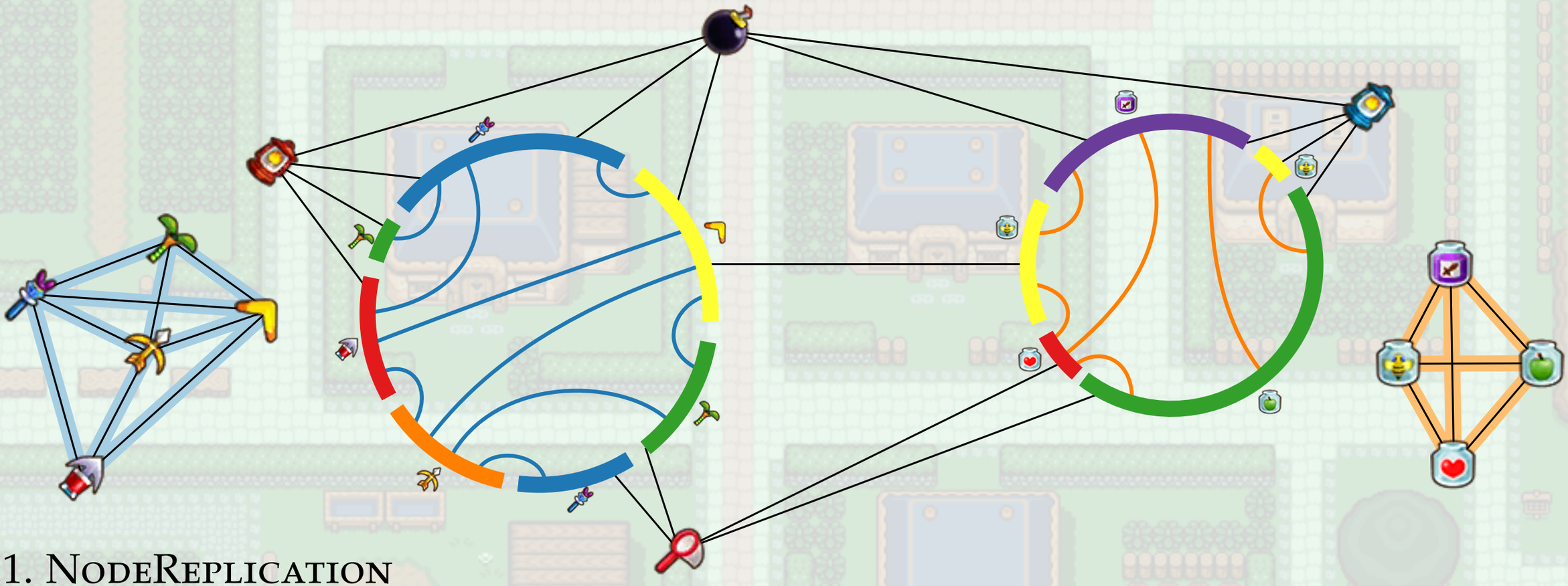




1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor
   Goal: max. #successive pairs with same icon
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster
   Choose any copies of the adjacent vertices
   Goal: min. #crossings



1. NodeReplication
2. NodePermutation — Can flip nodes with common neighbor; Goal: max. #successive pairs with same icon — NP-complete
3. NodeMerging
4. ChordInsertion — Reintroduce chords in cluster; Choose any copies of the adjacent vertices; Goal: min. #crossings

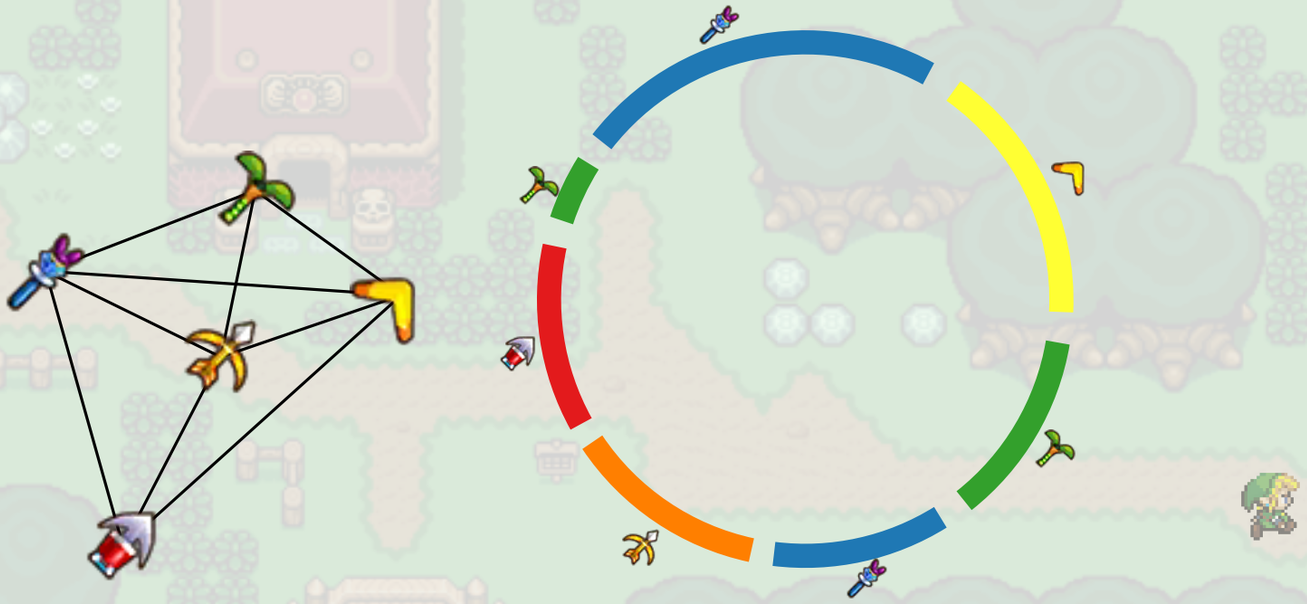


1. NodeReplication		
2. NodePermutation	Can flip nodes with common neighbor Goal: max. #successive pairs with same icon	NP-complete
3. NodeMerging		
4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings	

4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings

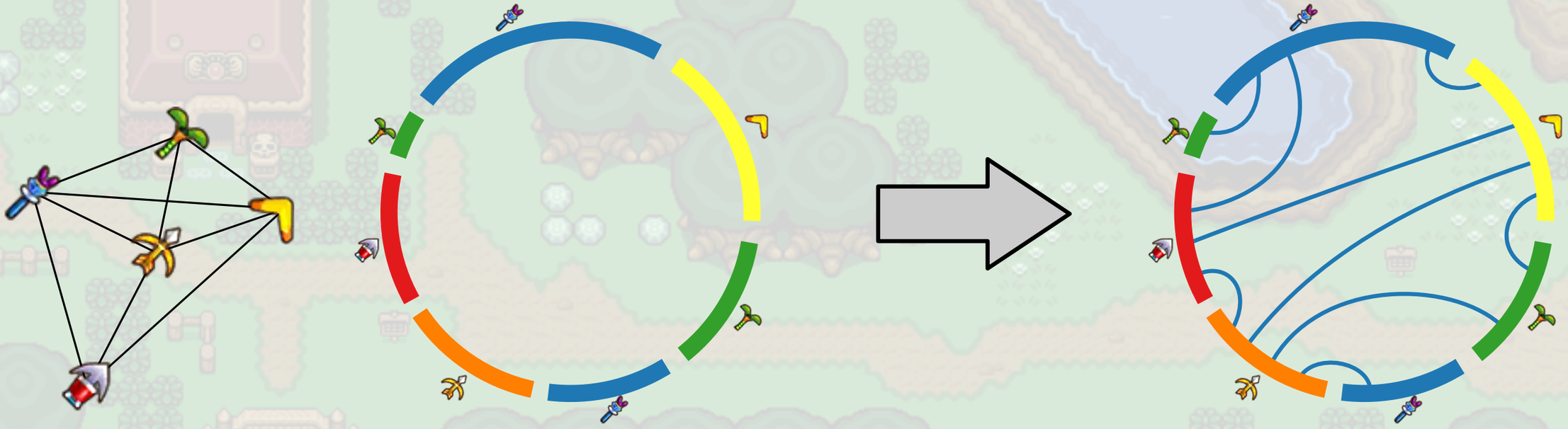
4. ChordInsertion

Reintroduce chords in cluster
Choose any copies of the adjacent vertices
Goal: min. #crossings



4. ChordInsertion

Reintroduce chords in cluster
Choose any copies of the adjacent vertices
Goal: min. #crossings



or

or

or

or

or

or

or

or

or

or

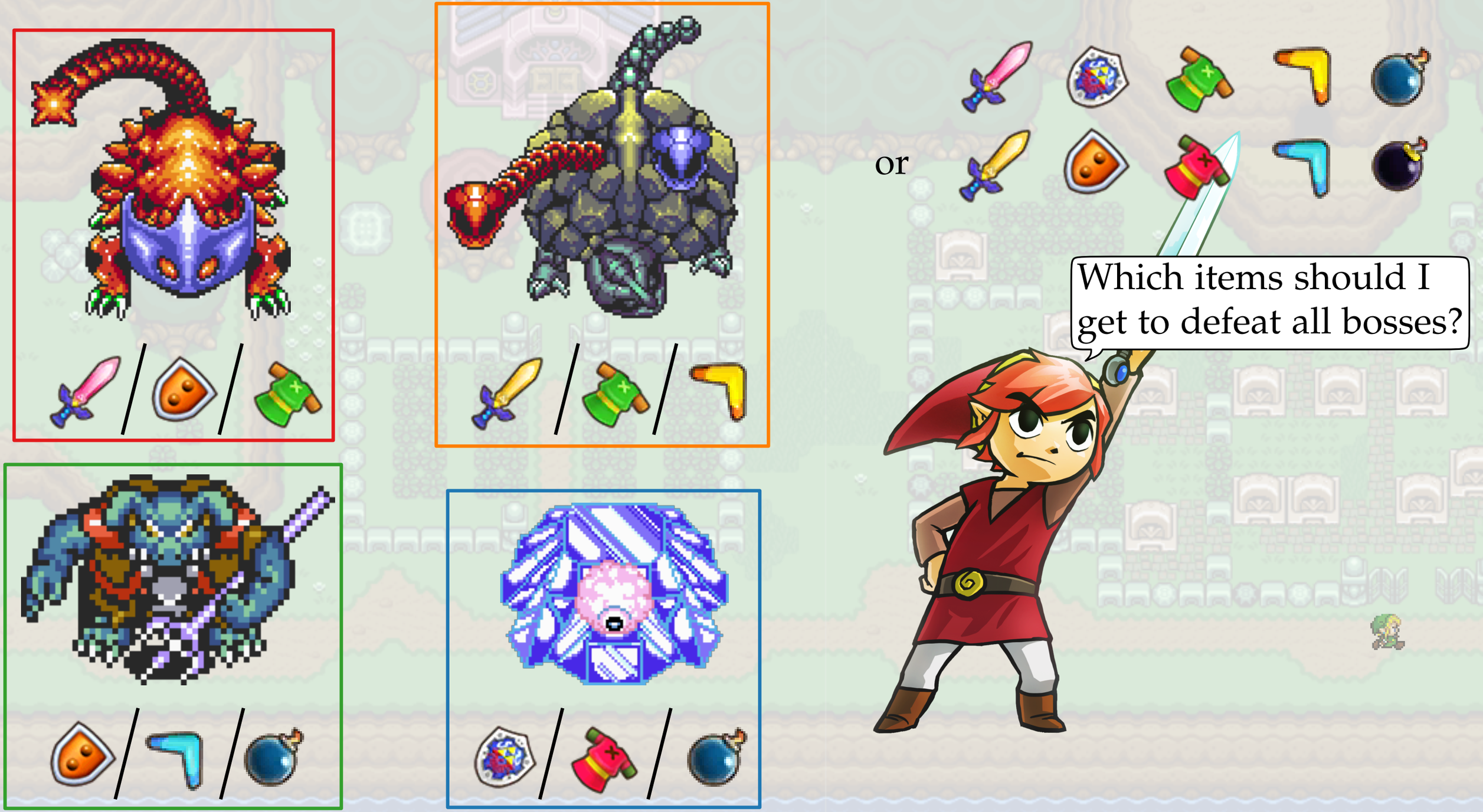
or

or

or

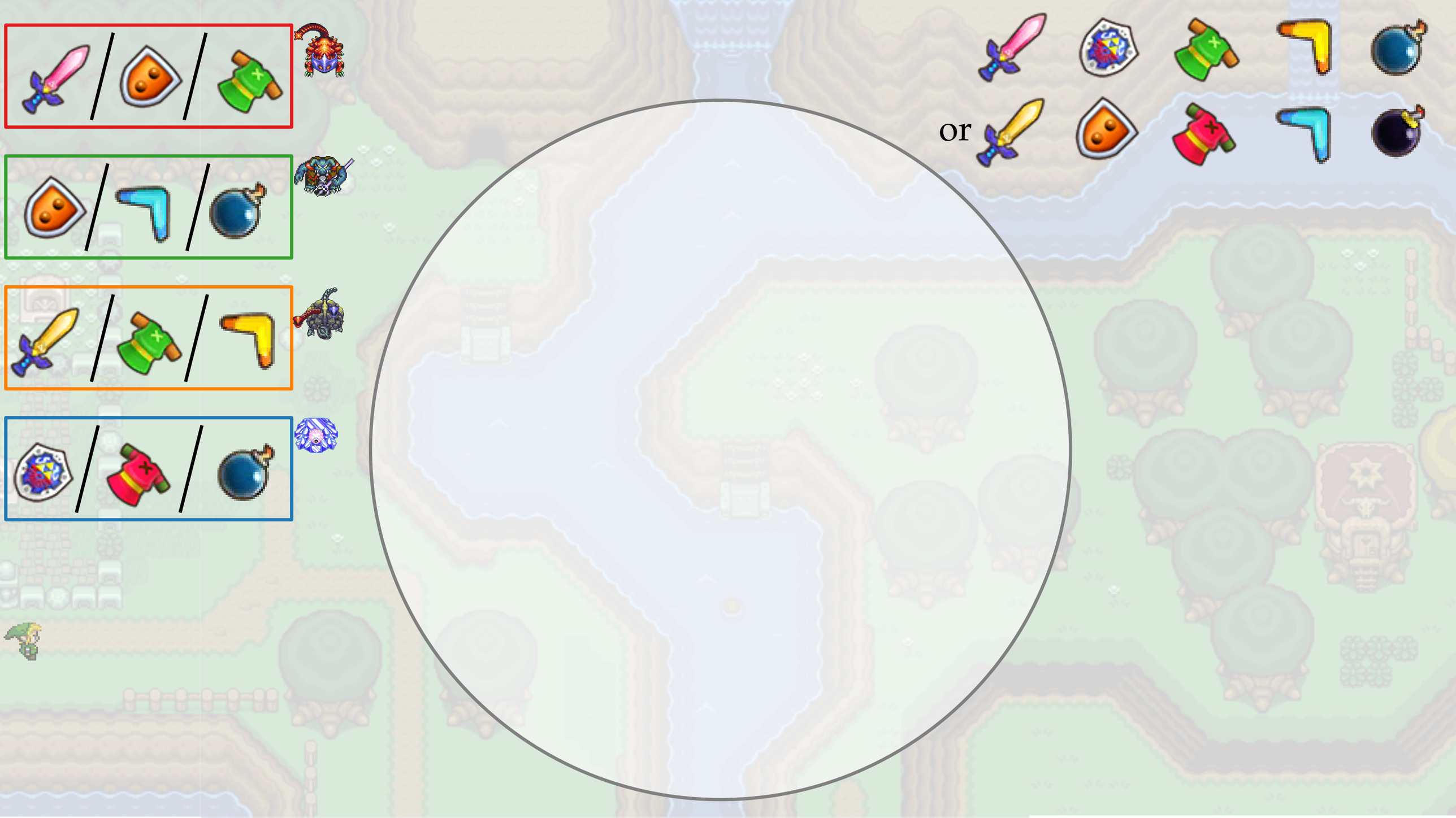
or

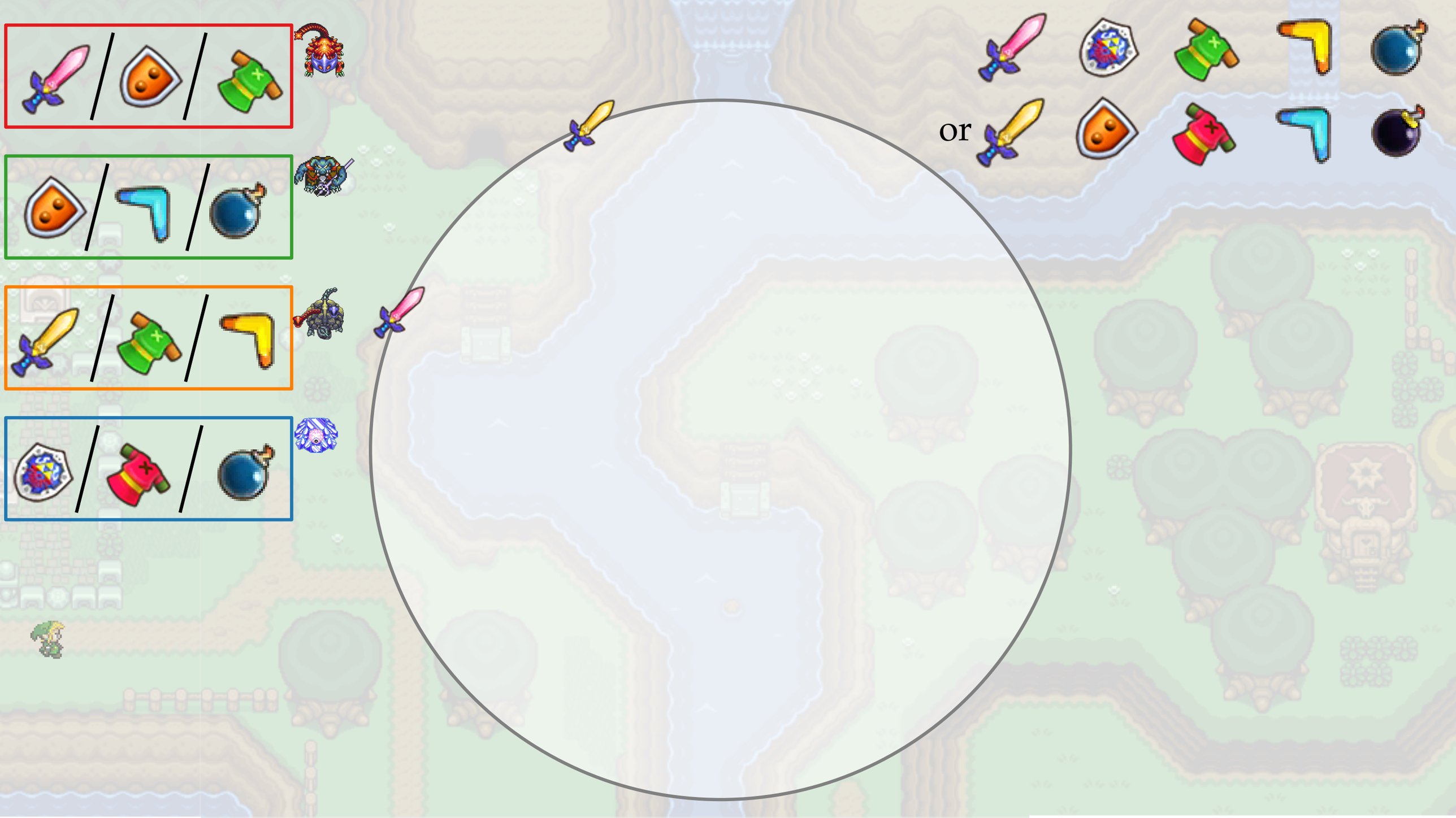
or

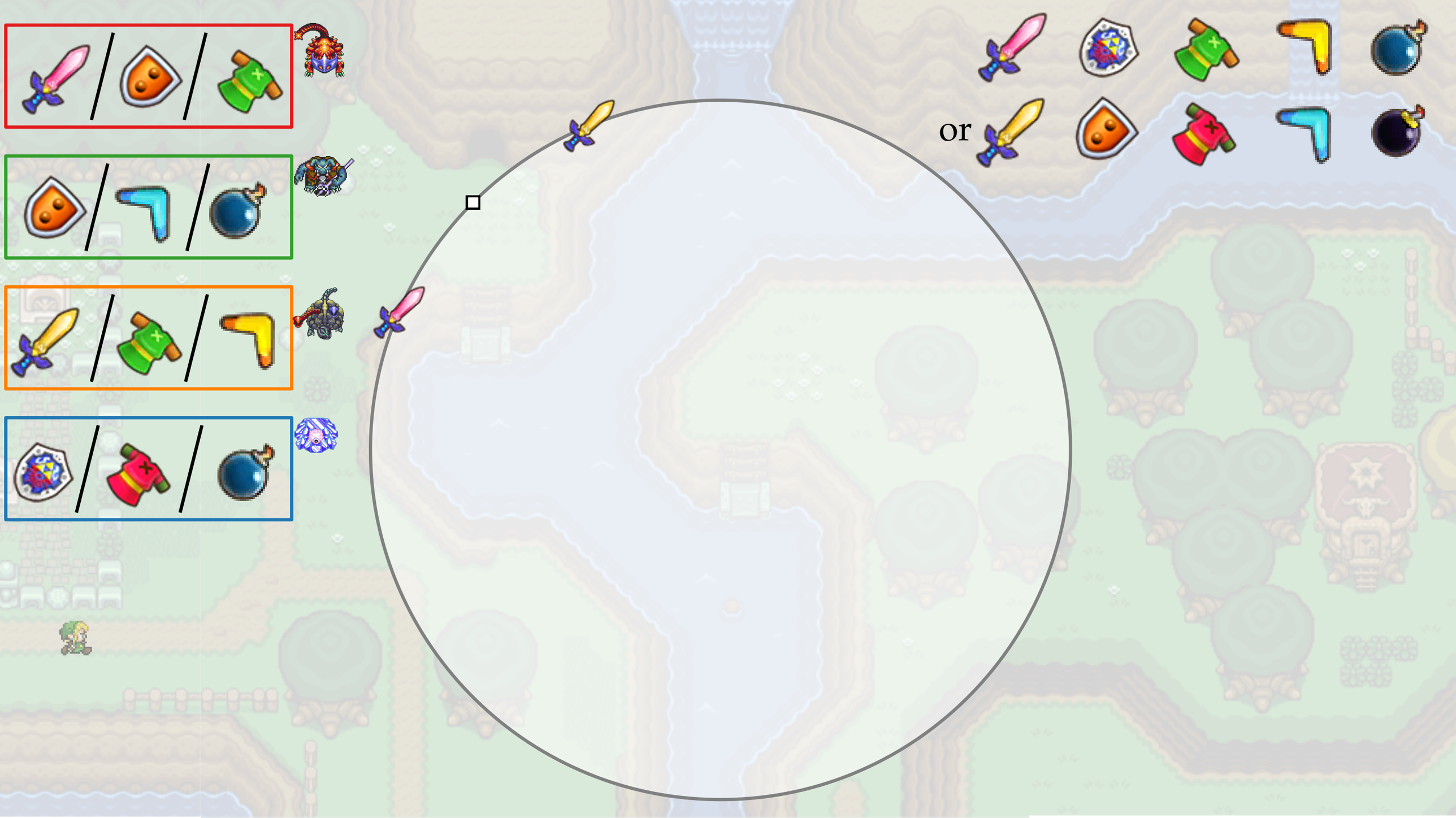

or
Which items should I get to defeat all bosses?

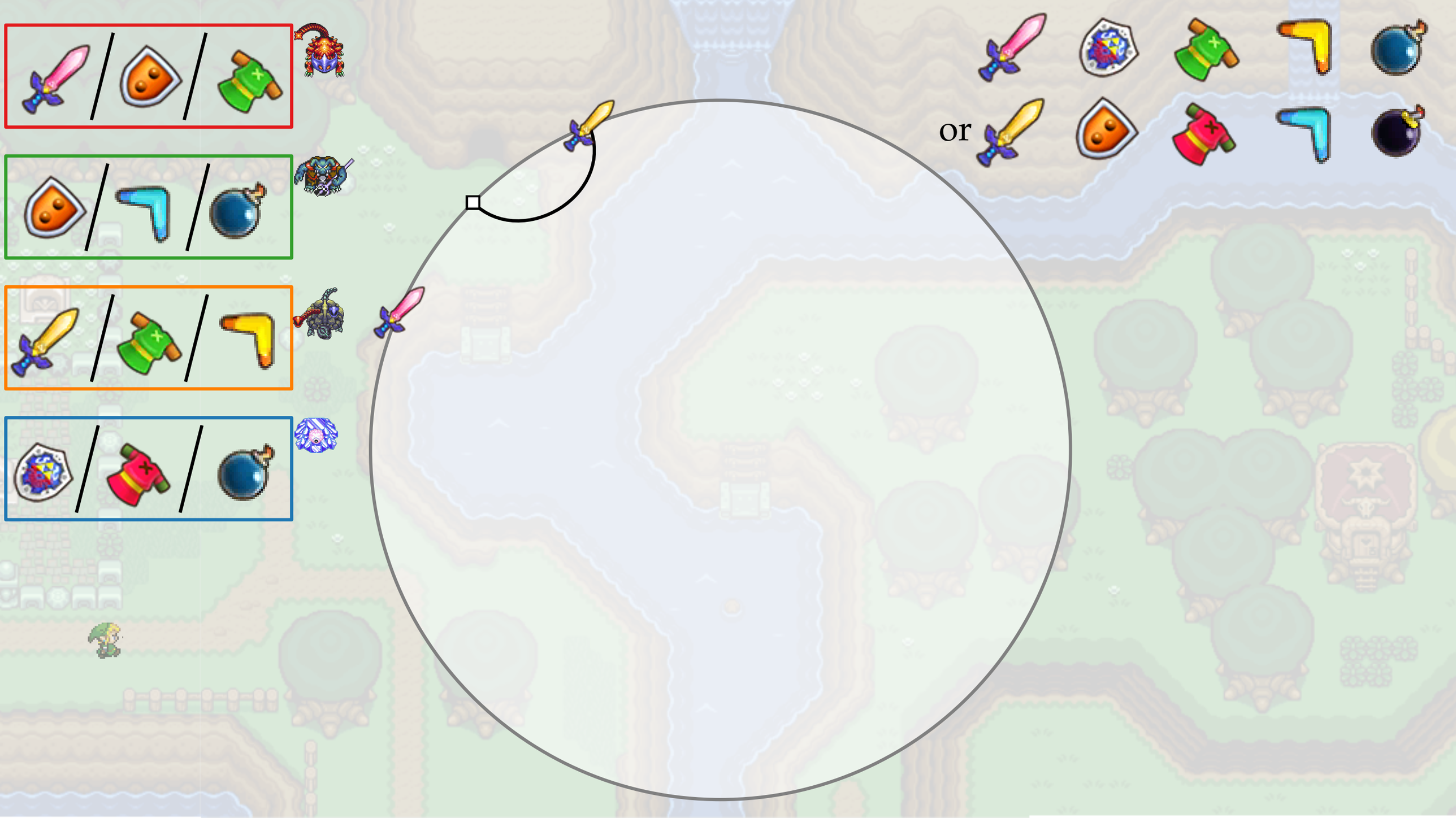
or
Which items should I get to defeat all bosses?
Is there a solution for the 3-SAT instance?

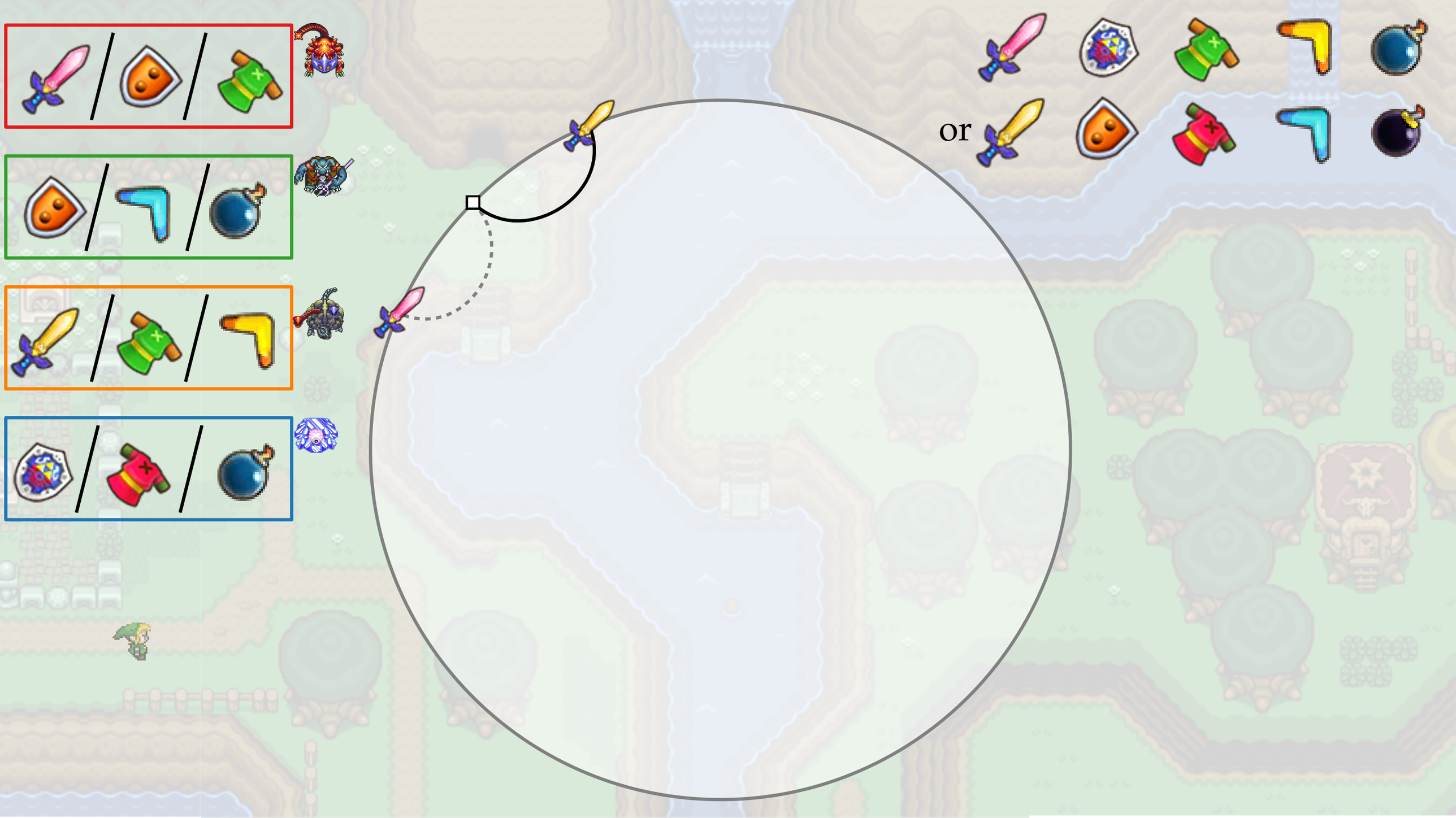
or

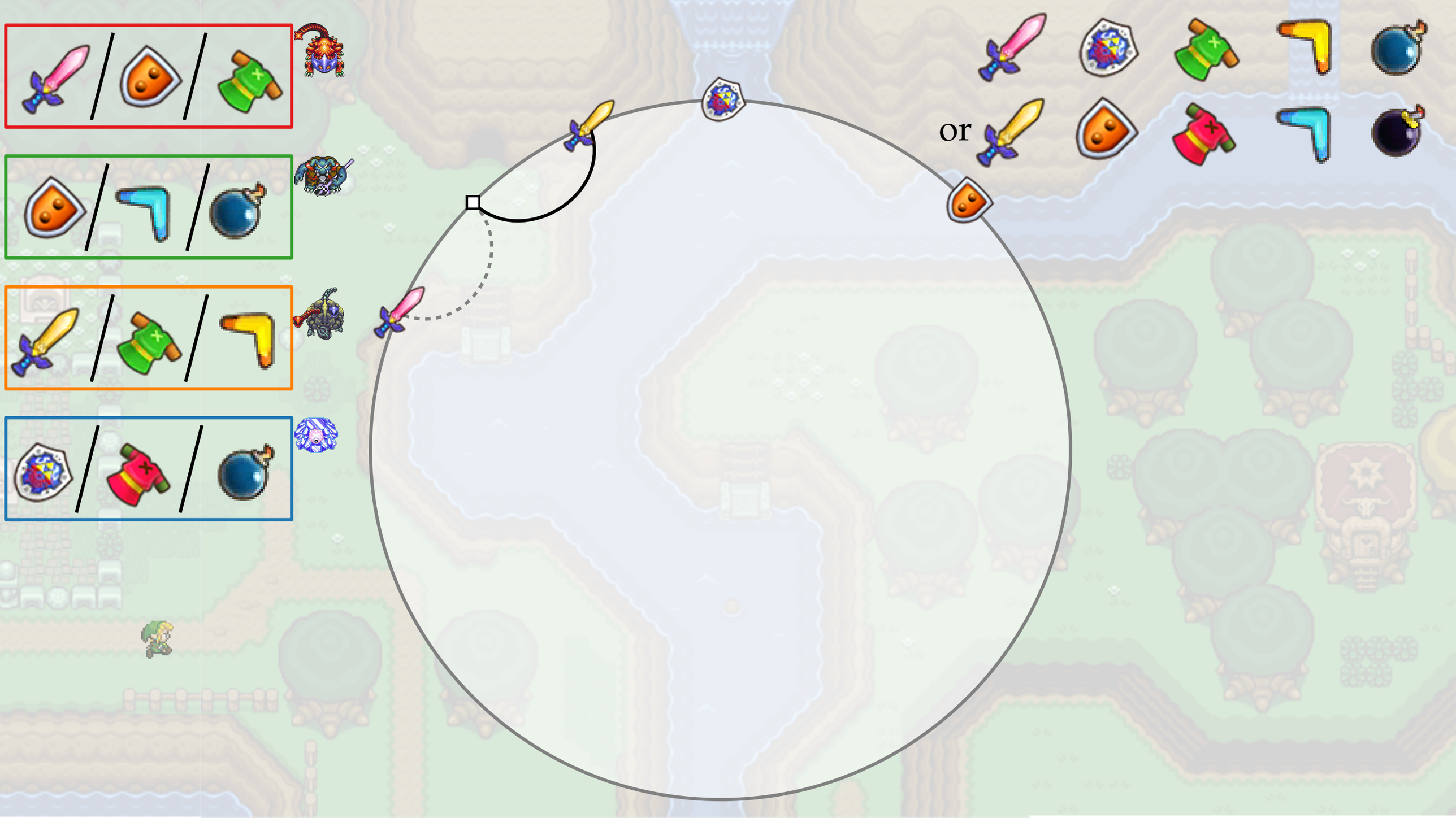

or

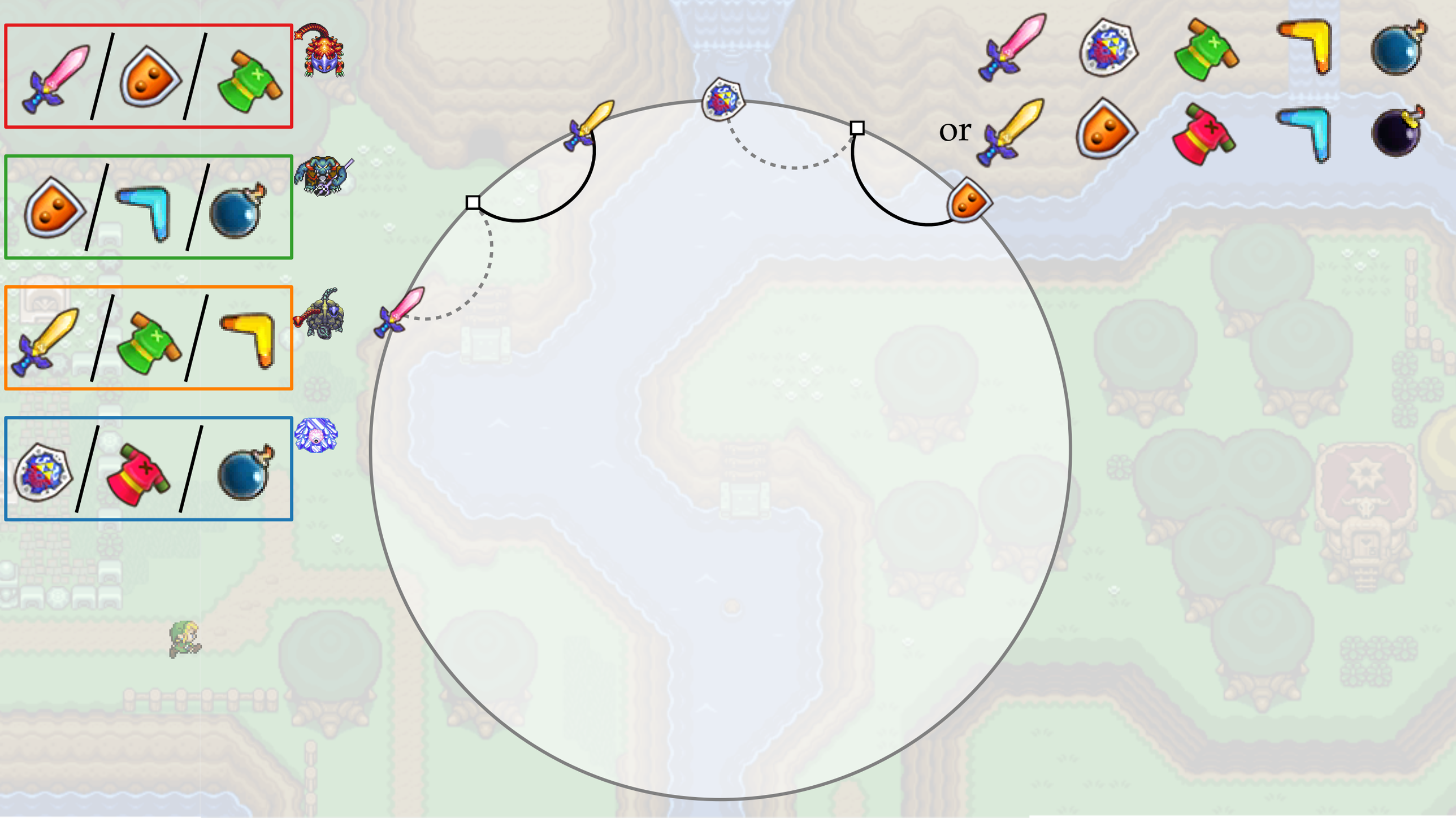

or

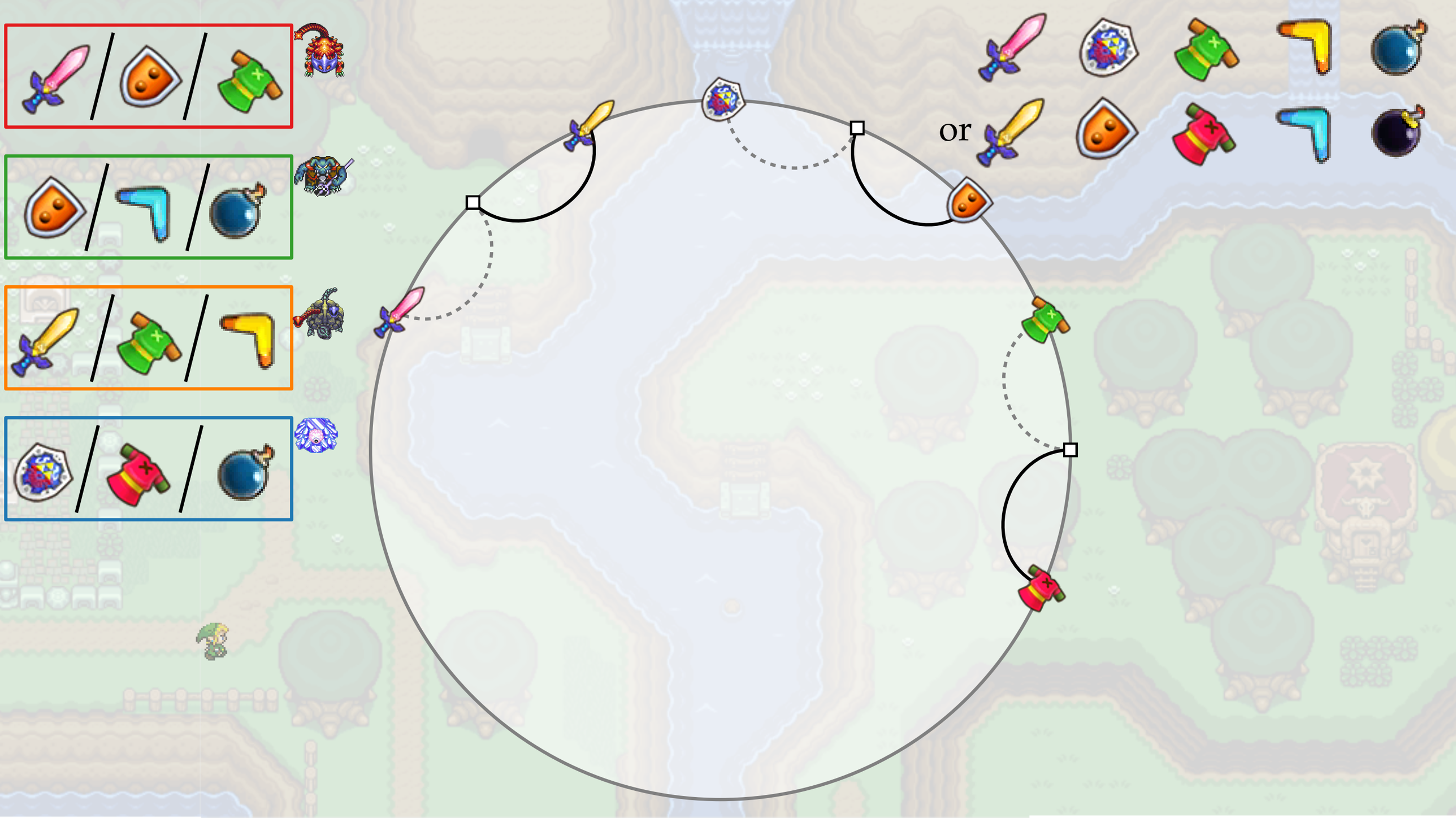

or

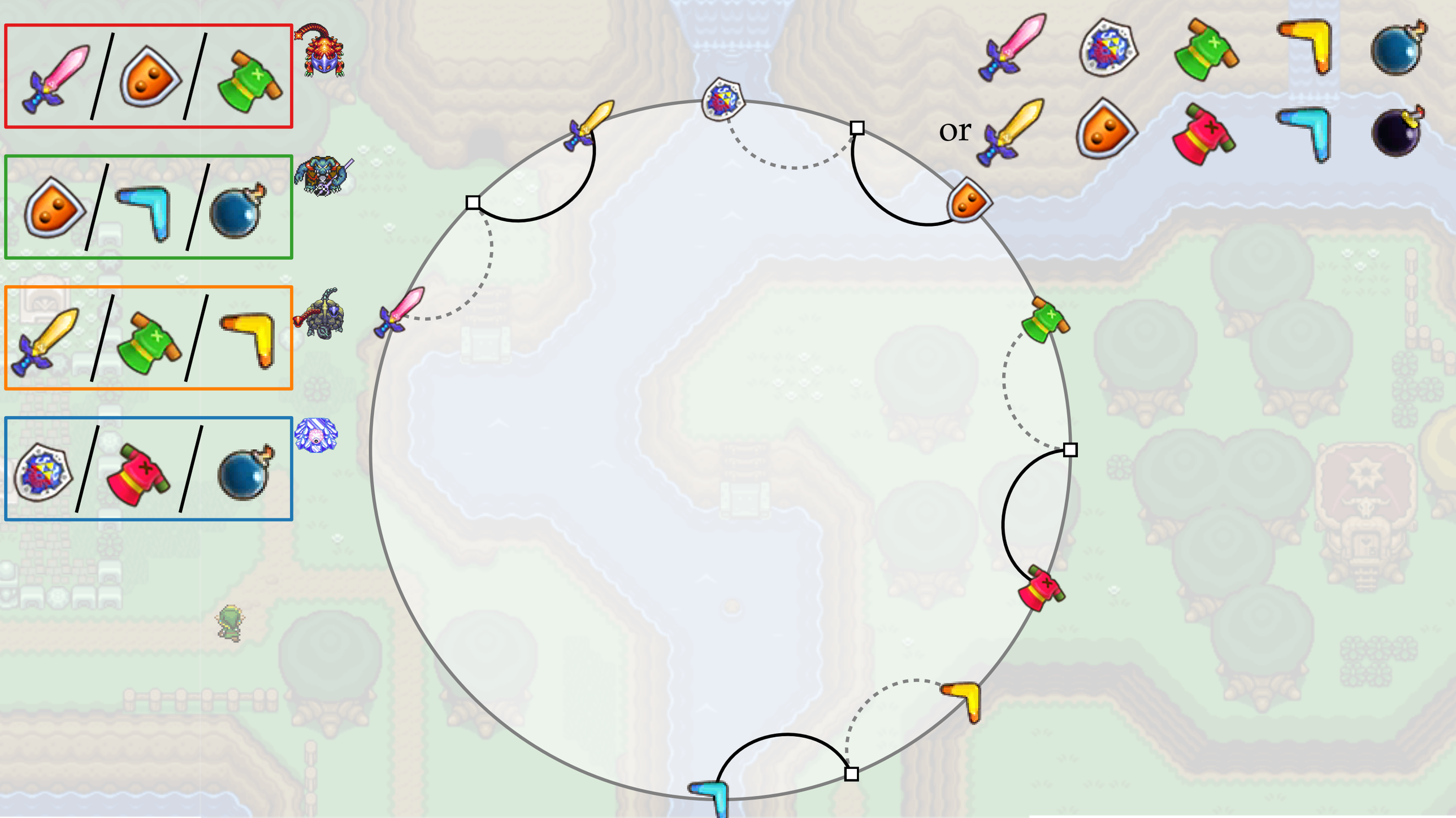

or

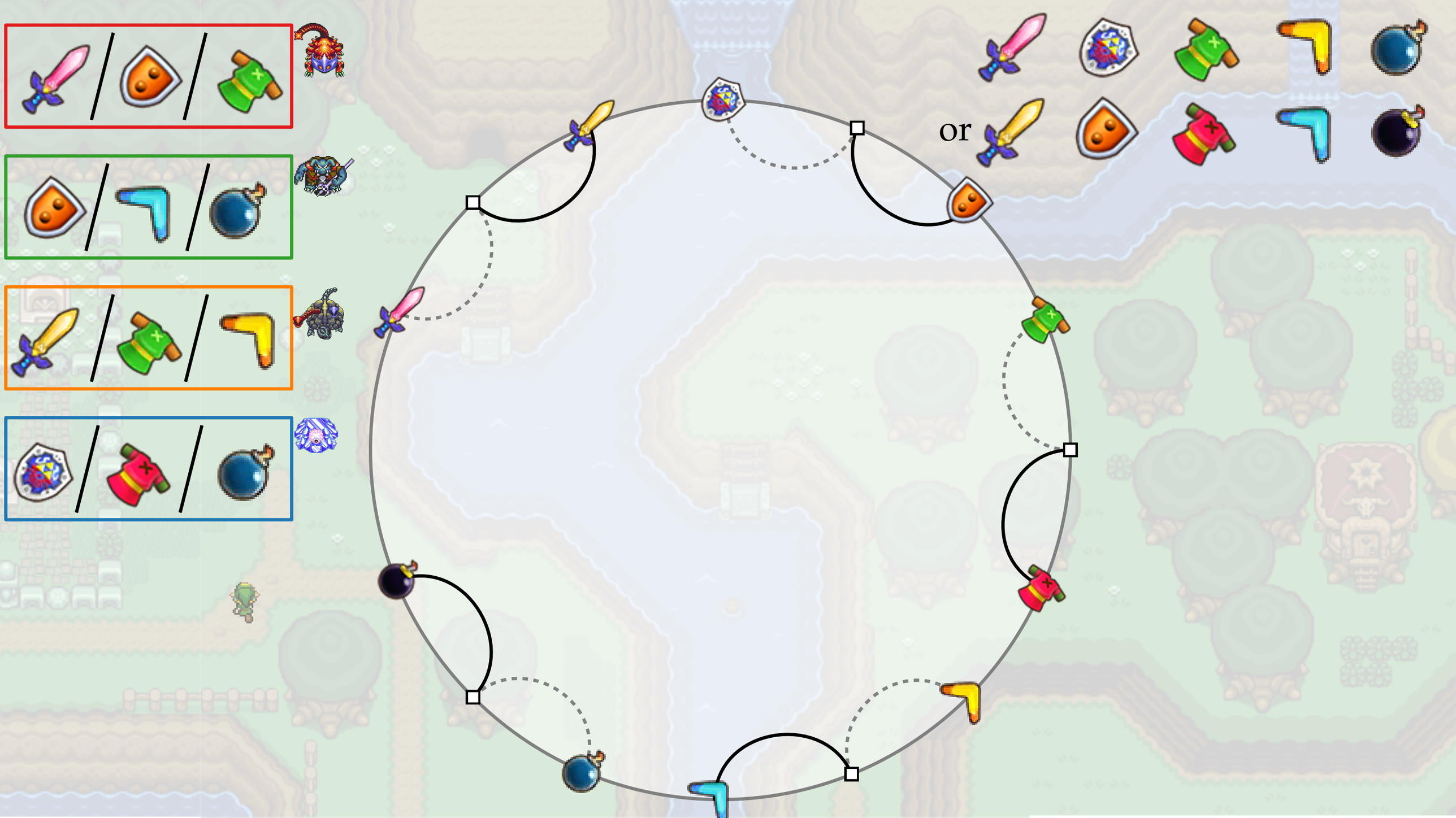

or

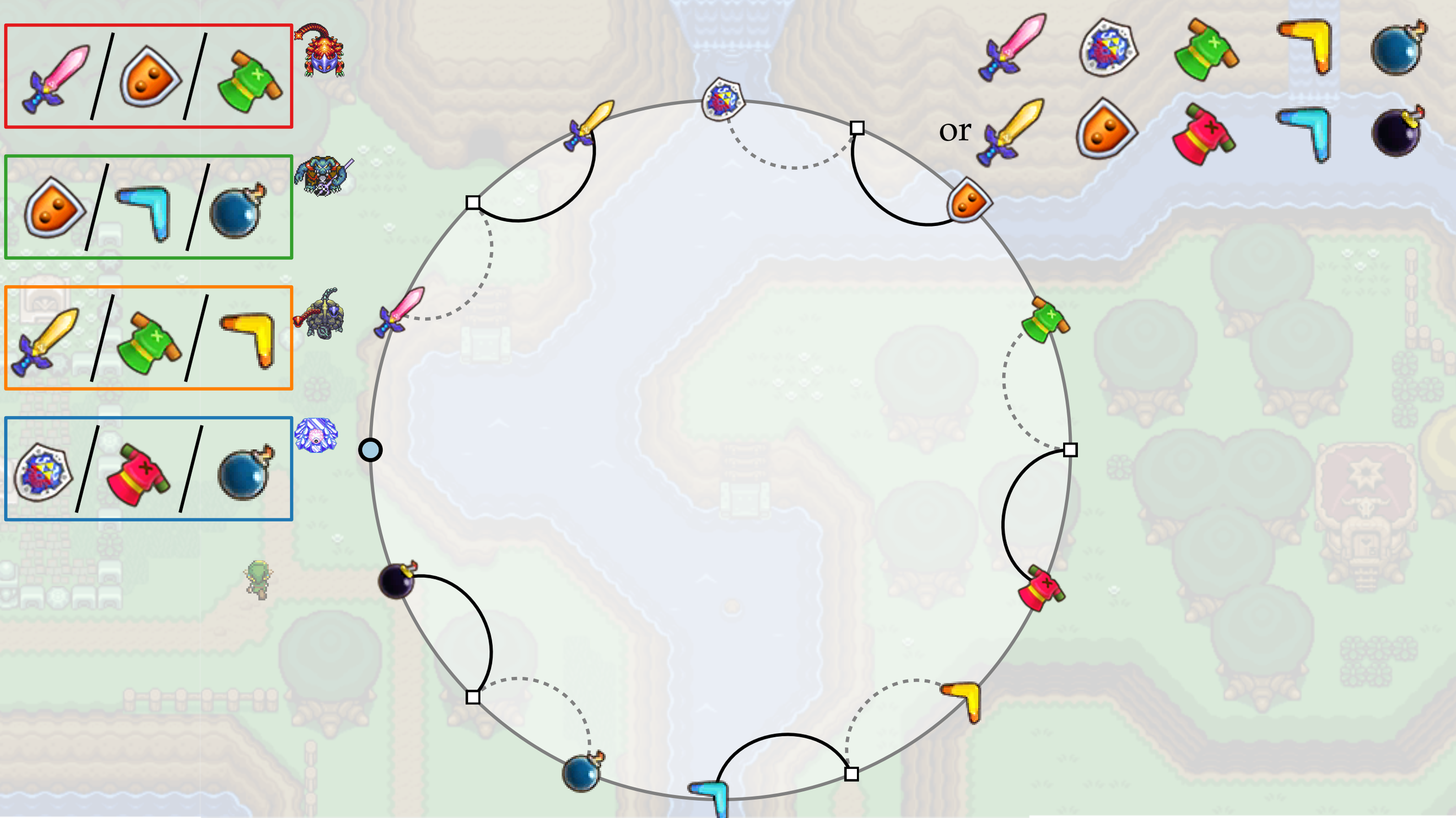

or

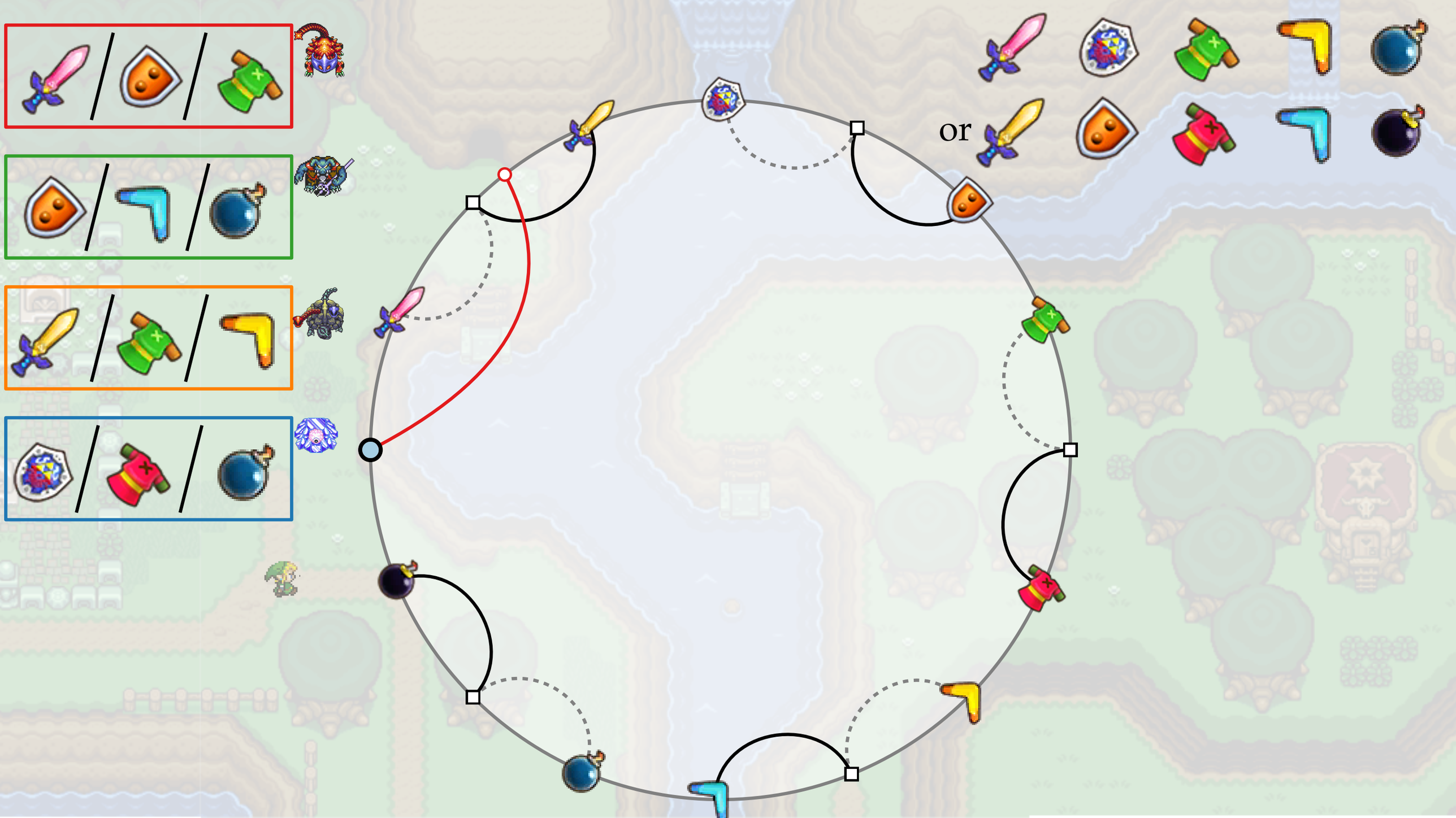

or

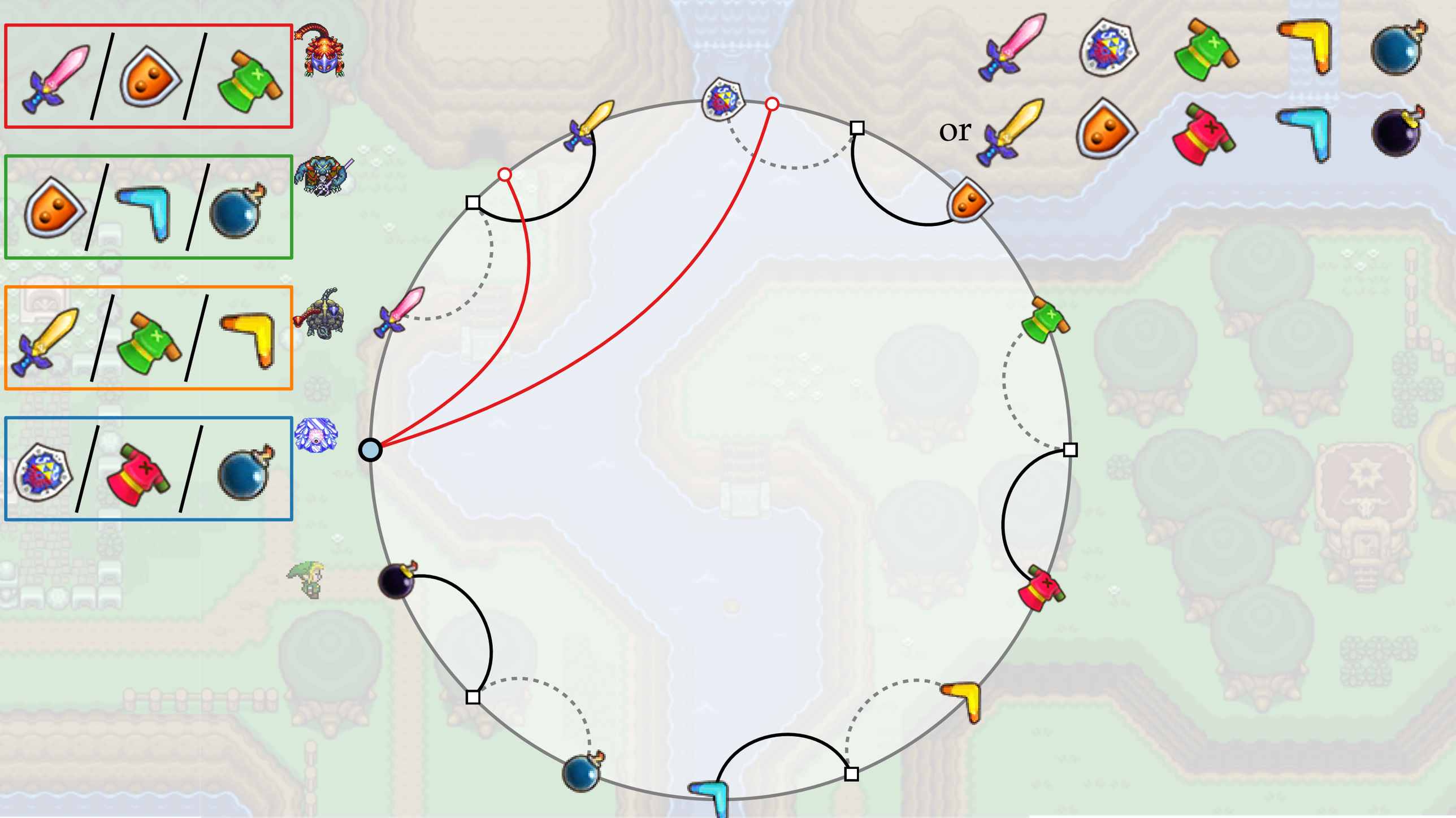

or

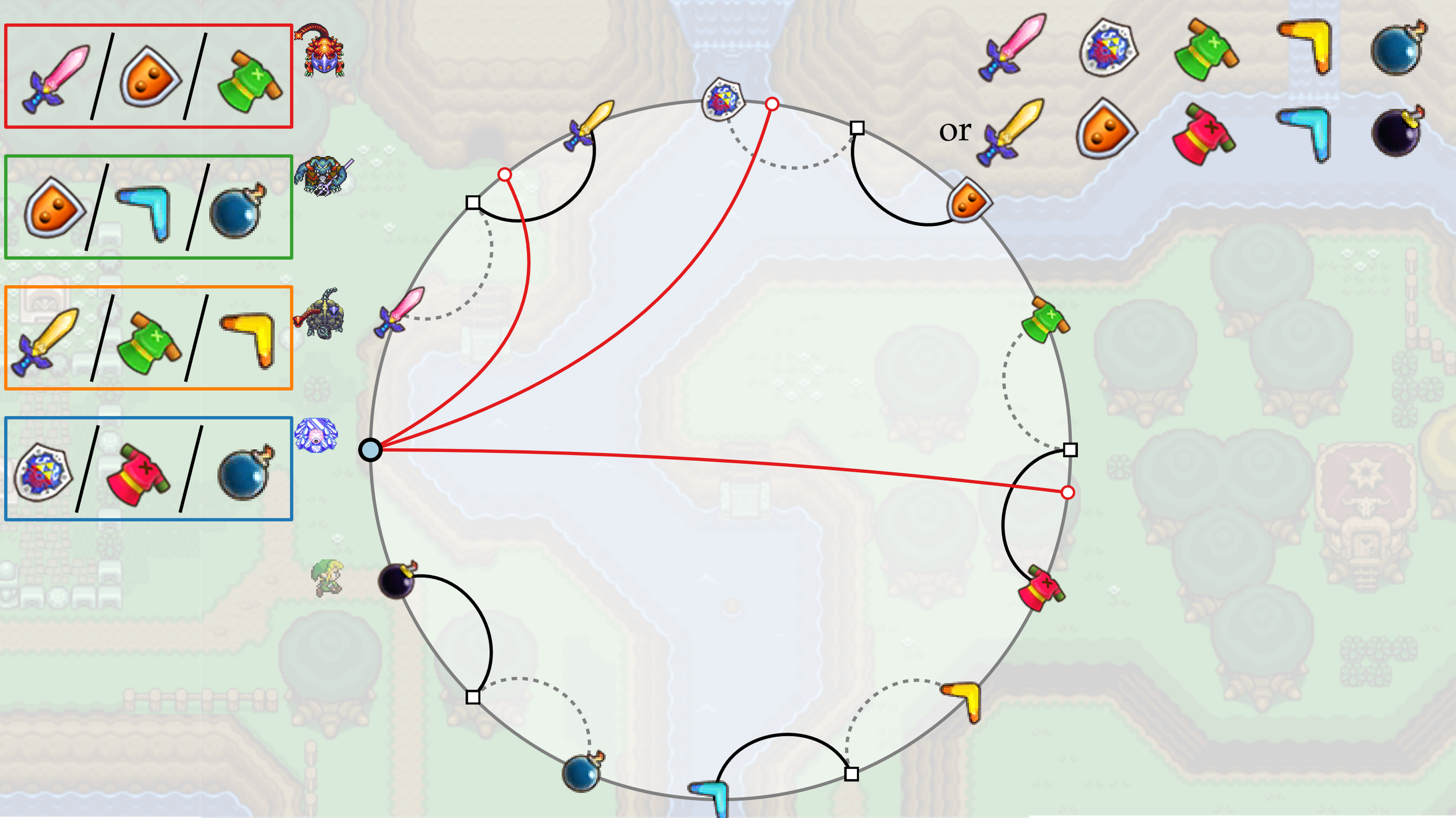

or

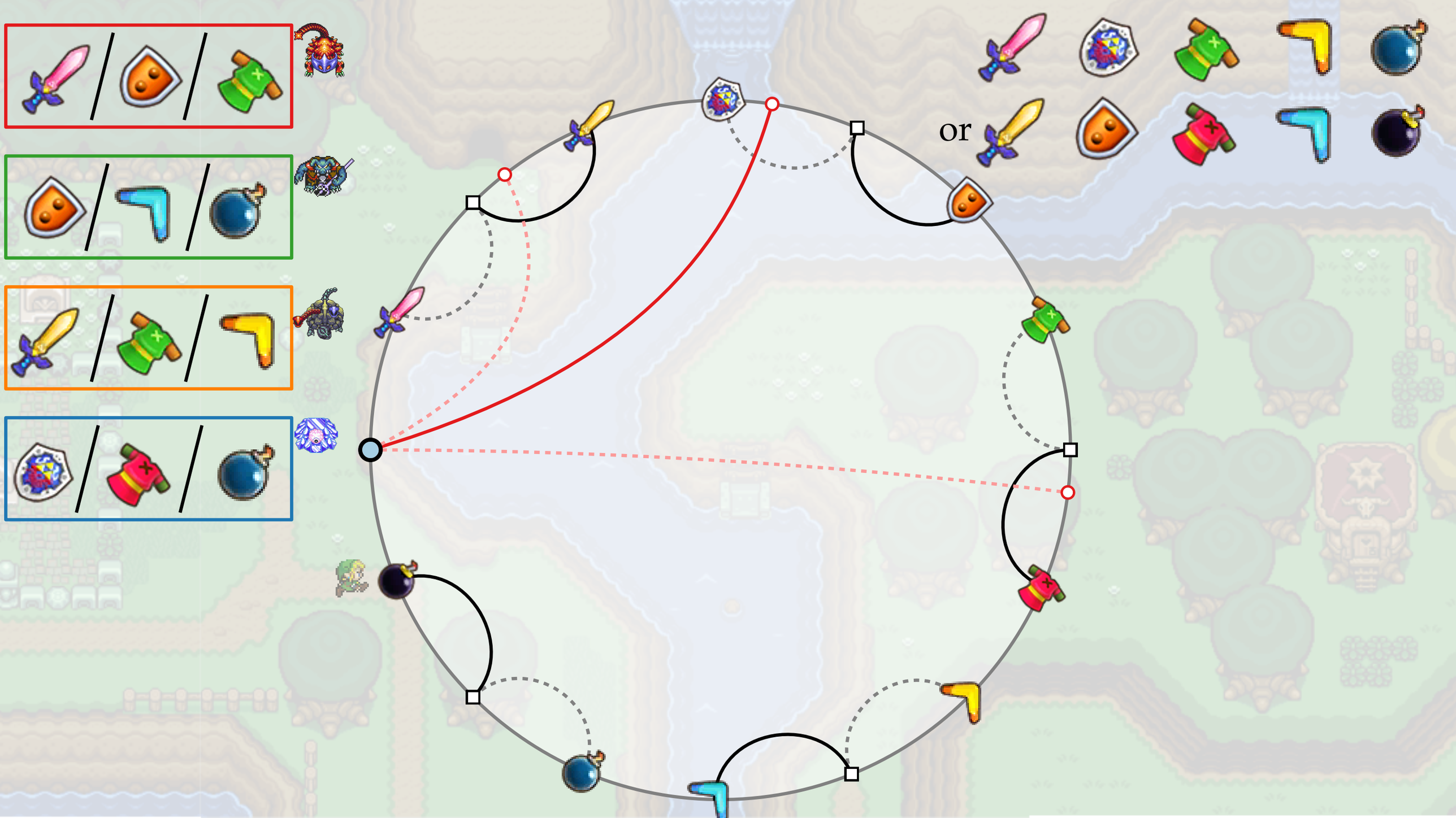

or

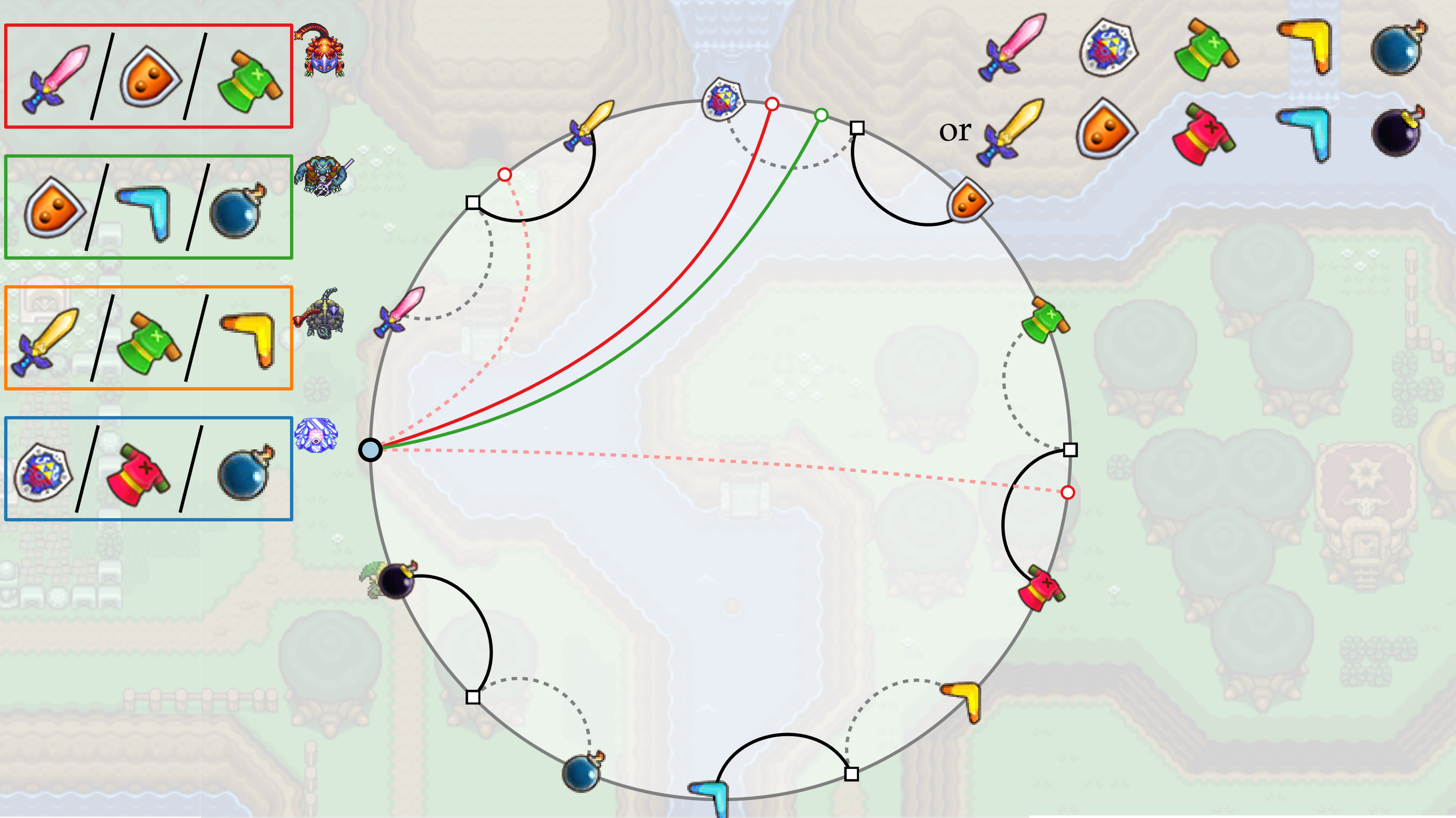

or

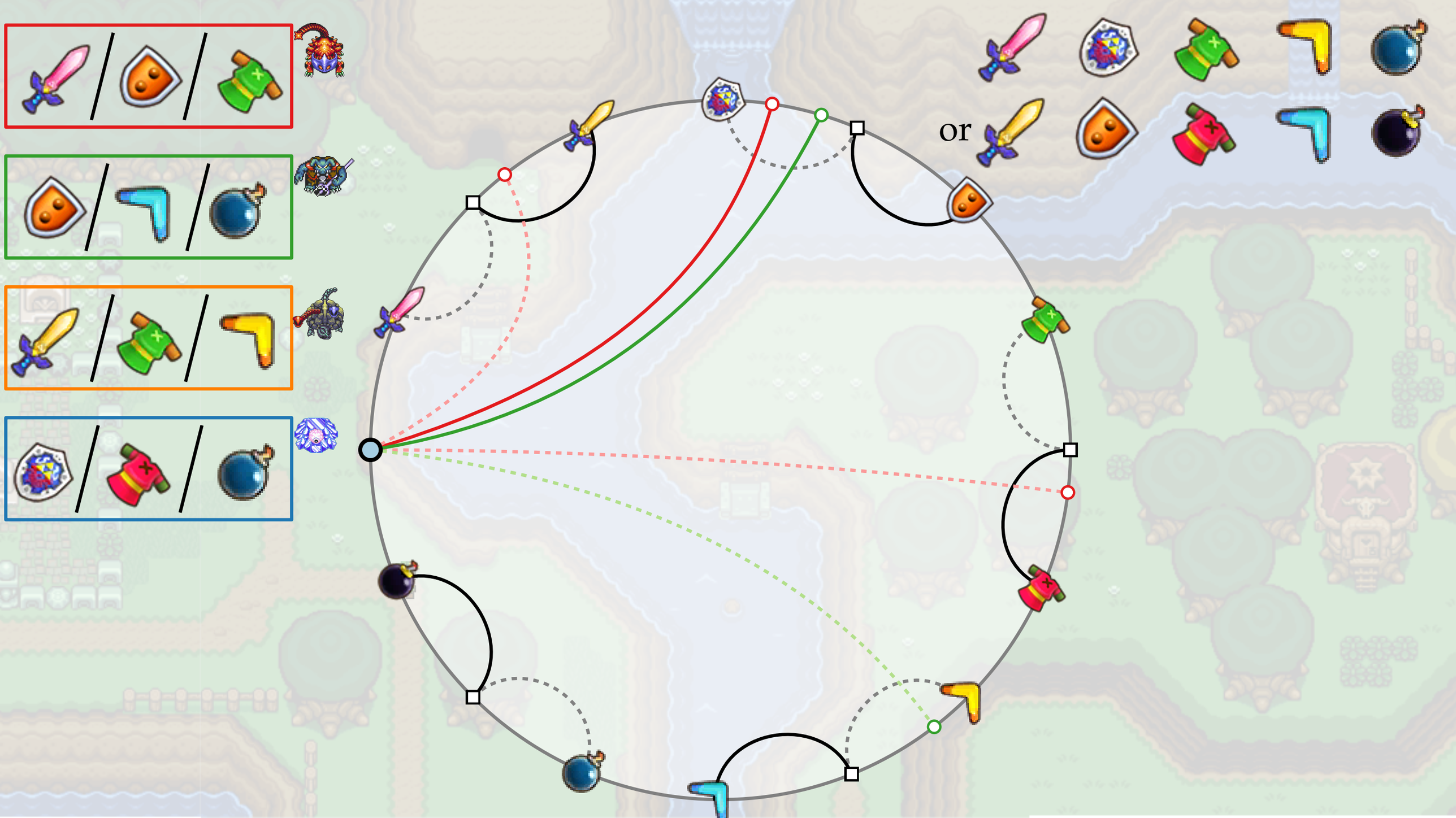

or

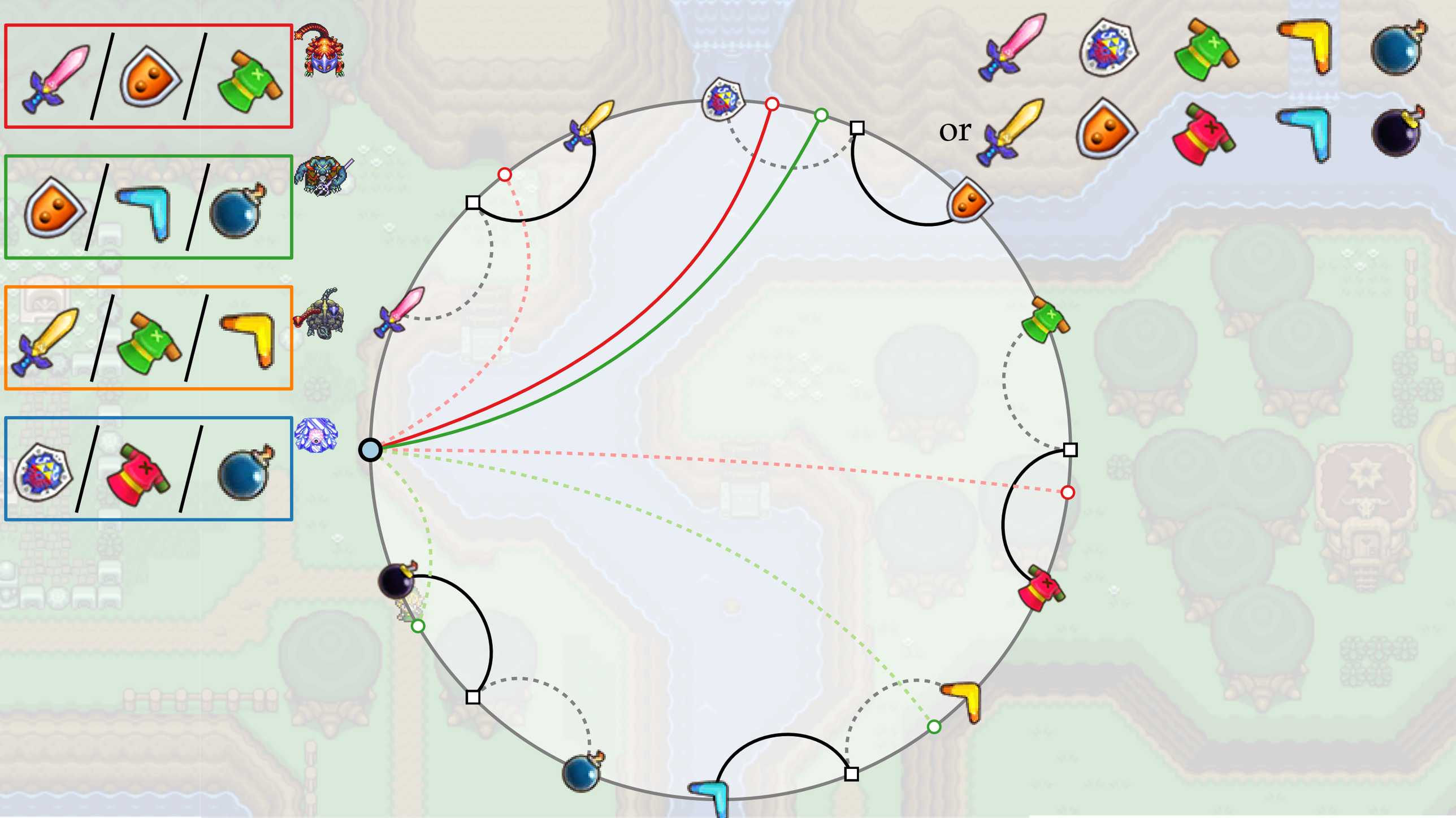

or

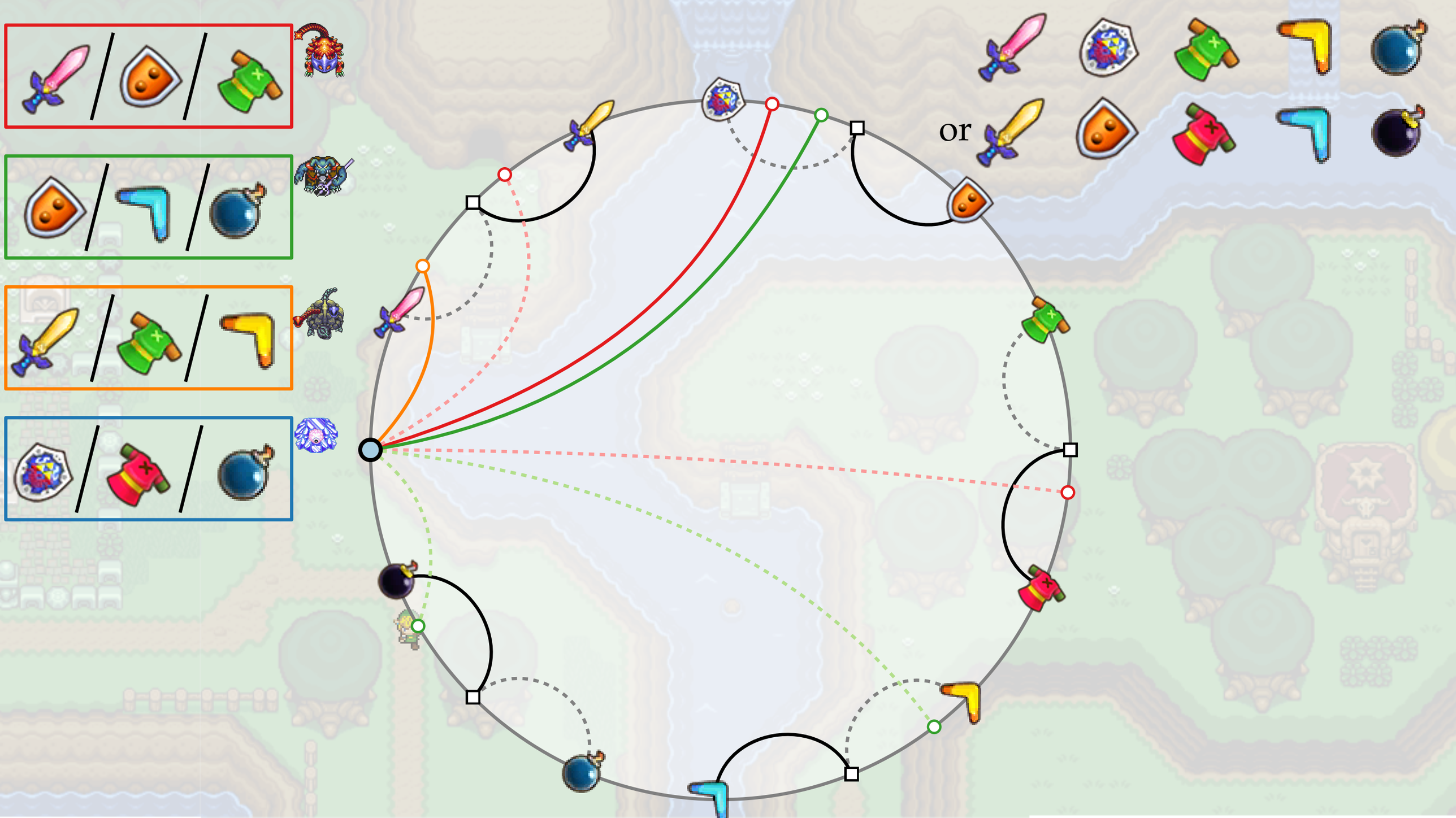

or

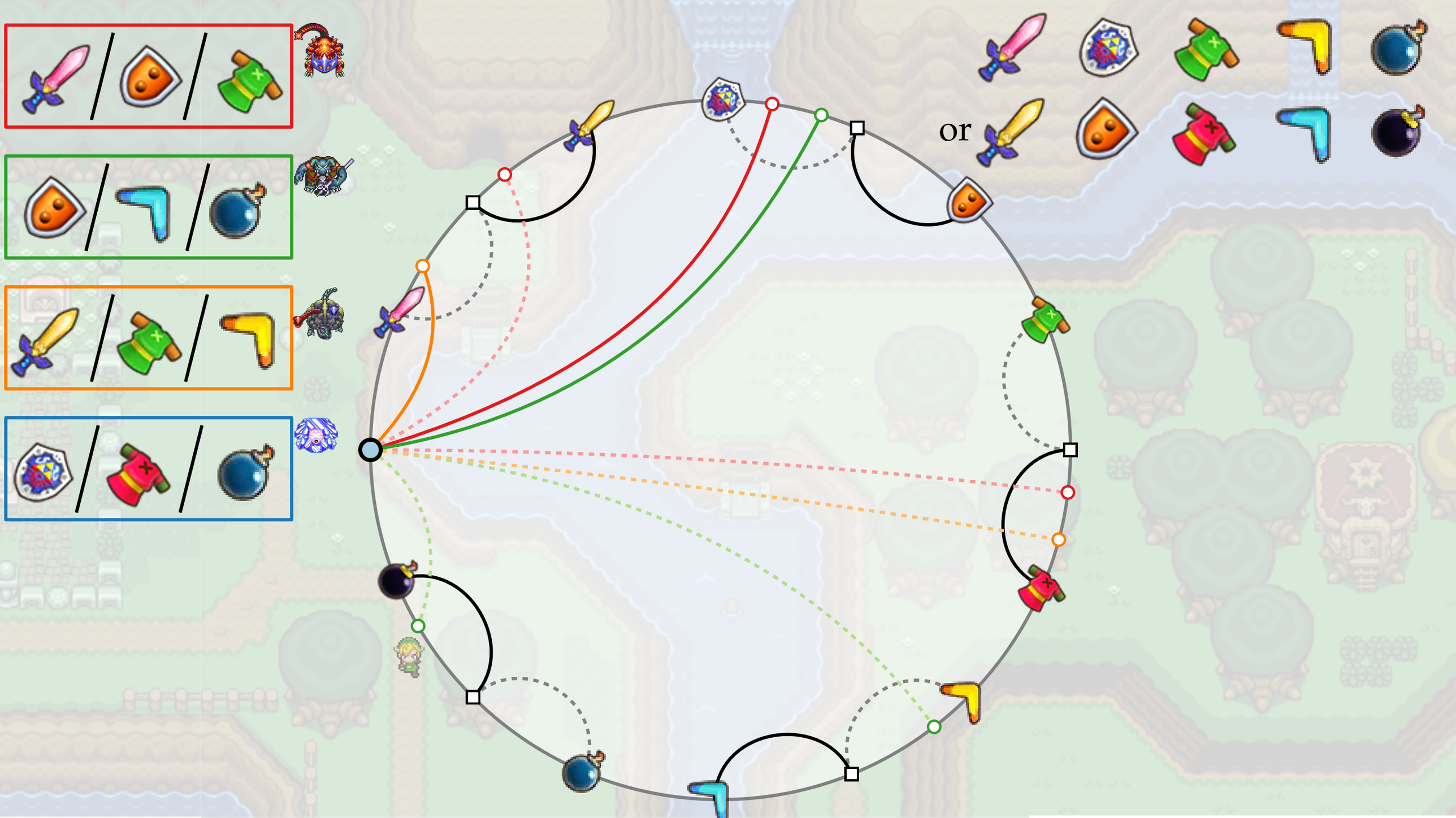

or

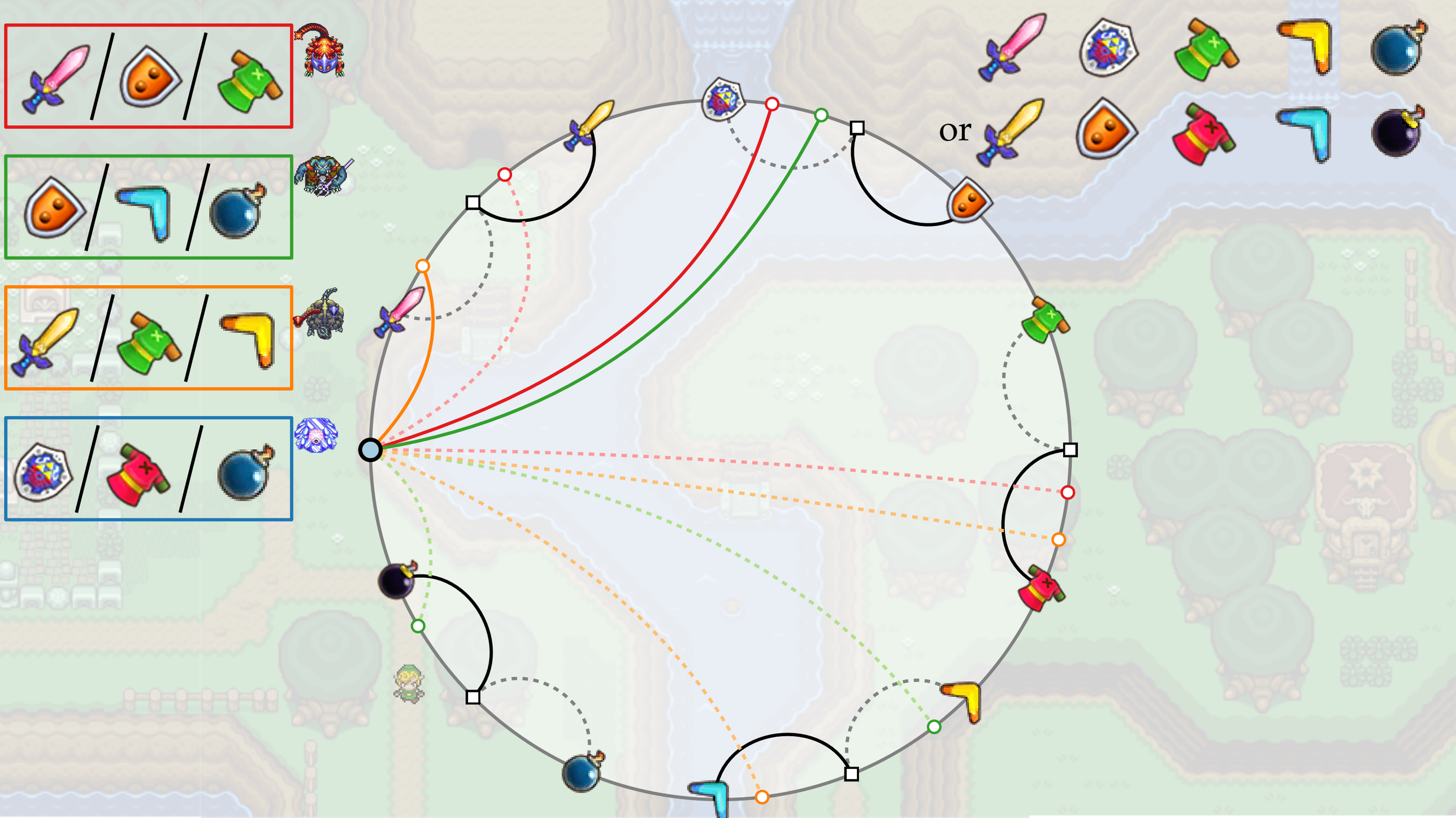

or

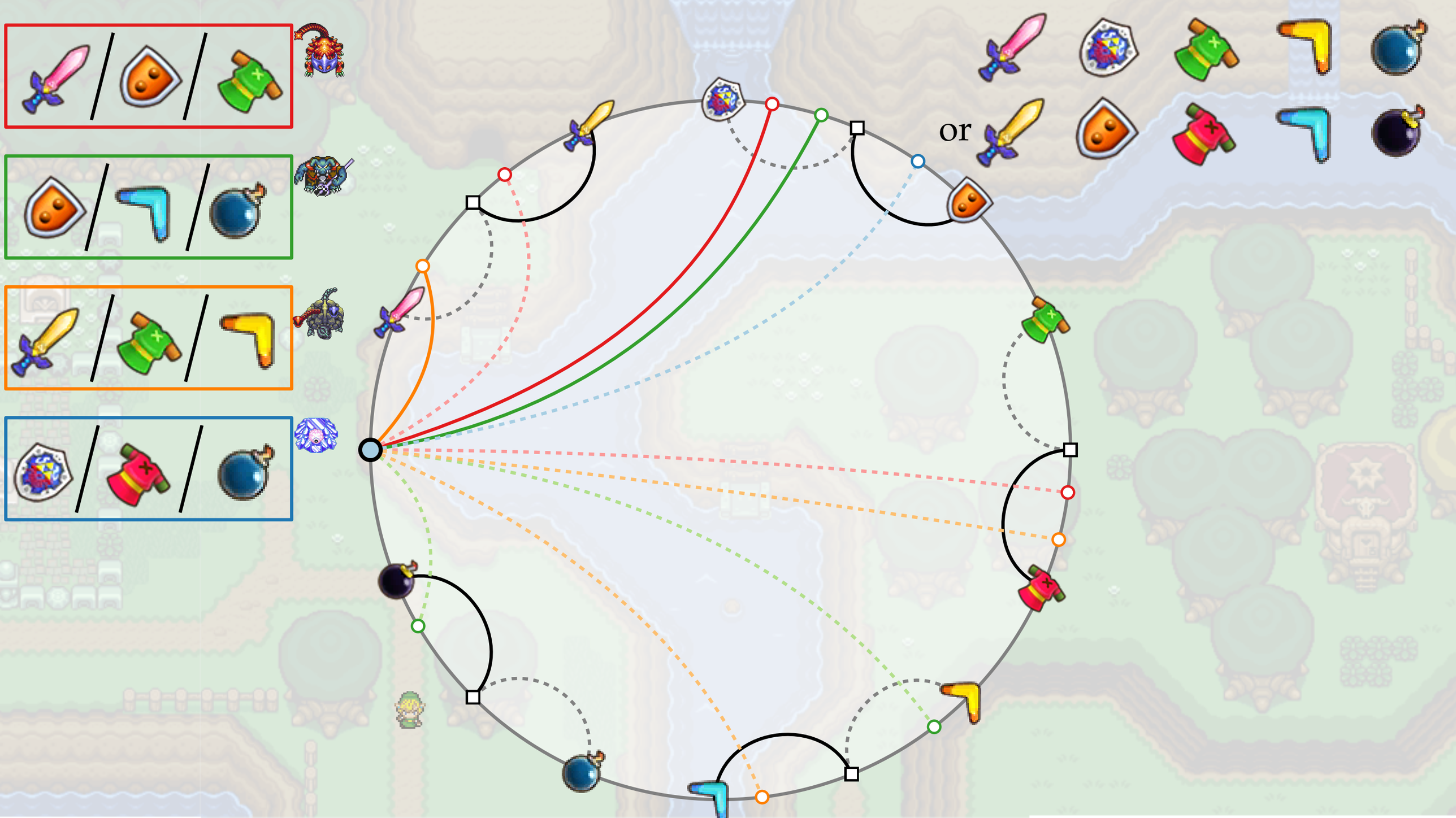

or

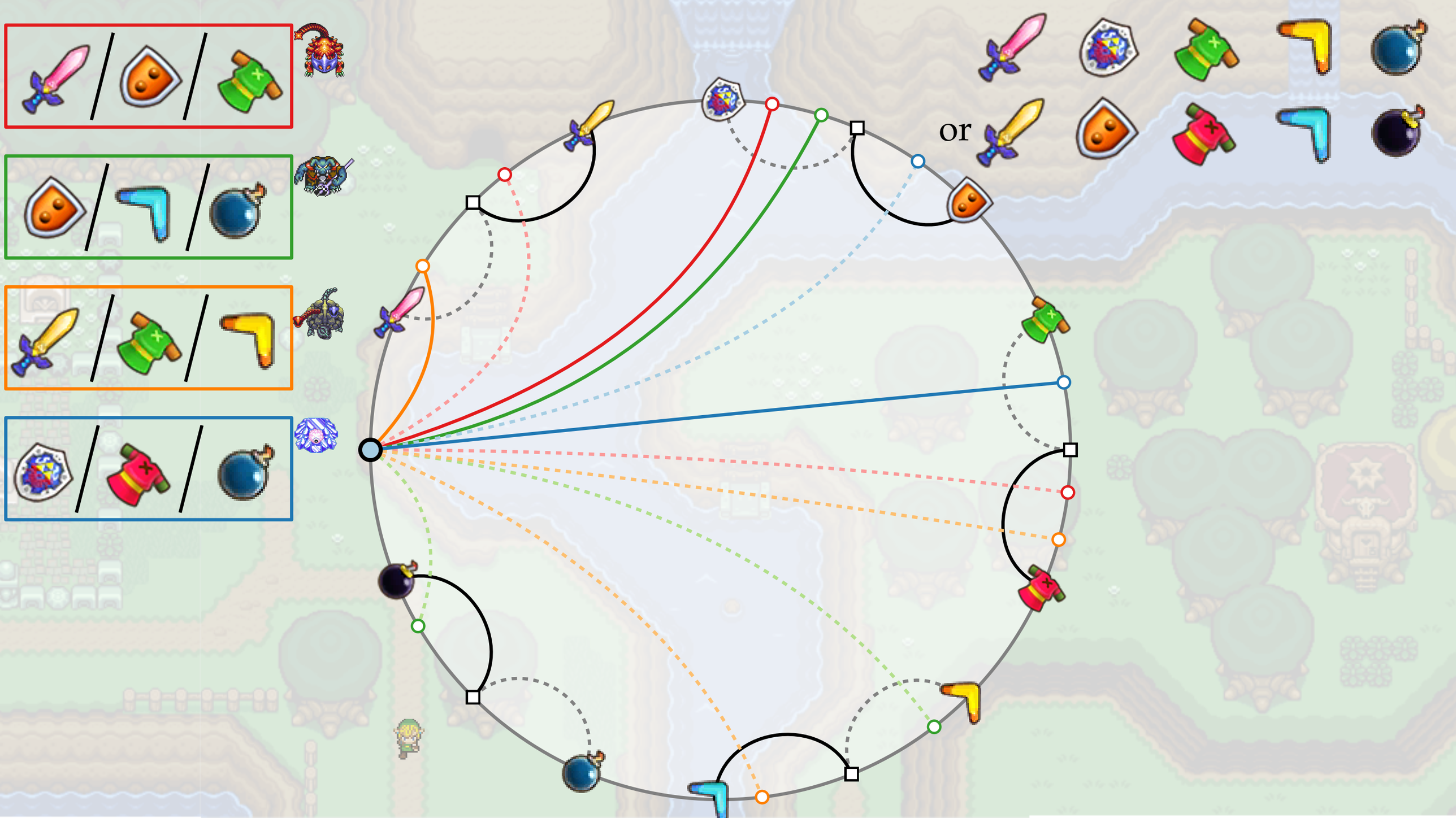

or

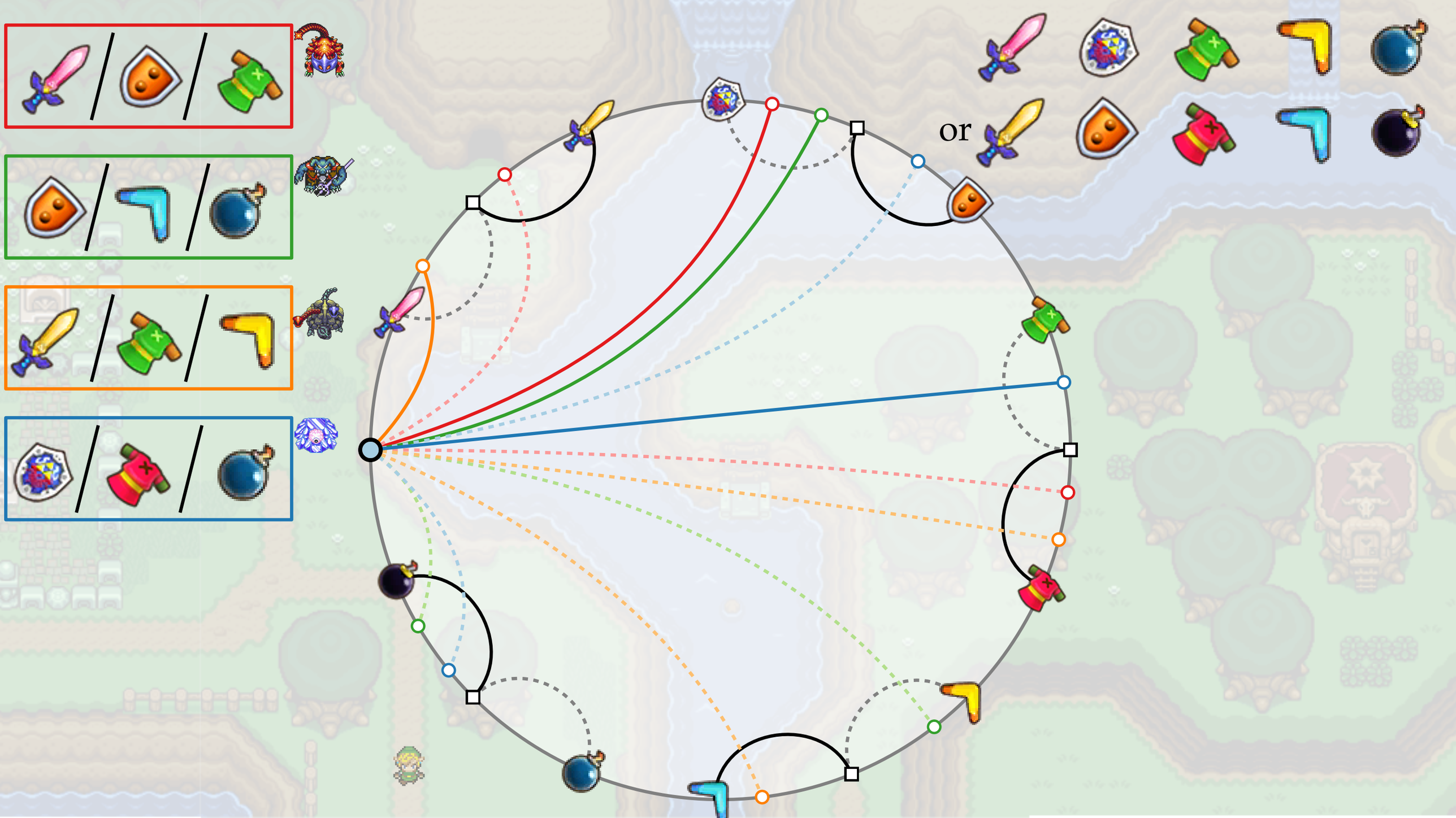

or

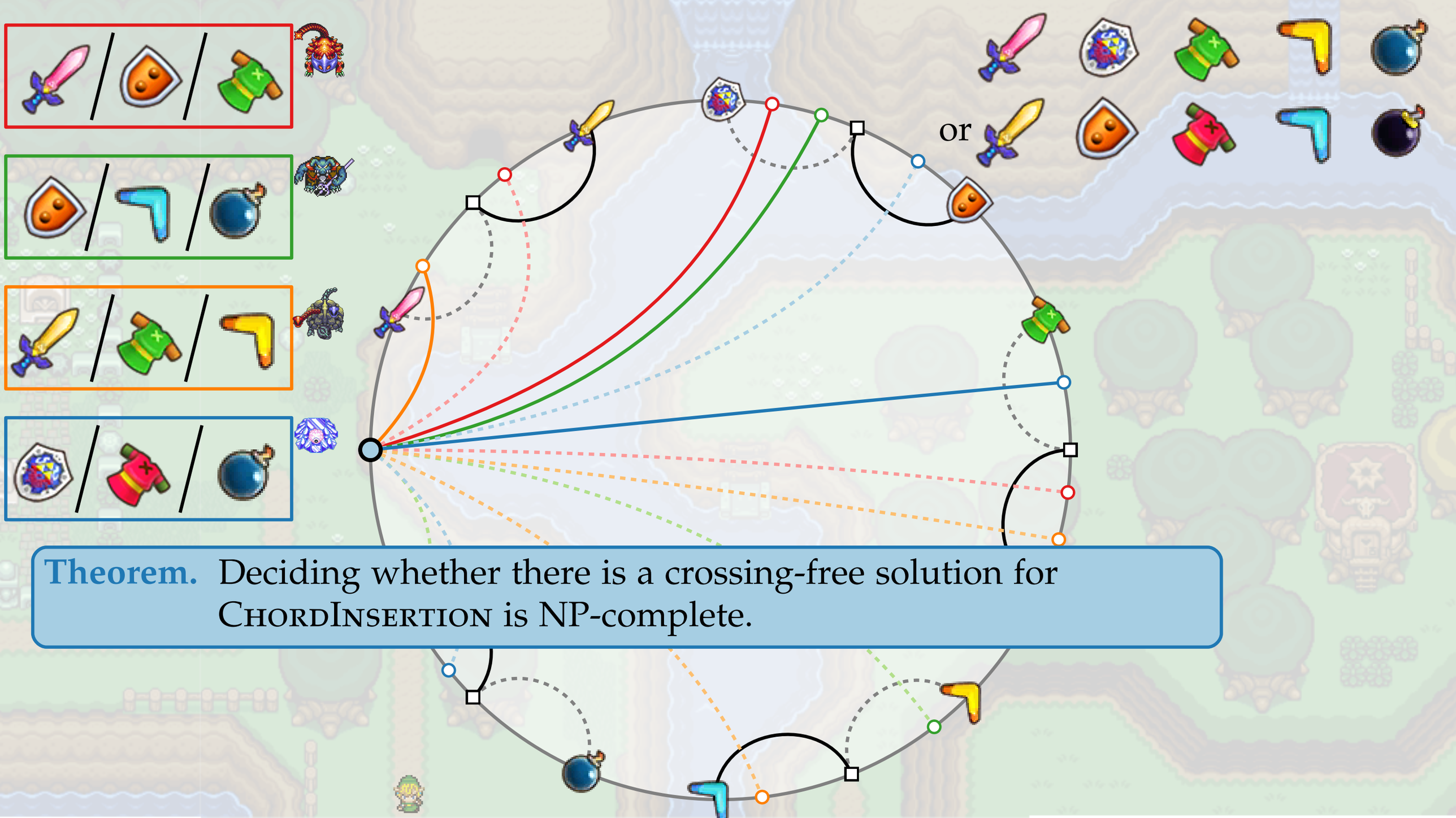

or


or

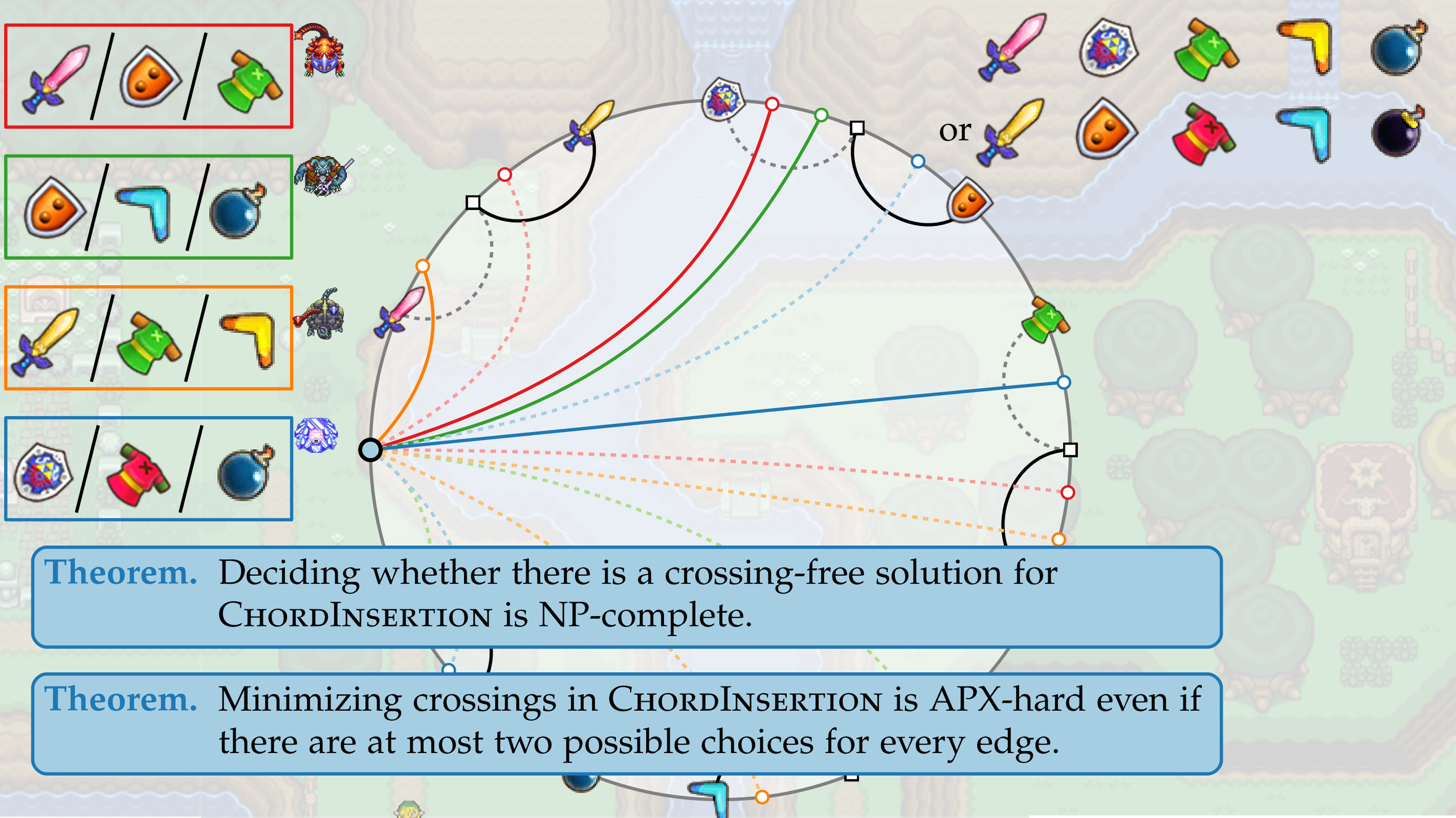

or


or


or

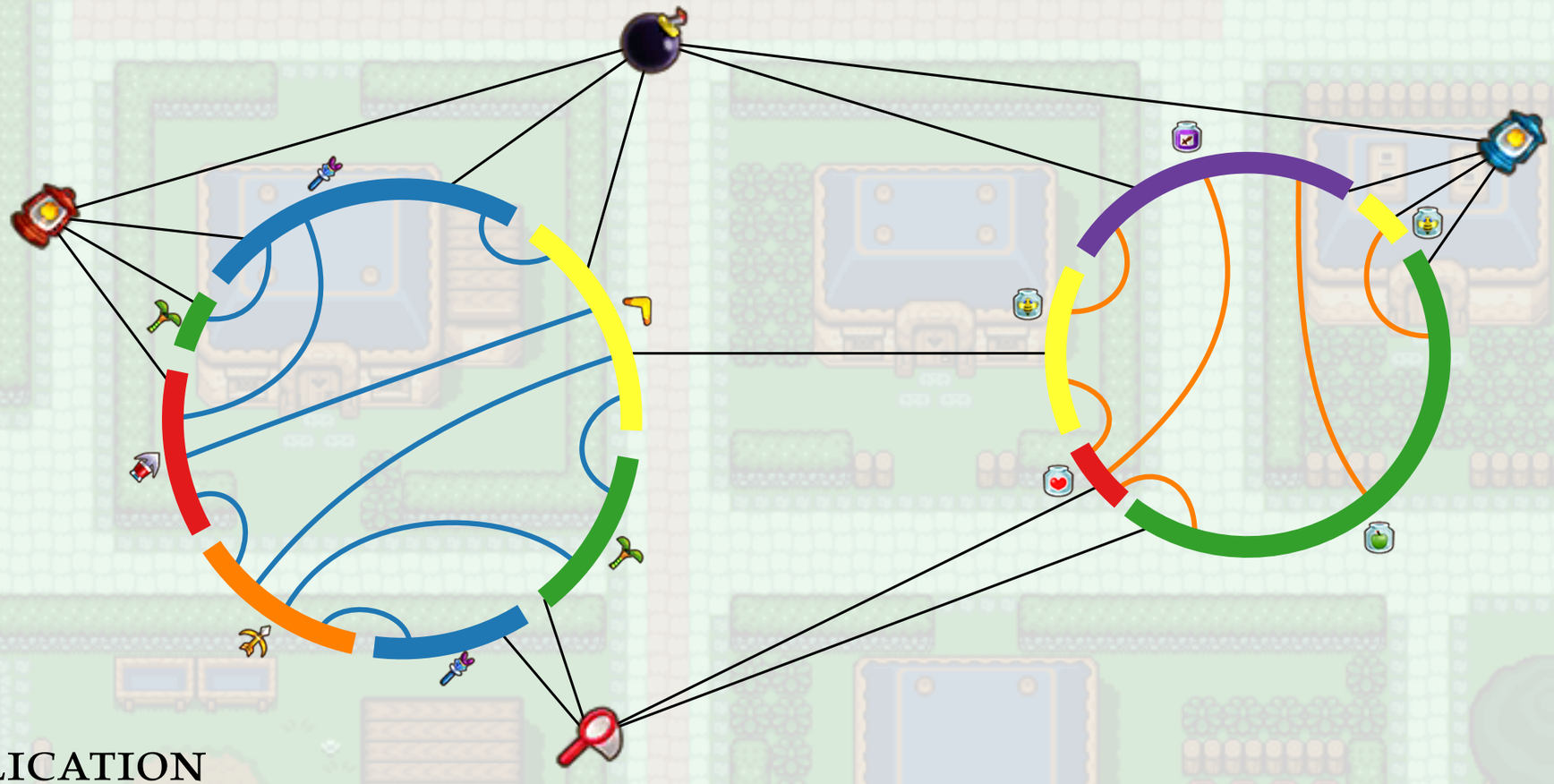


Theorem. Deciding whether there is a crossing-free solution for ChordInsertion is NP-complete.

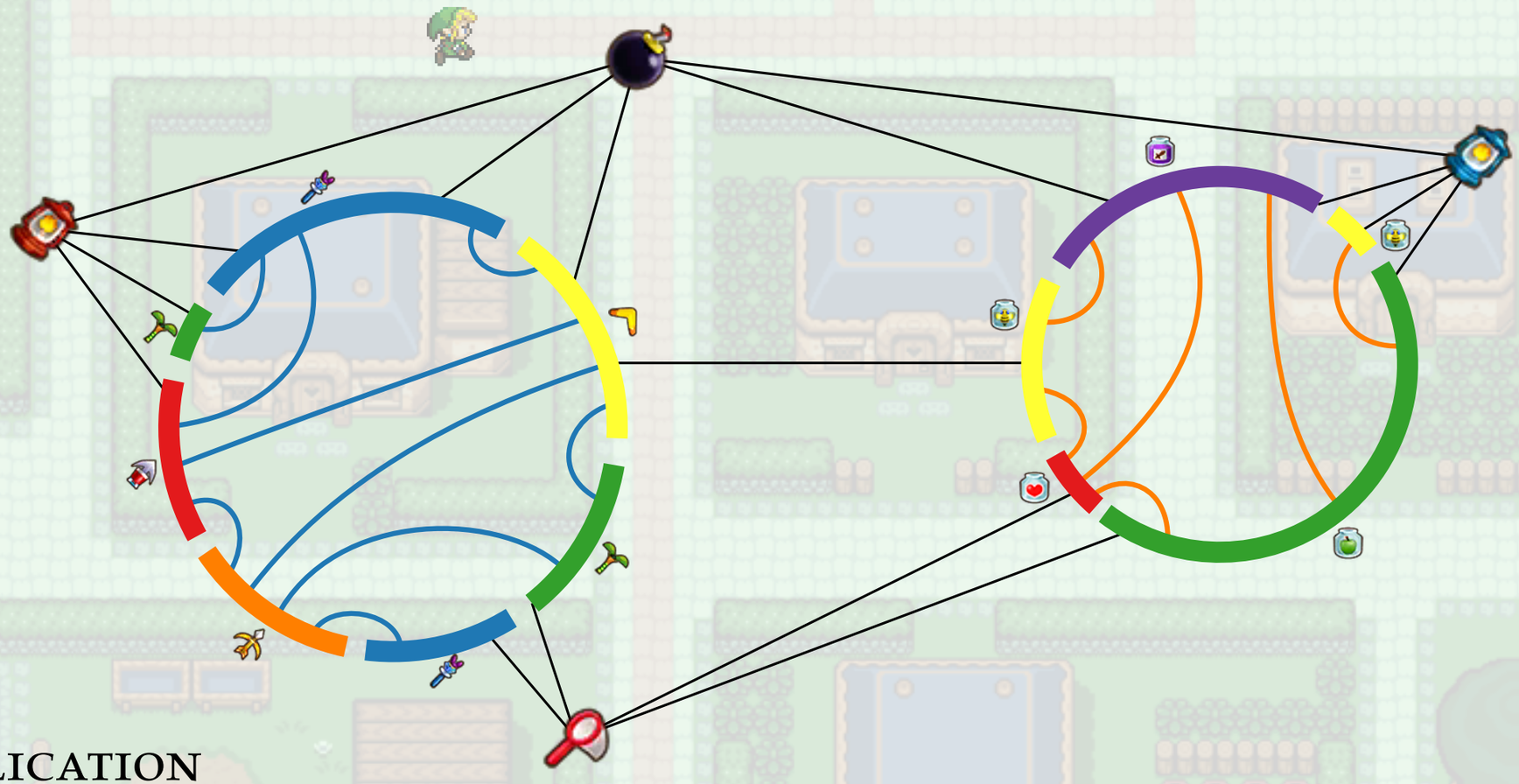


Theorem. Deciding whether there is a crossing-free solution for ChordInsertion is NP-complete.

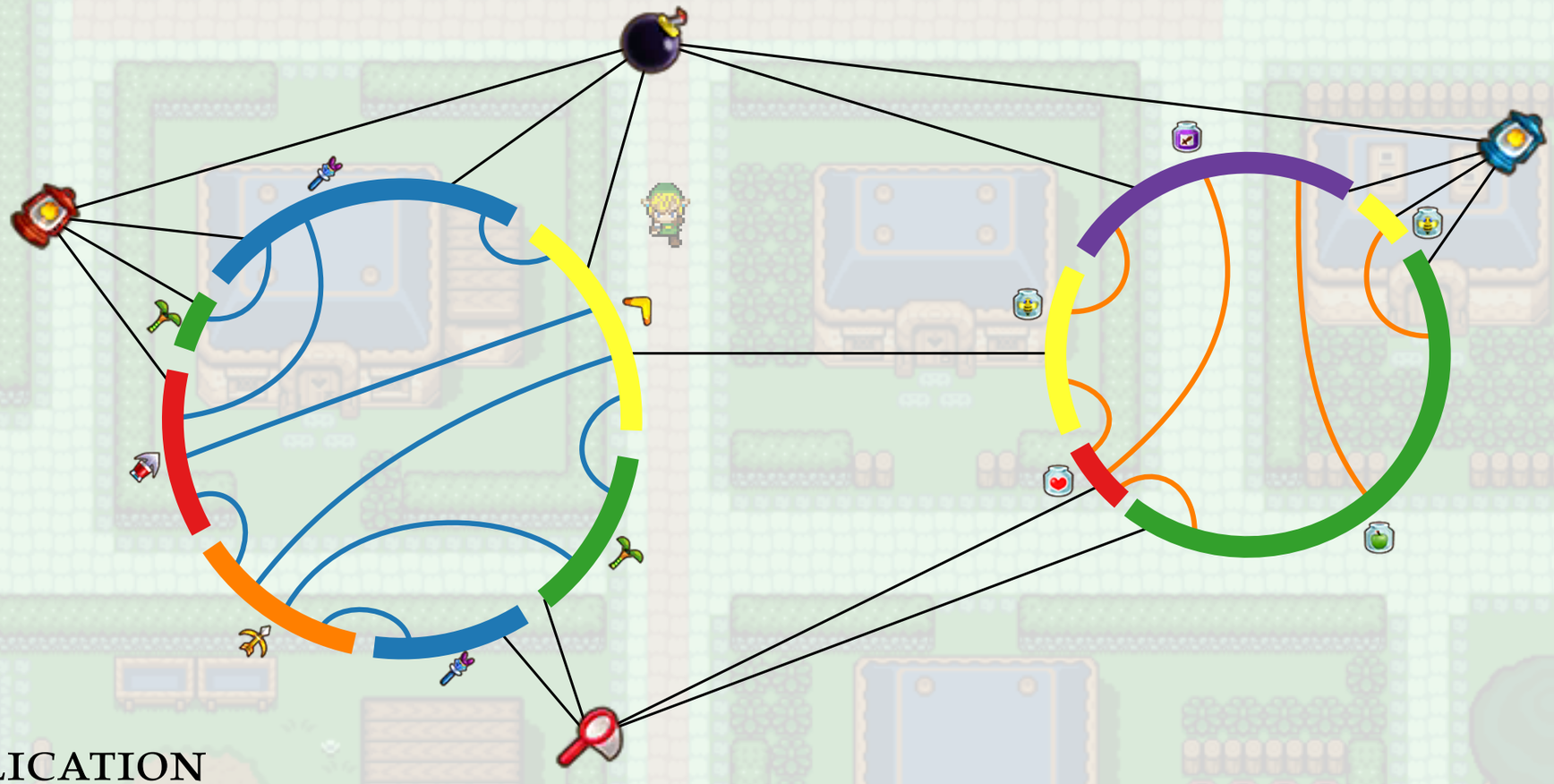
Theorem. Minimizing crossings in ChordInsertion is APX-hard even if there are at most two possible choices for every edge.



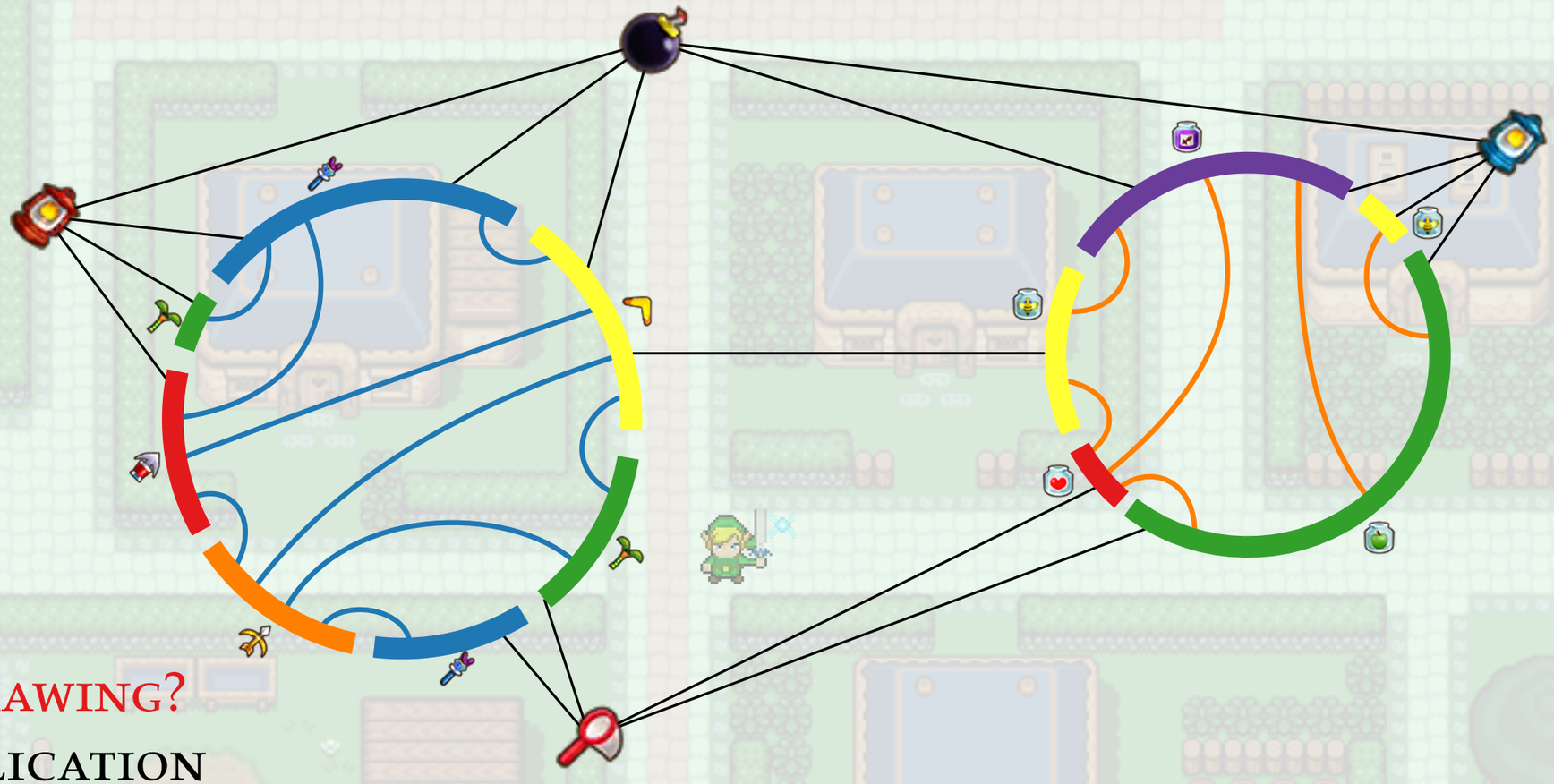
1. NodeReplication		
2. NodePermutation	Can flip nodes with common neighbor Goal: max. #successive pairs with same icon	NP-complete
3. NodeMerging		
4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings	APX-hard



1. NodeReplication		
2. NodePermutation	Can flip nodes with common neighbor Goal: max. #successive pairs with same icon	NP-complete
3. NodeMerging		
4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings / angular resolution	APX-hard



1. NodeReplication		
2. NodePermutation	Can flip nodes with common neighbor Goal: max. #successive pairs with same icon	NP-complete
3. NodeMerging		
4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings / angular resolution / ?	APX-hard



0. InitialDrawing?		
1. NodeReplication		
2. NodePermutation	Can flip nodes with common neighbor Goal: max. #successive pairs with same icon	NP-complete
3. NodeMerging		
4. ChordInsertion	Reintroduce chords in cluster Choose any copies of the adjacent vertices Goal: min. #crossings / angular resolution / ?	APX-hard