

Lecture 21

Level Design

Do We Really Need Level Design?

- Level design makes sense for single player games
- What if our game is **open world**?
 - Each location is a level
 - All that changes is the transition
- What if our game is **multiplayer**?
 - Are the maps always the same?
 - What about game modes?
- What if is a **strategic card game** (e.g. *Magic*)?
 - Are all the cards available at start?
 - How does someone learn how to play?

What is Level Design?

- Layout of **game geography**
 - Location and relationship of challenges
 - Movement of dynamic features (e.g. NPCs)
- Understanding of **player capabilities**
 - Abilities, mechanics available to the player
 - Assumptions of current player skill level
- Layout of **player progression**
 - How the player should move through the game
 - How the player visualizes this progression

Aspects of Game Design

- Games as **Exploration**
 - Focuses on game *geography* and *capabilities*
 - Typically involves heavy storyboarding
- Games as **Education**
 - Train player skill and understanding
 - Focuses primarily on *player capabilities*
- Games as **Storytelling**
 - Focuses on *player progression*
 - Most challenging element of game design

Aspects of Game Design

- Games as **Exploration**
 - Focuses on game *geography* and *capabilities*
 - Typically involves heavy storyboarding
- Games as **Education**
 - Train player skill and understanding
 - Focuses primarily on *player capabilities*
- Games as **Storytelling**
 - Focuses on *player capabilities*
 - Most important element of game design

Not in this Lecture

Aspects of Game Design

- **Games as Exploration**

- Focuses on game *geography* and *capabilities*
- Typically involves heavy storyboarding

- **Games as Education**

- Train player skill and understanding
- Focuses primarily on *player capabilities*

- **Games as Storytelling**

- Focuses on narrative
- Most important element of game design

Not in this Lecture

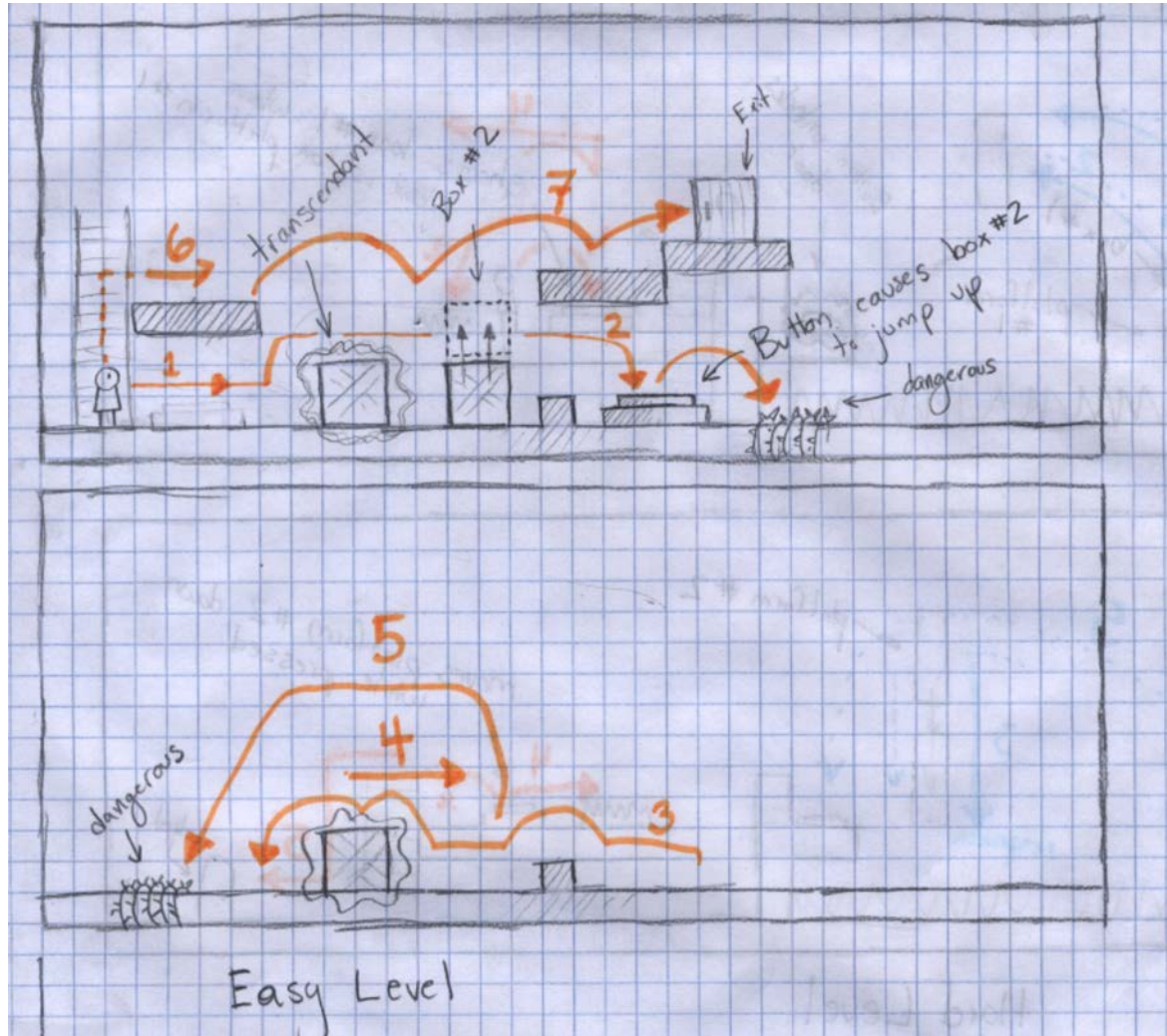
Players Want to Explore the World

- Exploring the **physical space**
 - What happens when I go here?
 - **Example:** Any open world RPG
 - But does not require complex game world
- Exploring the **ludic space**
 - What happens when do this action?
 - Requires deep, complex interactions
 - **Example:** Goofing on Bethesda NPCs

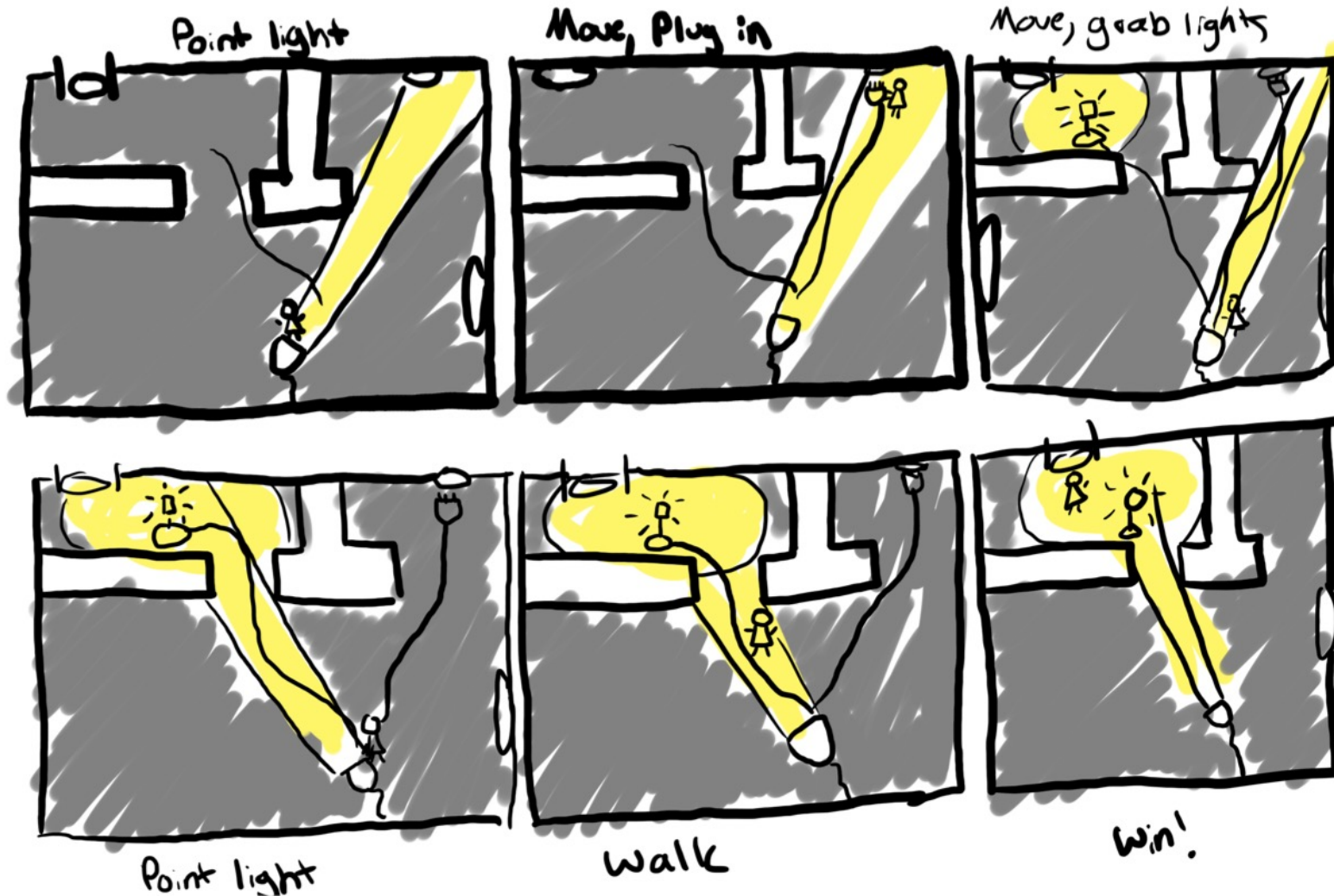
Storyboarding

- Diagrams player action throughout level
 - Different from film storyboarding
 - Focuses on *possibilities*, not direction
- **Single Scene Storyboarding**
 - Shows a single scene tracing player position
 - Works if environment is static/predictable
- **Multi-scene Storyboarding**
 - Shows separate scenes for each major step
 - Important if environment is dynamic/mutable

Example: Single Scene Storyboard



Example: Multi-Scene Storyboard



The Problem With Storyboarding

- You are **not** the player!
 - You storyboard what you *think* player will do
 - Player may do something completely *different!*
- Level design is about **constraining** player
 - You design level to force player to do things
 - Challenges are doors blocking progress
 - Player must use skill to open the door
- Storyboarding **maps** these constraints

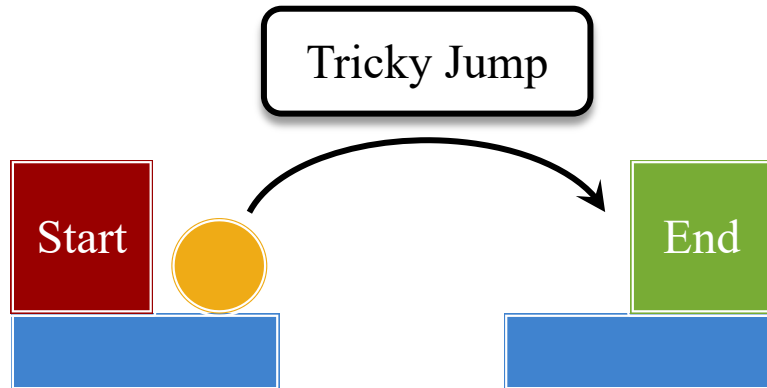
Design Patterns

- Design uses building blocks
 - Mechanic/challenge pairs
 - Start and end location
 - String together to make level
- Key building block features
 - Requires verb/interaction
 - Must be possible to *fail*
 - Difficulty is *tunable*
- **Patterns** are common blocks
 - Appear many times in game
 - Even across multiple games

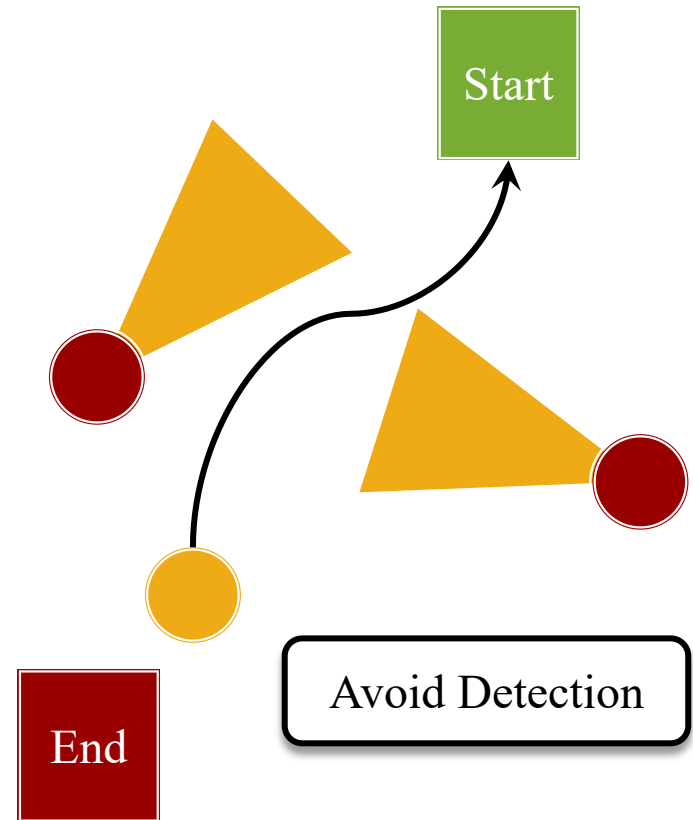


Design Pattern Examples

Platformer

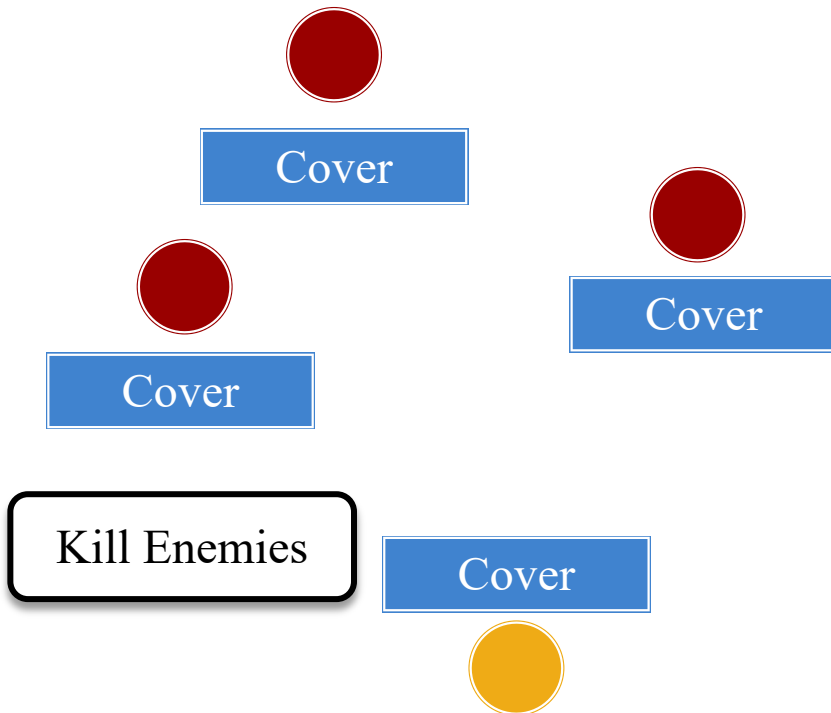


Stealth Game

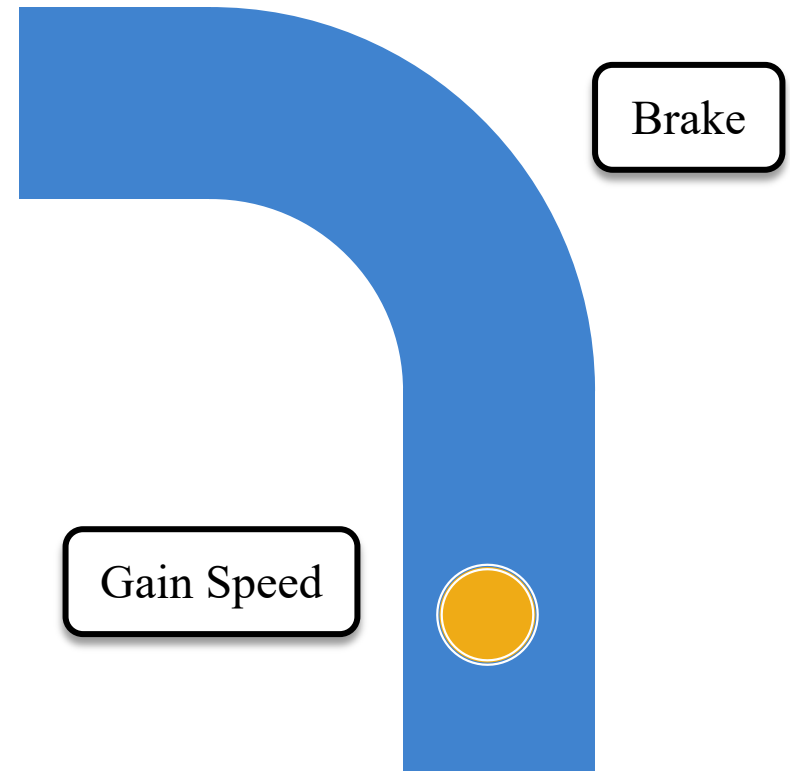


Design Pattern Examples

Shooter/Action Game



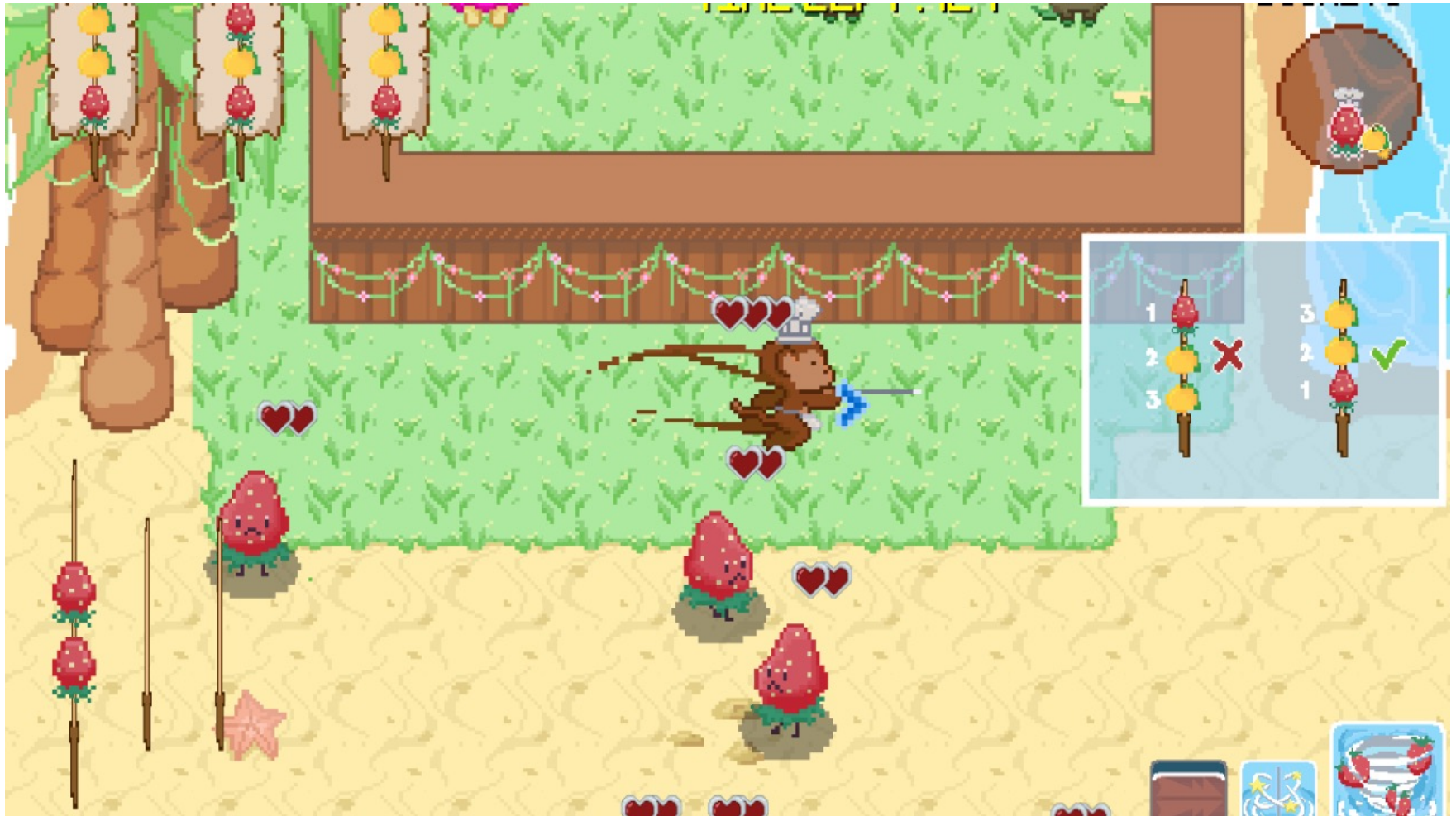
Racing Game



Pitfalls To Avoid

- It must be possible to **fail** a pattern
 - No fail condition means no **challenge**
 - But does not have to be a game-over fail
- A pattern **cannot** be something **player does**
 - It is a **constraint** on the player
 - Players actions should be a reaction to the constraint
- **Patterns** != **gameplay elements**
 - **E.g.** An enemy by itself is not a challenge
 - Elements must be partnered with (mini)**goal**

Case Study: Munchkey



Case Study: Munchkey



Munchkey: Bad Patterns



- These stacked enemies will scuttle around the map. Each stab will put one berry on the player's skewer, so they must be careful to be accurate when challenging this foe.



Munchkey: Bad Patterns

Basic Patterns - Blueberry Stacks!

This is just
an enemy

- These stacked enemies will scuttle around the map. Each stab will put one berry on the player's skewer, so they must be careful to be accurate when challenging this foe.

Why?



Munchkey: Bad Patterns

Basic Patterns - Blueberry Stacks!

- The
- Eac
- so t
- cha

Combine with order with
berries in the middle

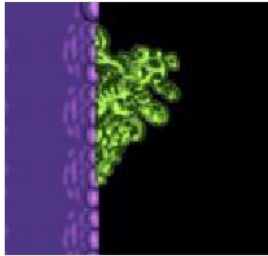
map.
ewer,



Case Study: *Sisyphus*



Sisyphus: Bad Patterns



Moving Hazard on Wall

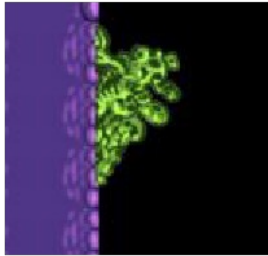


Ground Enemy with Area of Effect Attack



Flying Enemy that Shoots Projectiles

Sisyphus: Bad Patterns



Moving Hazard on Wall

These are
just enemies



Flying Enemy that Shoots
Projectiles

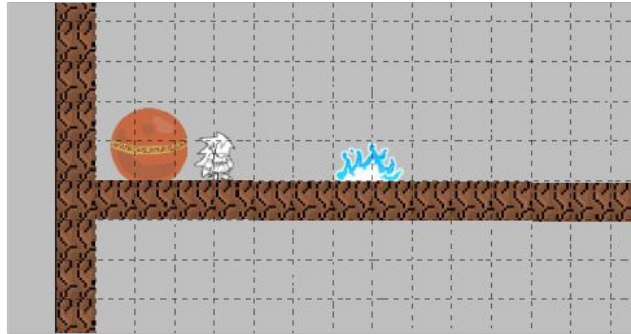
Killing enemies
is not the goal



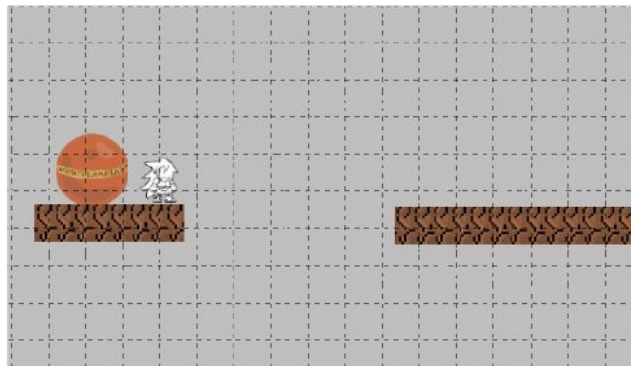
Ground Enemy with Area of
Effect Attack

What constraint
is there here?

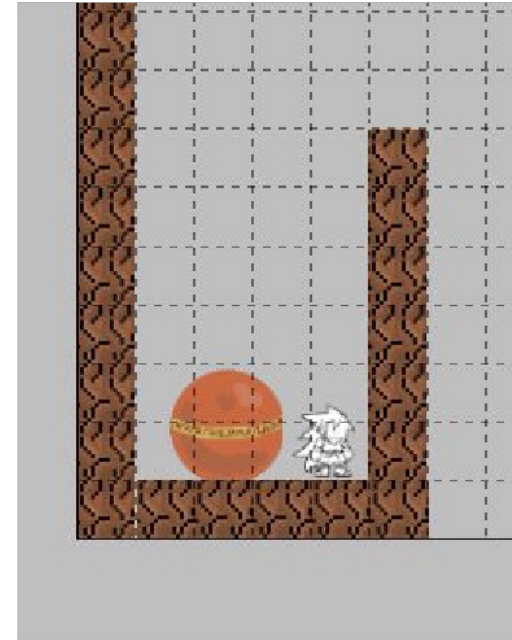
Sisyphus: Good Patterns



Static Hazard in the Way

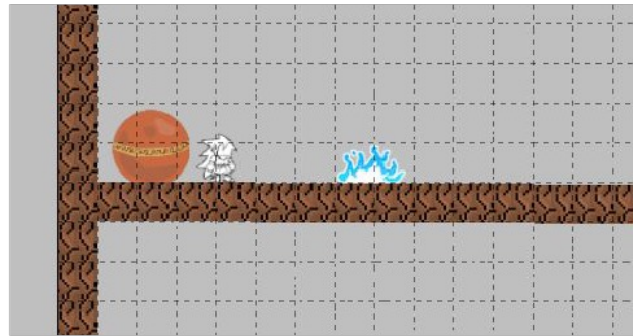


Large Gap

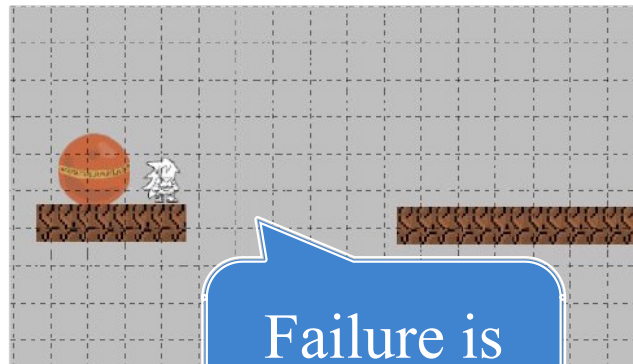


Deep Pit/Tall Wall

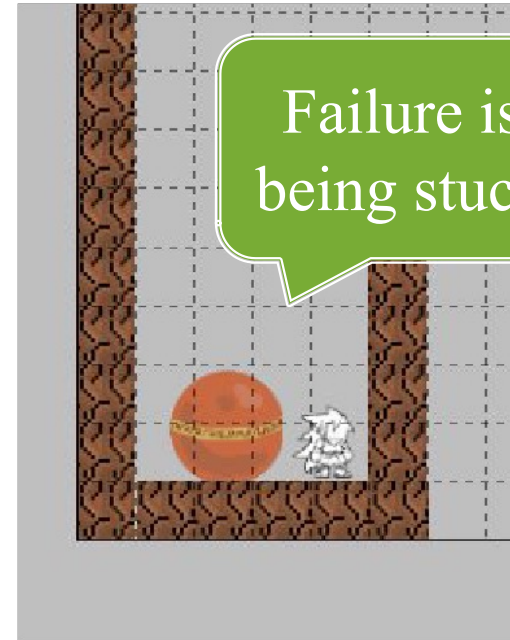
Sisyphus: Good Patterns



Static Hazard in the Way



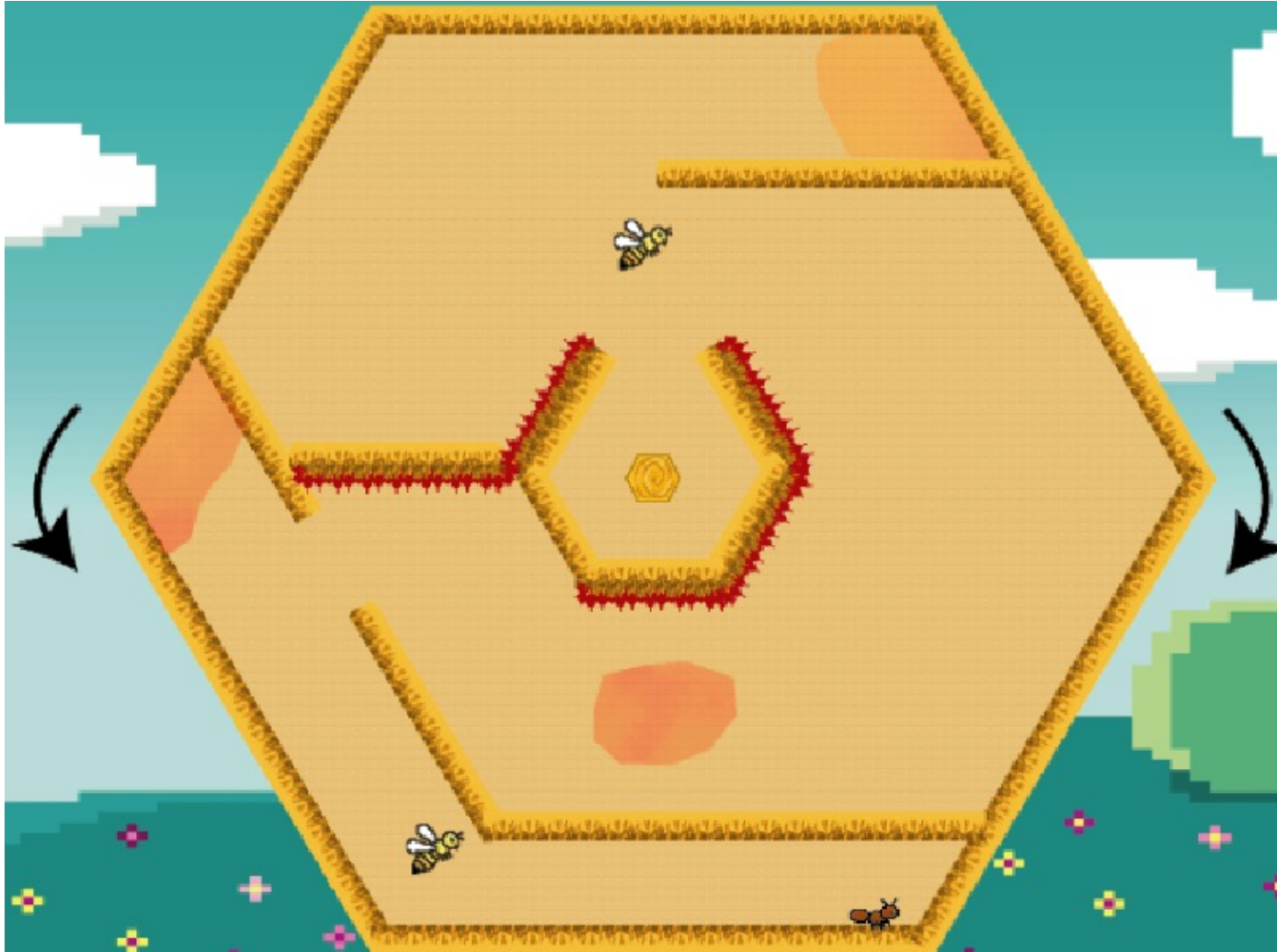
Failure is
falling



Failure is
being stuck

Deep Pit/Tall Wall

Case Study: *Honey Heist*

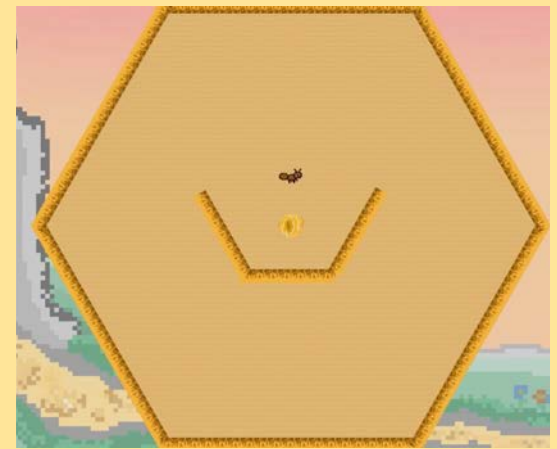


Honey Heist: (Semi)Bad Patterns

Basic Pattern 1: Rotation to Get to Objective

Challenge: Goal or platform is currently unreachable.

Solution: Rotate the world in order to drop down to the goal.



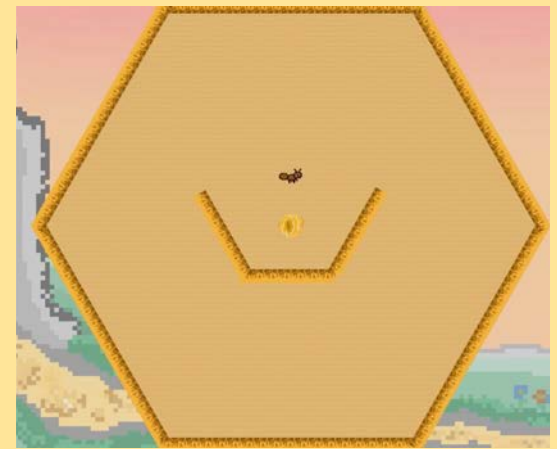
Honey Heist: (Semi)Bad Patterns

Basic Pattern 1: Rotation to Get to Objective

Challenge: Goal or platform is
Solution: Rotate the world in o

Worded as what
the player does

goal.



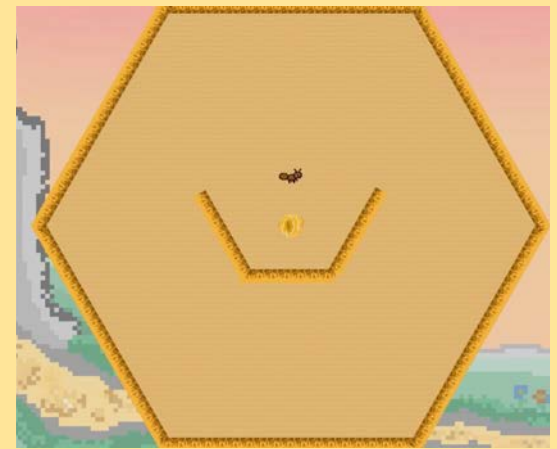
Rewording the Pattern

Basic Pattern 1:

Objective Has No Floor

Challenge: Goal or platform is currently unreachable.

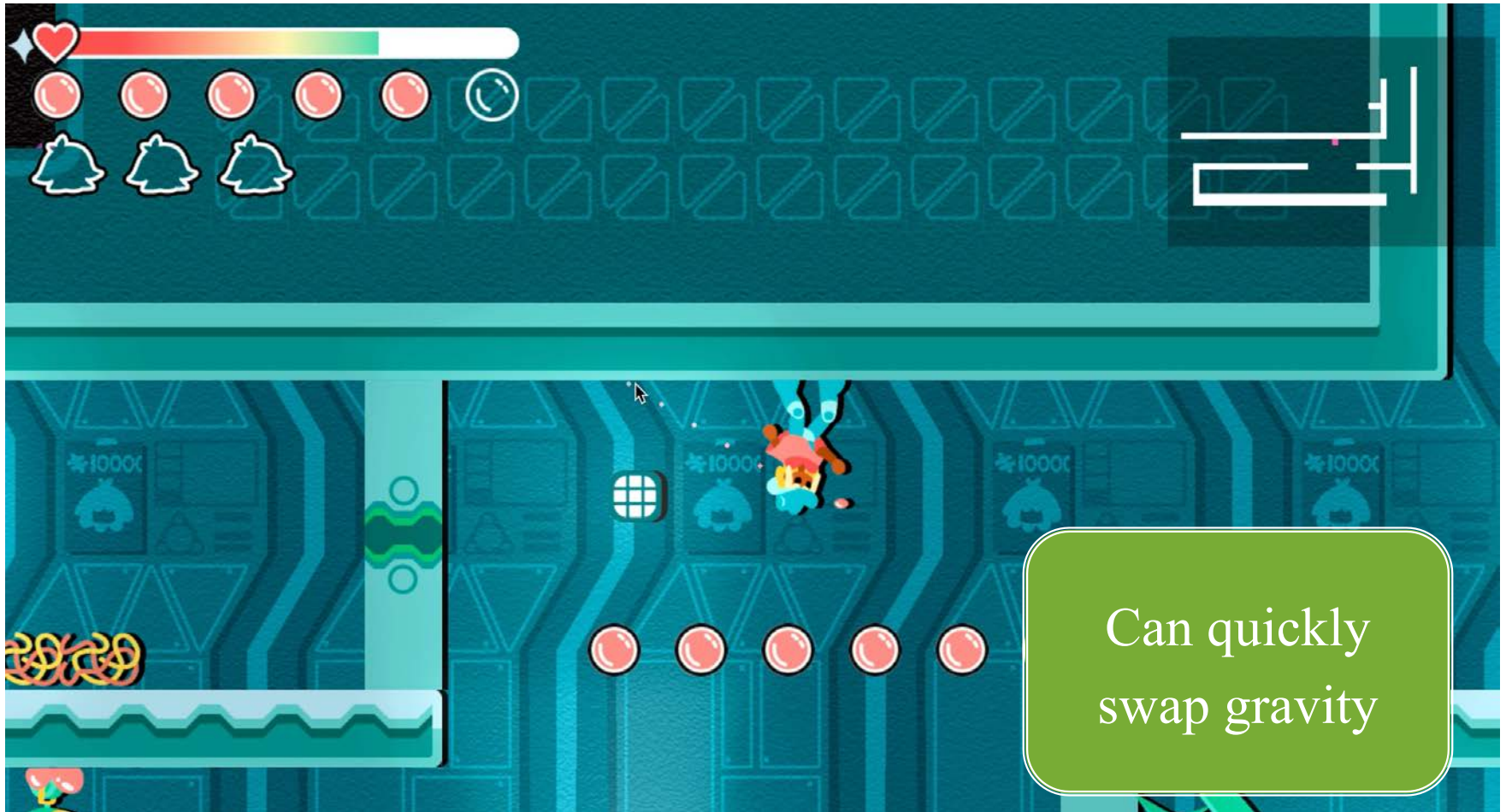
Solution: Rotate the world in order to drop down to the goal.



Case Study: *Bubblegum Bandit*



Case Study: *Bubblegum Bandit*



Bubblegum Bandit Patterns

Enemy in Tight Hallway

Safe passage through a hallway is being blocked by an enemy.



Bubblegum Bandit Patterns

Enemy in Tight Hallway

Good

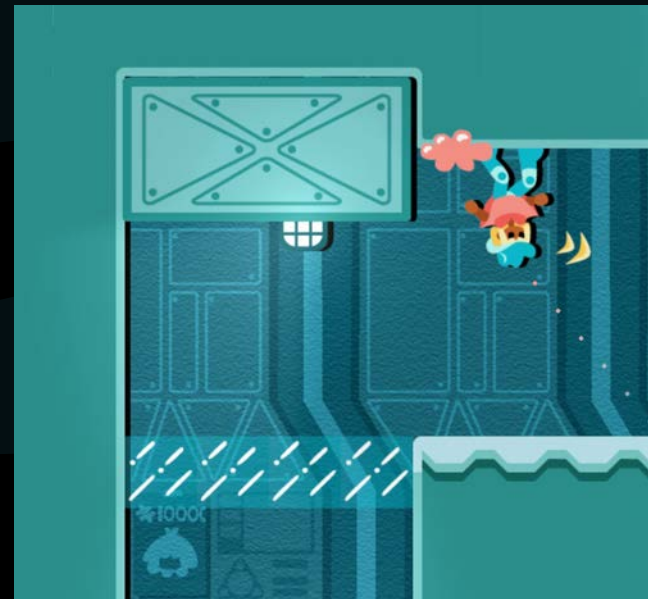
Safe passage through a hallway is being blocked by an enemy.



Bubblegum Bandit Patterns

Breakable Wall Blocking Path

Some areas of the ship are closed off by breakable glass walls, which can only be passed by dropping a heavy cargo block on the wall to break it.



Bubblegum Bandit Patterns

Breakable Wall Blocking Path

Excellent!

Some areas of the ship are closed off by breakable glass walls, which can only be passed by dropping a heavy cargo block on the wall to break it.



Bubblegum Bandit Patterns

Hazard Tiles with Vertical Gap

Broken wires are scattered across the ships, which can electrocute and cause damage to the player on contact.

Hazard tiles in a vertical gap configuration make it so neither the floor or ceiling is safe to walk on.



Bubblegum Bandit Patterns

Hazard Tiles with Vertical Gap

Excellent!

Broken wires are scattered across the ships, which can electrocute and cause damage to the player on contact.

Hazard tiles in a vertical gap configuration make it so neither the floor or ceiling is safe to walk on.

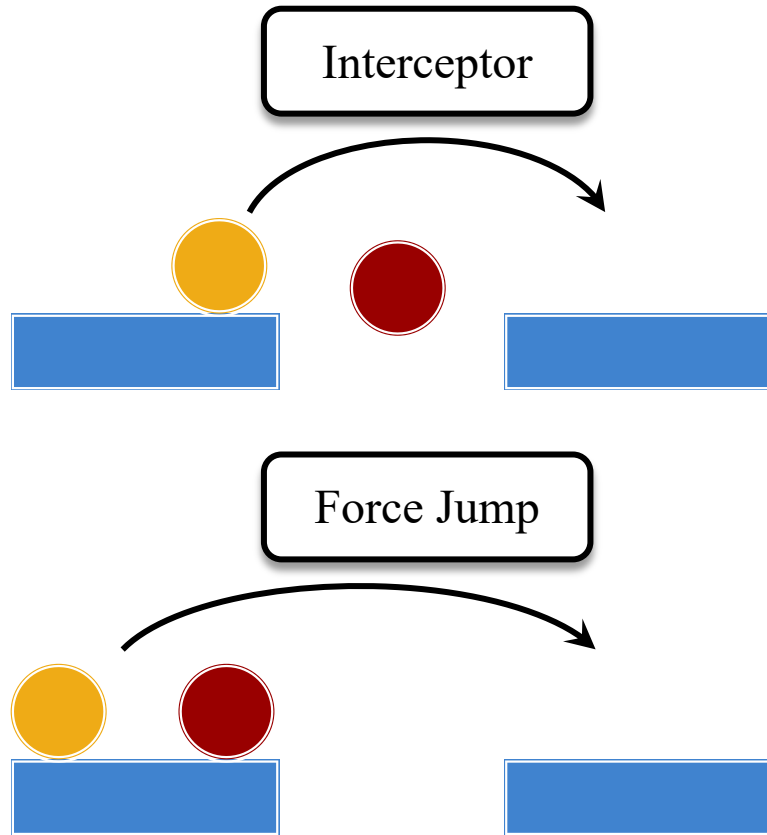


Advanced Patterns

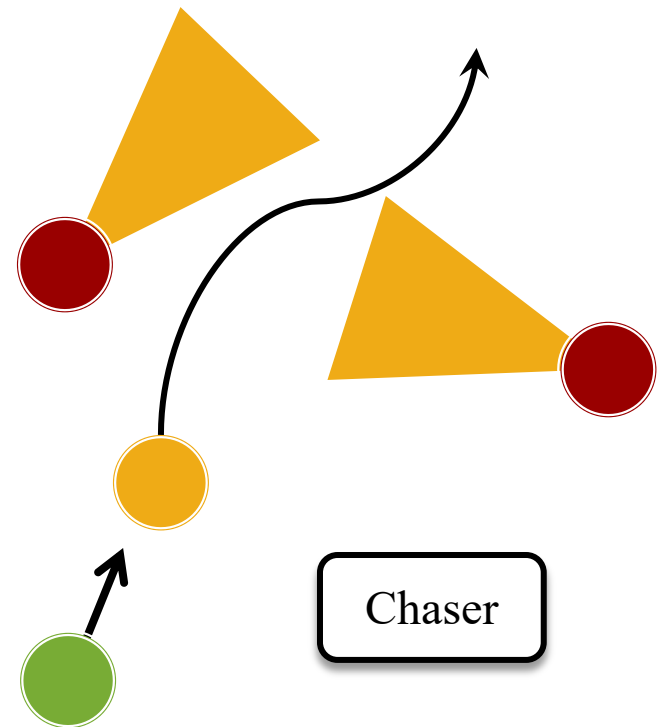
- Piecewise design creates a very linear feel
 - **Pattern A** followed by **Pattern B** followed by...
 - Player is explicitly aware of building blocks
- **Evolutionary patterns** add new variations
 - Base pattern plus additional game elements
 - Makes original pattern much more difficult
- **Composite patterns** creates combinations
 - Two patterns combined in the same space
 - Challenges play to handle them both

Advanced Patterns

Platformer

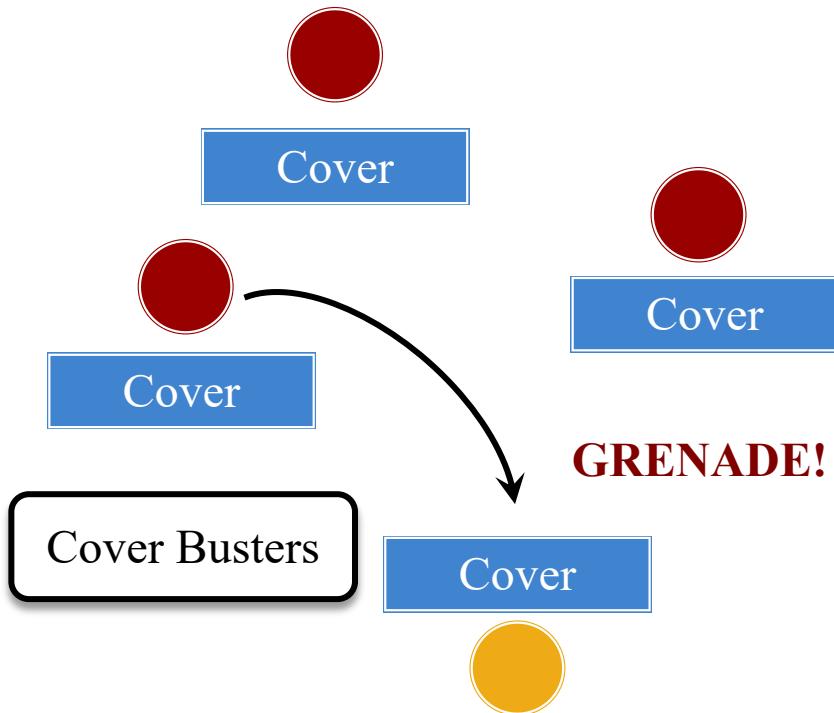


Stealth Game

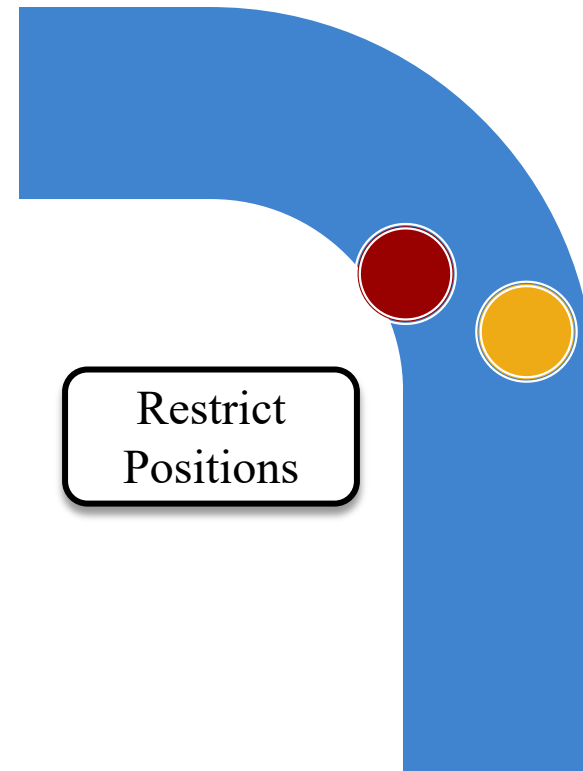


Advanced Patterns

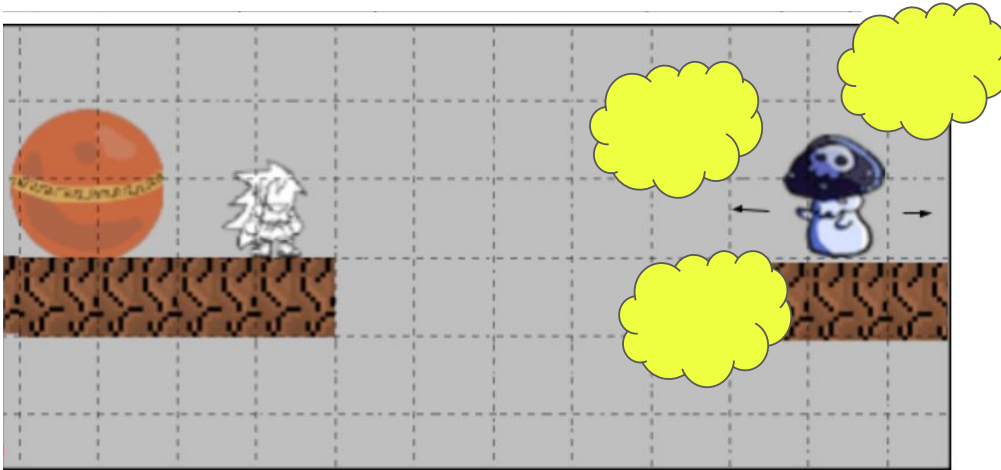
Shooter/Action Game



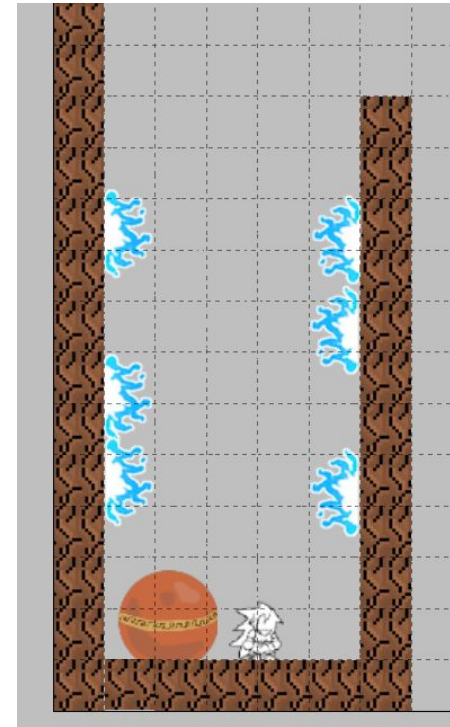
Racing Game



Sisyphus: Evolutionary



Gap + Enemy

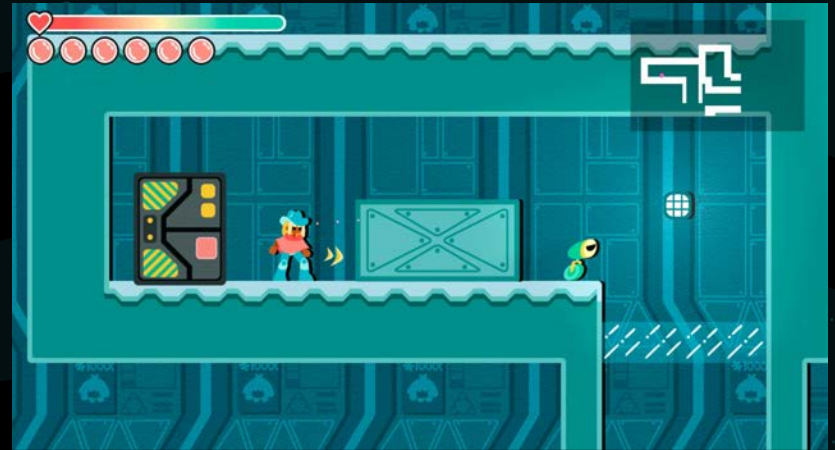


Pit + Obstacles

Bubblegum Bandit: Composite

Breakable Wall & Enemy in Tight Hallway

The movement of the cargo block to the glass is complicated by the enemy who prevents the block from moving while it is in the way.



Putting it All Together

- Once you have patterns, time for level design
 - Arrange patterns together about the map
 - Often link them together linearly
 - But some challenges may be *optional*
- Key decision: **pacing**
 - What is the time between challenges?
 - Can the player rest between challenges?
- Depends on the type of game that you want
 - **Action:** quick paced, limited down time
 - **Puzzle:** contemplative with frequent down time

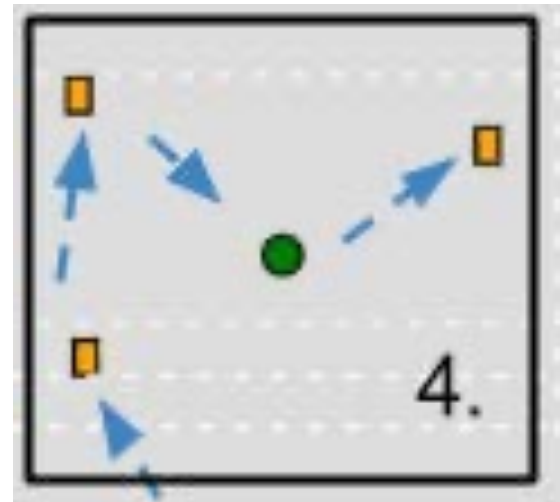
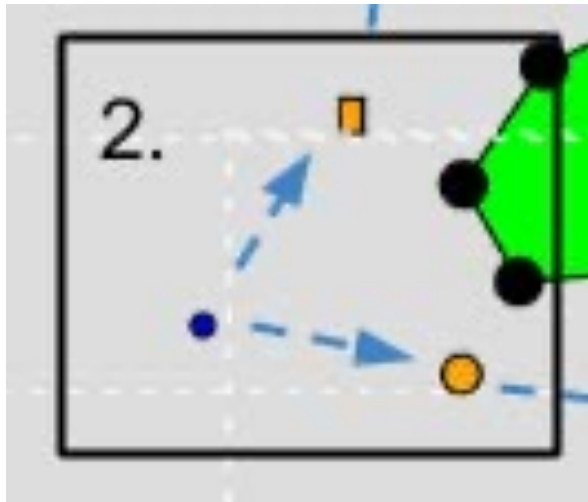
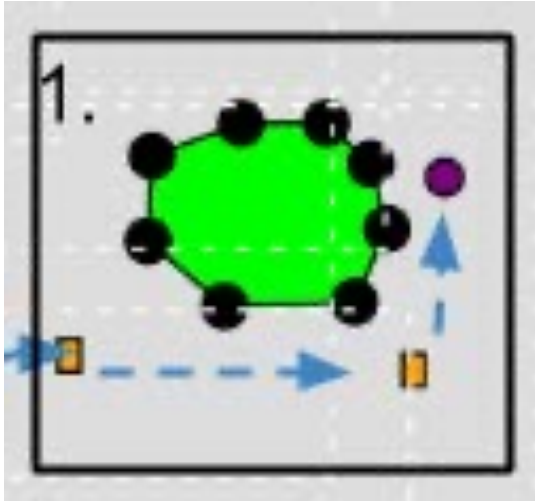
Putting it All Together

- Once you have patterns, time for level design
 - Arrange patterns together about the map
 - Often link them together linearly
 - But some challenges may be *optional*
- Key decision: **spacing**
 - What is the time between challenges?
 - Can the player rest between challenges?
- Depends on the type of game that you want
 - **But all games have some rest time**
 -

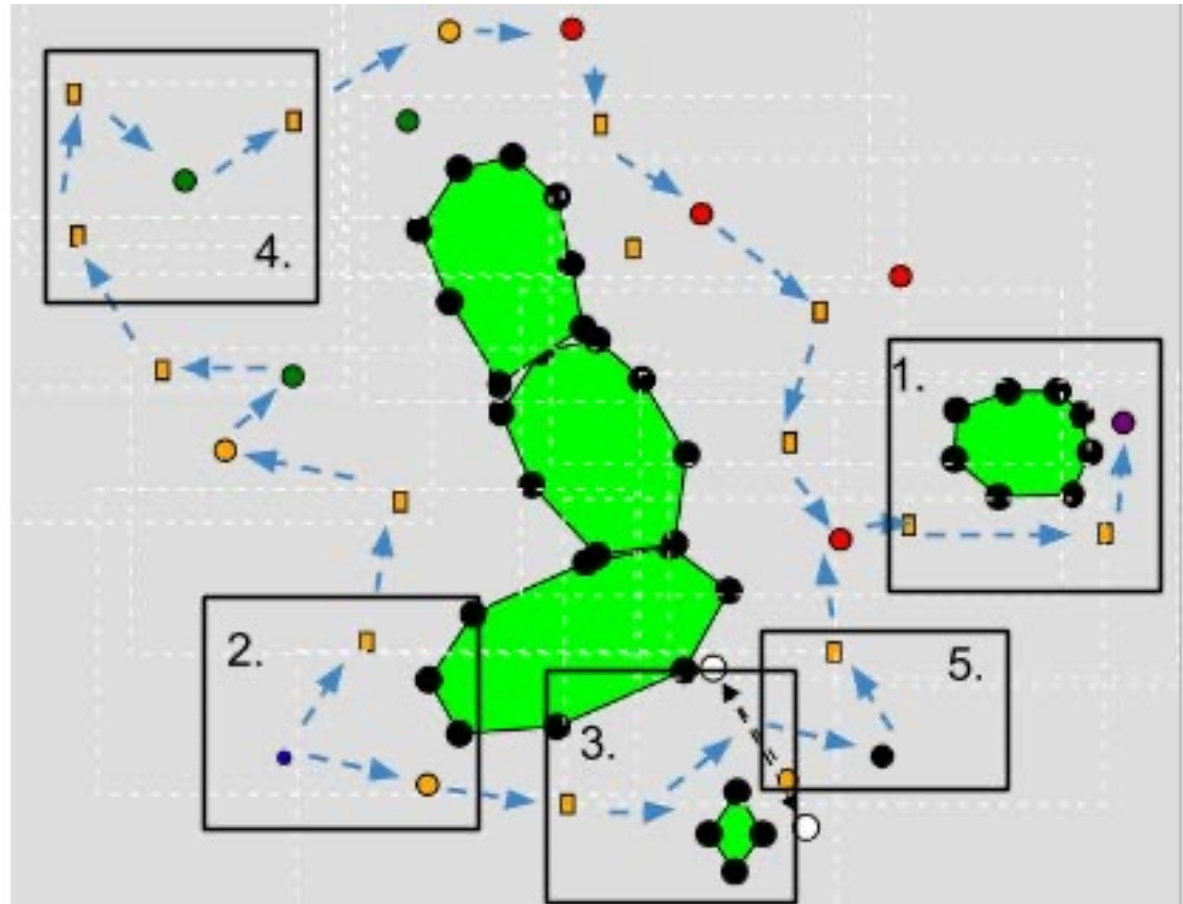
Case Study: *Dash*



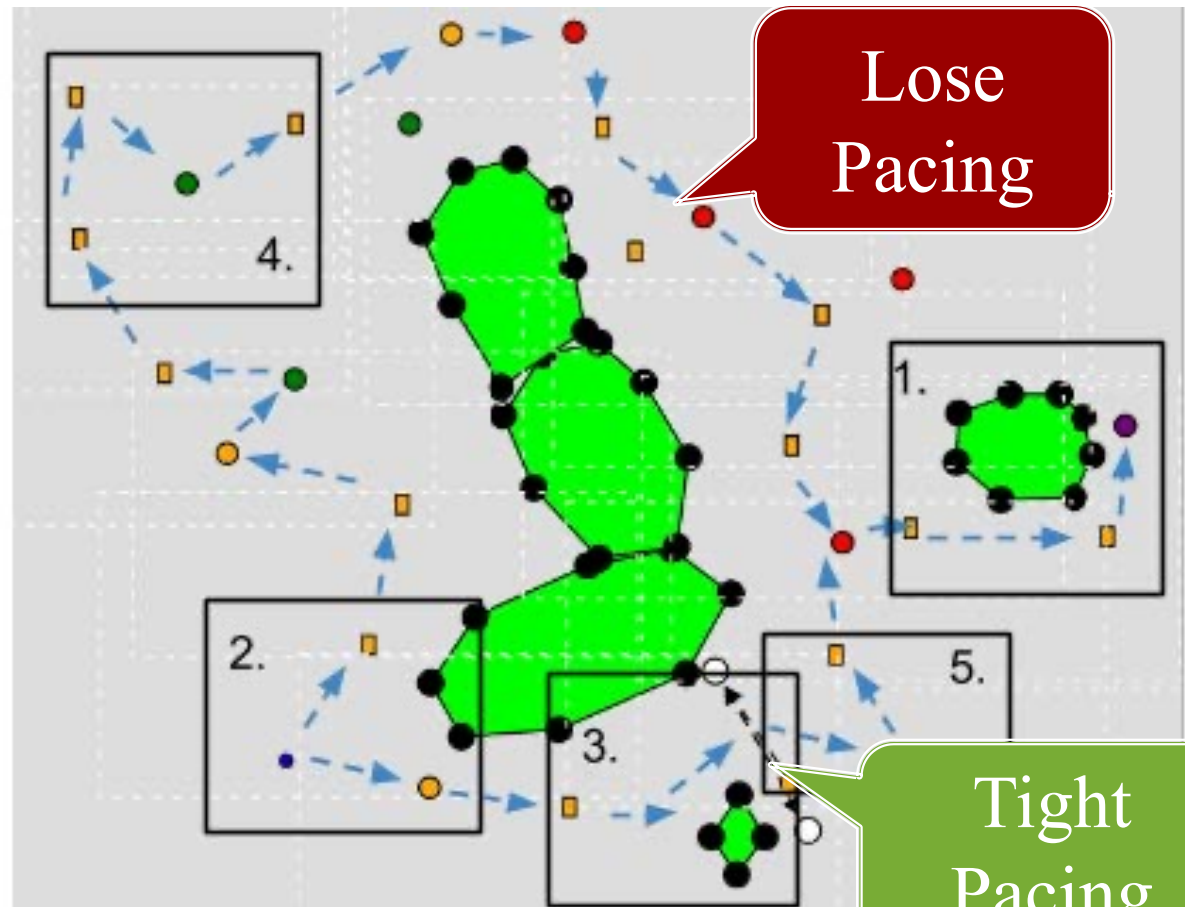
Dash: Basic Design Patterns



Dash: Pacing



Dash: Pacing



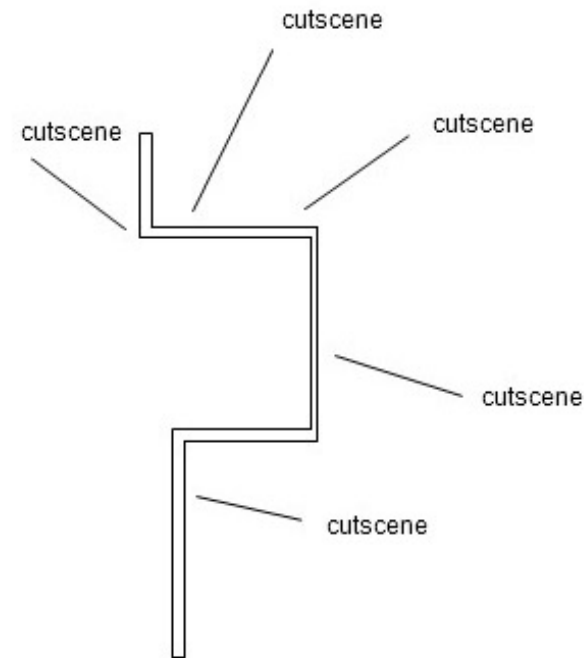
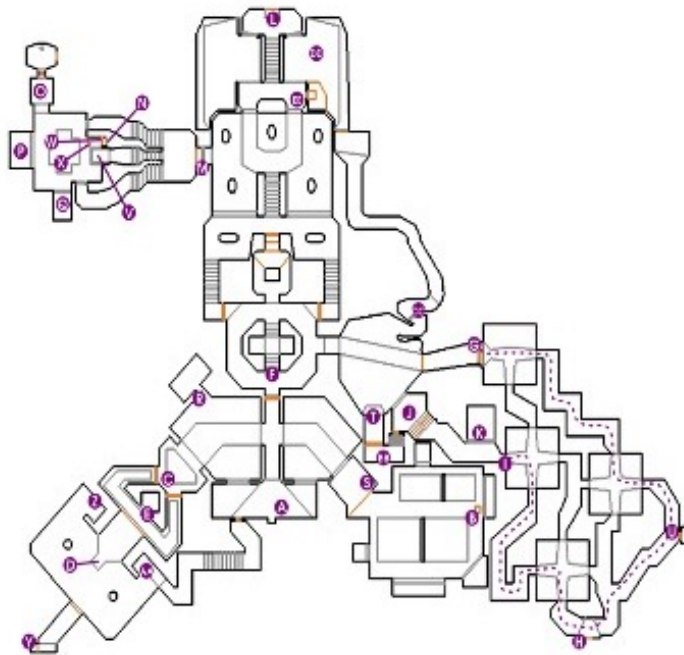
Is Linearity a Problem?

[Image attribution unknown]

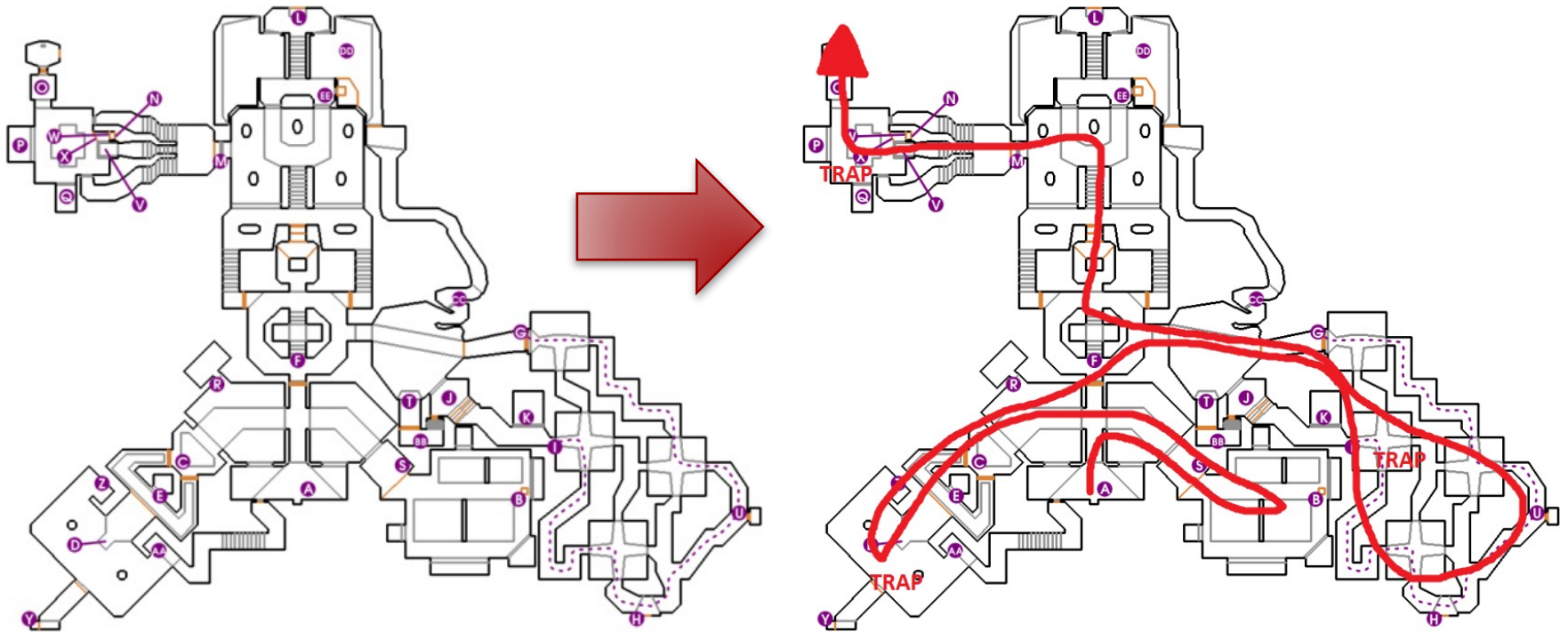
FPS map design

1993

2010

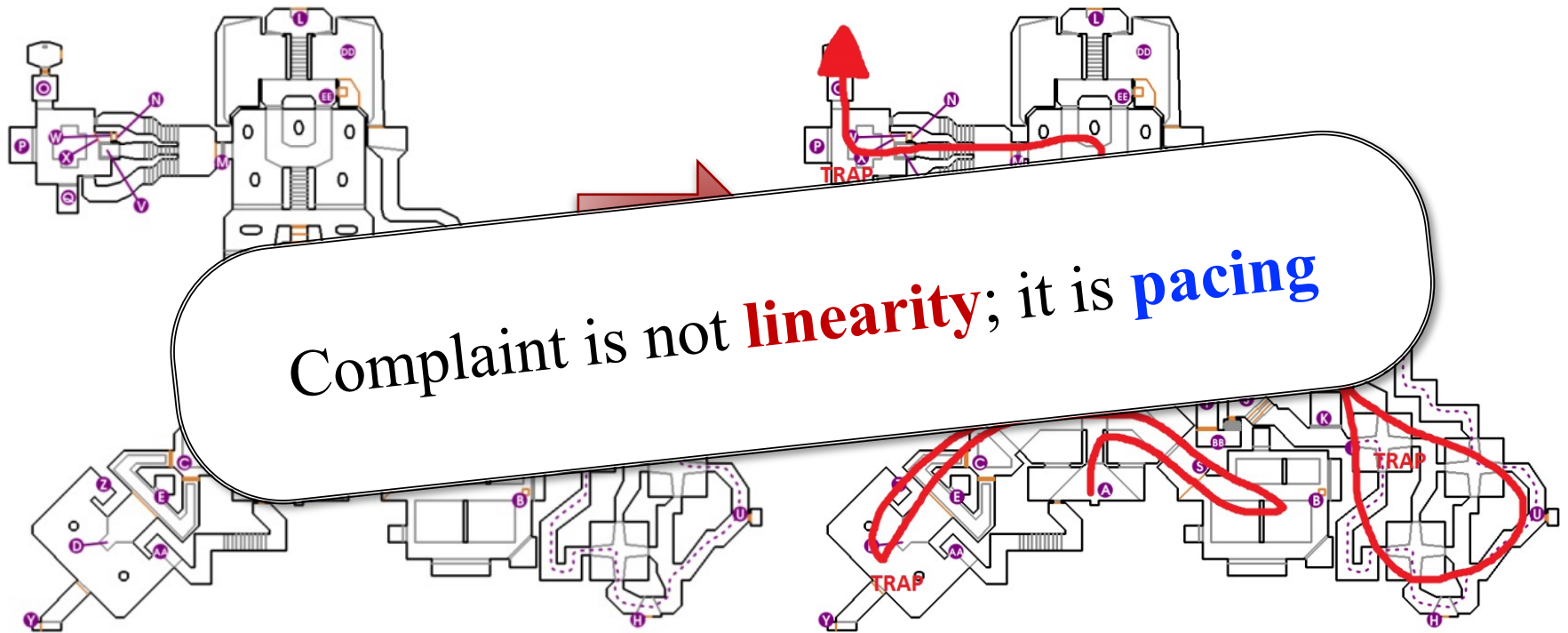


But Actually...



[refugeinaudacity.wordpress.com]

But Actually...



[refugeinaudacity.wordpress.com]

Aspects of Game Design

- Games as **Exploration**
 - Focuses on game *geography* and *capabilities*
 - Typically involves heavy storyboarding
- **Games as Education**
 - Train player skill and understanding
 - Focuses primarily on *player capabilities*
- Games as **Storytelling**
 - Focuses on *player progression*
 - Most challenging element of game design

Learning How to Play

- Mechanics are (often) new and unfamiliar
 - Players have to learn how to interact with them
 - **Aside:** why innovation is not always popular
- Players could learn by reading the *manual*
 - This is boring! Let me play already
- **Tutorial levels** allow the player to...
 - Get started playing immediately
 - Learn the mechanics while playing

Classic Approach: Restrict the Player

- Start with your **gameplay specification**
 - Remove all but the barest mechanics
 - Remove verbs by disabling controls
 - Remove interactions by omitting "board elements"
- Levels add new mechanics back one at a time
 - **Example:** Platformer with a "no-jump" level
- Do not need to add a new mechanic each level
 - "Deep" mechanics allow many levels per mechanic
 - This can influence game geography (e.g. worlds)

Example: Starcraft Campaign



Explicit Restrictions

- Mechanics are unavailable for current level
 - Controls for actions are explicitly disabled
 - Interactions disabled, even if elements present
- **Motivation:** Prevents player confusion
 - Do not waste time on useless mechanics
 - Key in the casual and young audience
- **Examples:** Many AAA commercial games
 - *Starcraft* single-player campaign
 - *Portal* (integrated into story)

Implicit Restrictions

- Mechanics are always available, but not needed
 - Challenges designed for an explicit mechanic
 - Other mechanics may succeed, but they are harder
 - Level has hints to guide player to right mechanic
- **Motivation:** Allow replay in tutorial levels
 - Players go back and try optional approaches
 - Achievements are structured to encourage this
- **Example:** Many amateur Flash games
 - *My First Quantum Translocator*

The Tyranny of Choice

- Too much choice can make us unhappy
 - We are often paralyzed by what to do
 - Studied by Myers & Lane; popularized by Barry Schwartz
- But games are about **meaningful choice**
 - Problem is when choices are too similar
 - Good choices must be *significantly* different
 - **Example:** Dagger adds +1 bonus to a stat of 102
- Players use rough heuristics for making choices
 - Pattern match current situation to determine action

The Tyranny of Choice

- Too much choice can make us unhappy
 - We are often paralyzed by what to do
 - Studied by Myers & Lane; popularized by Barry Schwartz
- But choice is also good
 - **Limiting choice helps train player**
 - Good choices are made
 - **Example:** Dagger adds +1 bonus to a stat of 102
- Players use rough heuristics for making choices
 - Pattern match current situation to determine action

Portal 2 Mechanics



Mechanics

● Introduction

● Variation

Grab

Grab with Portal Device

Grab in flight

Portals

Single Portal Device

Dual Portal Device

Weighted Storage Cube

Multiple cubes

Heavy Duty Super Colliding Super Button

Multiple buttons

Emancipation Grill

Timed switches

Switches

Panels

Momentum

Toxic Water

Thermal Discouragement Beam

Multiple beams

Discouragement Redirection Cube

Unstationary Scaffold or Victory lift

New Mechanics

Recombination

Kill inside chamb

Reinforcement

How long to “dwell” on mechanic before a new one?

Actions:

A = jump

B = dash

A B

vs.

A A A B

Recombination

How often to combine with other mechanics

Actions:

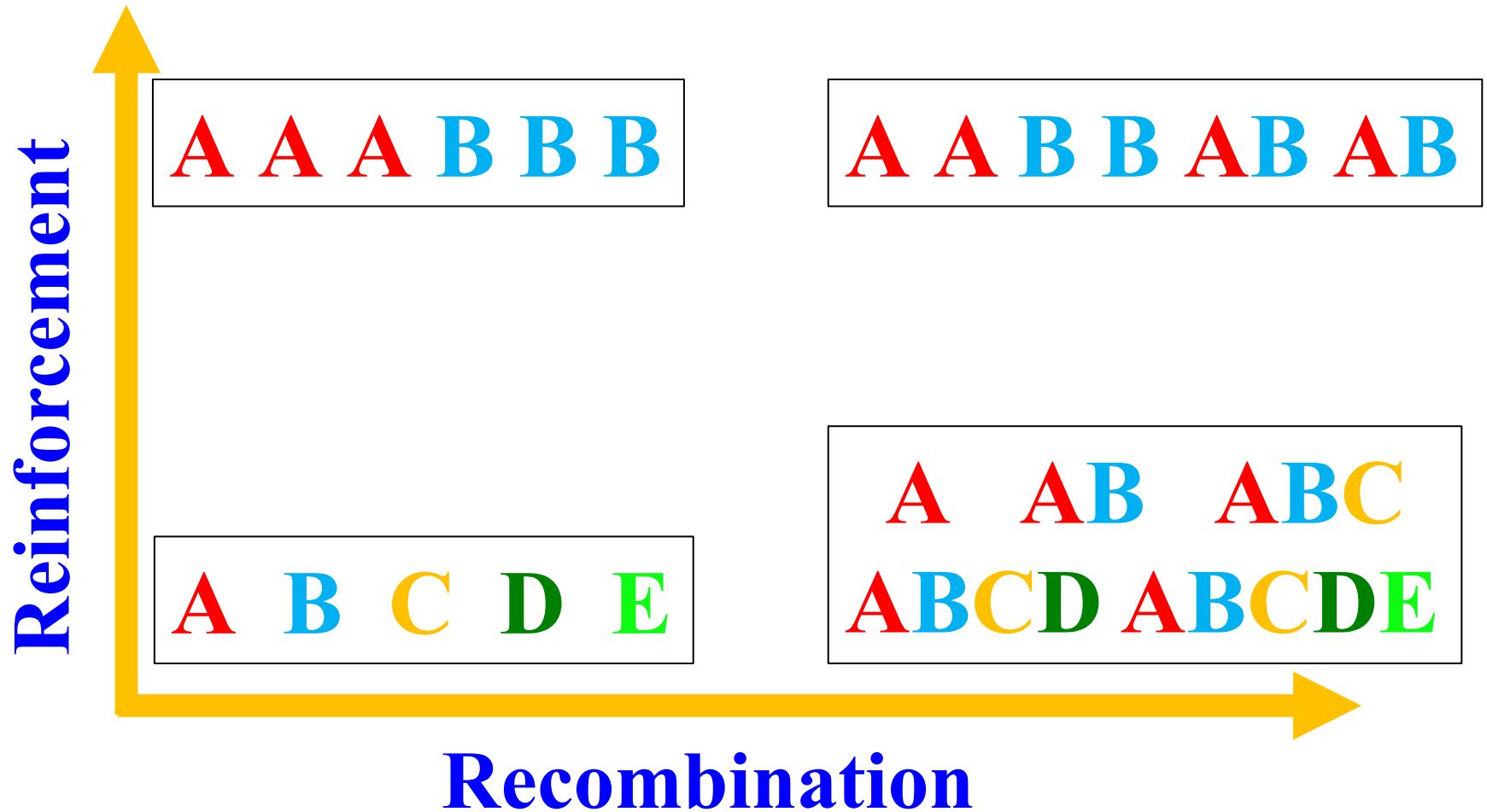
A = jump

B = dash

C = shoot fireball

A B C vs. **A AB ABC**

Reinforcement vs. Recombination



Robot Unicorn Attack



Robot Unicorn Attack Progression

Mechanics:

A = jump

B = dash

A A A B A A B

High reinforcement, low recombination

Hello World!



COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 7
PAR TIME: 45
SPEED TIME: 12

ALPHA

COINS 🟡: 1/6
STARS ⭐: 0
POINTS: 0

Hello World!

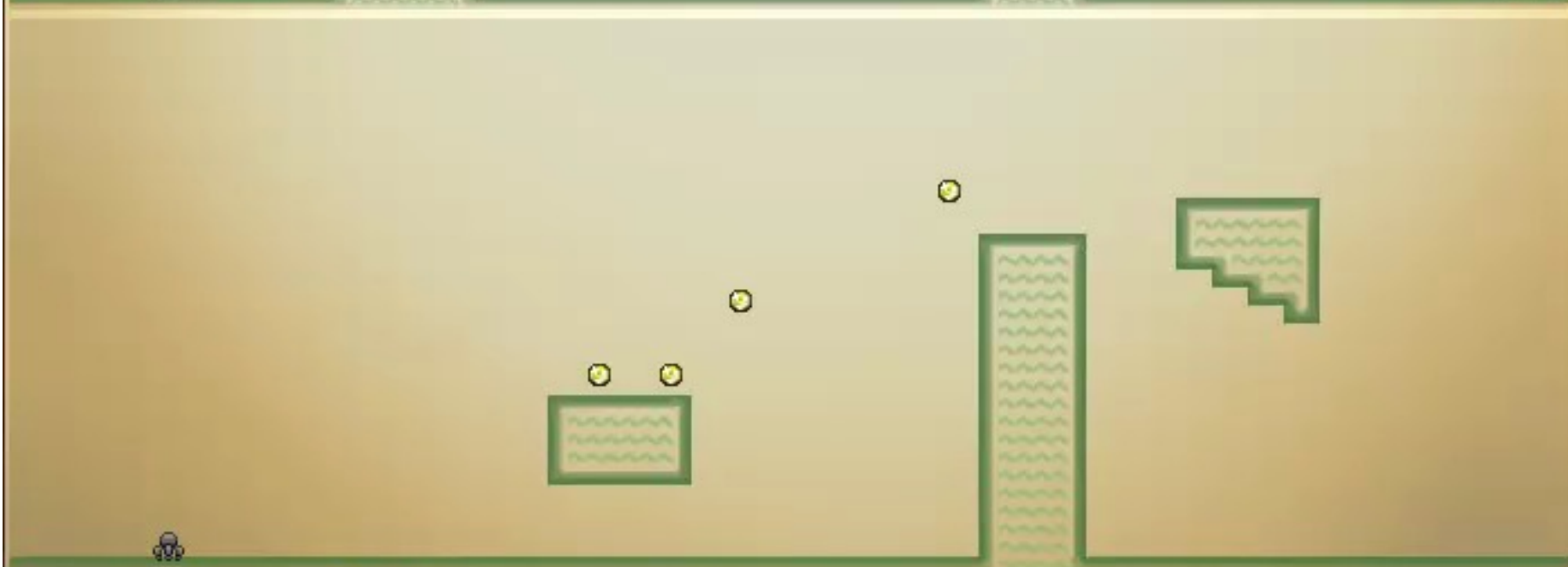
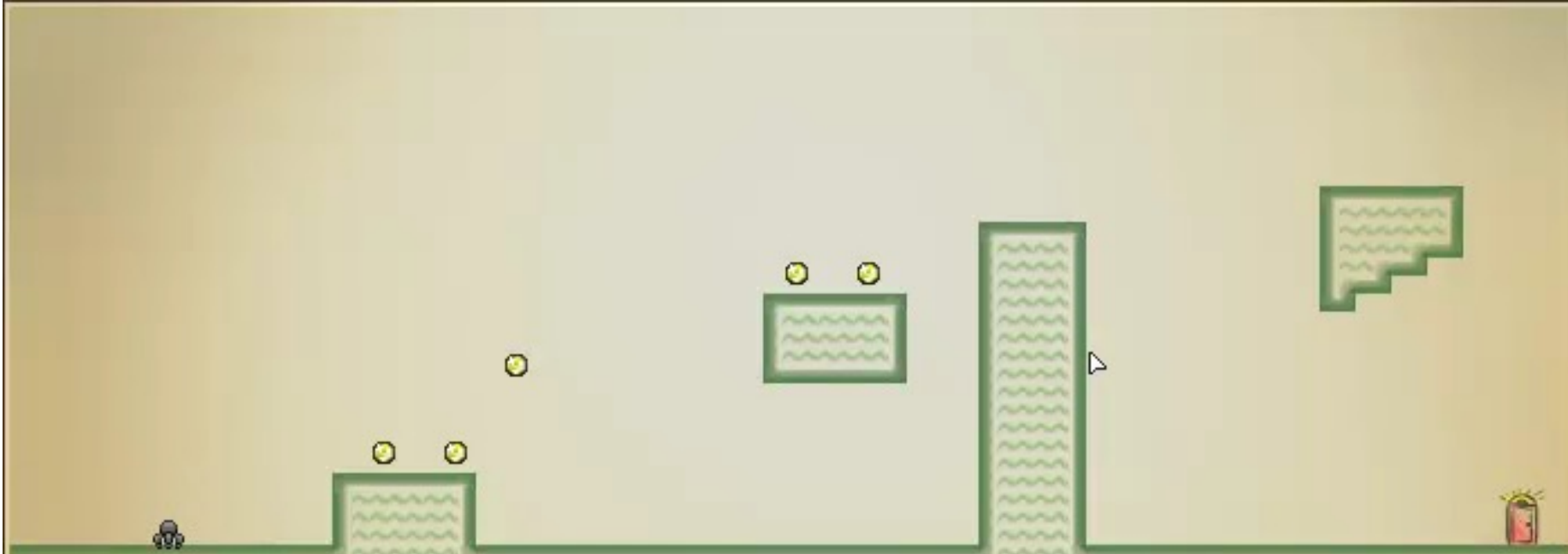


COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 7
PAR TIME: 45
SPEED TIME: 12

ALPHA

COINS 🟡: 1/6
STARS ☆: 0
POINTS: 0

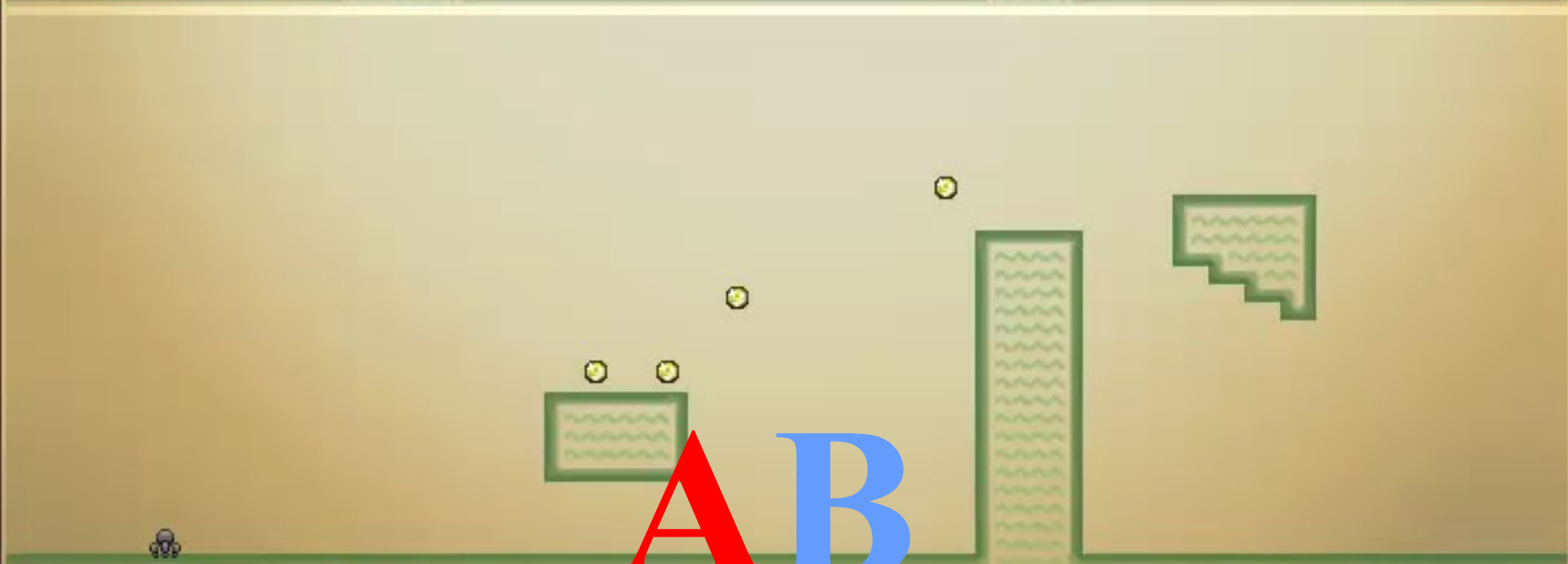
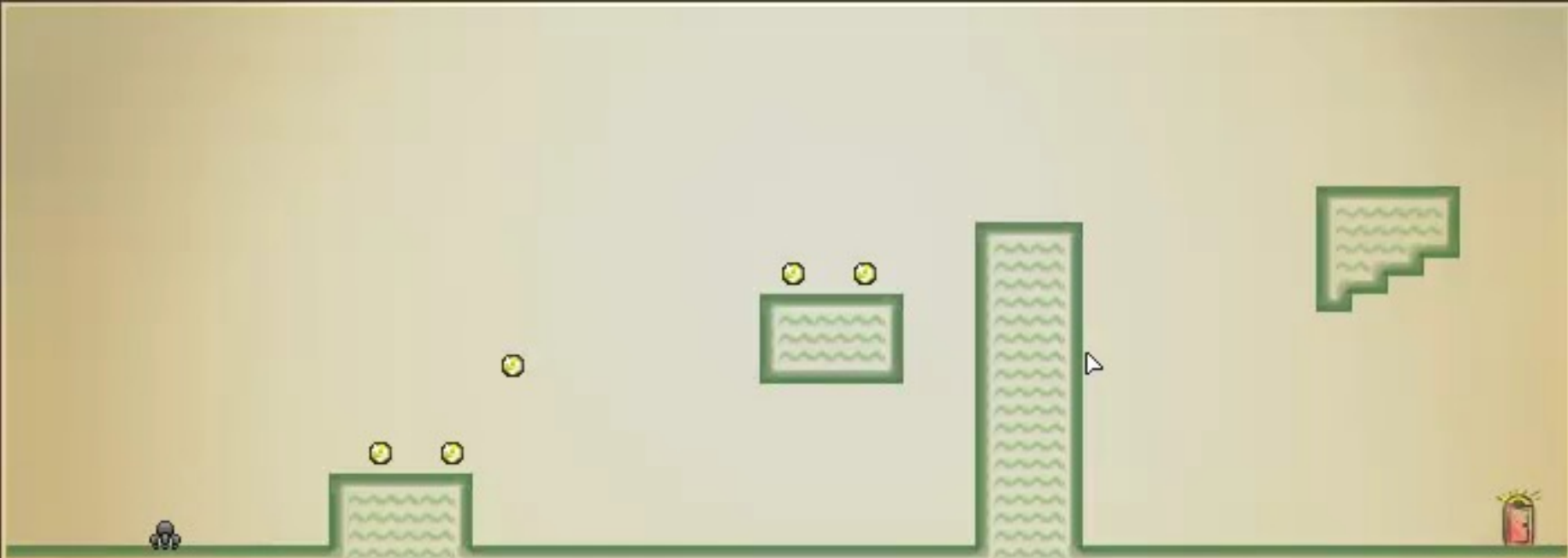


COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 0
PAR TIME: 60
SPEED TIME: 10

BETA

COINS 0/9
STARS 3
POINTS: 255



AB

COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 0
PAR TIME: 60
SPEED TIME: 10

BETA

COINS 0/9
STARS 3
POINTS: 255



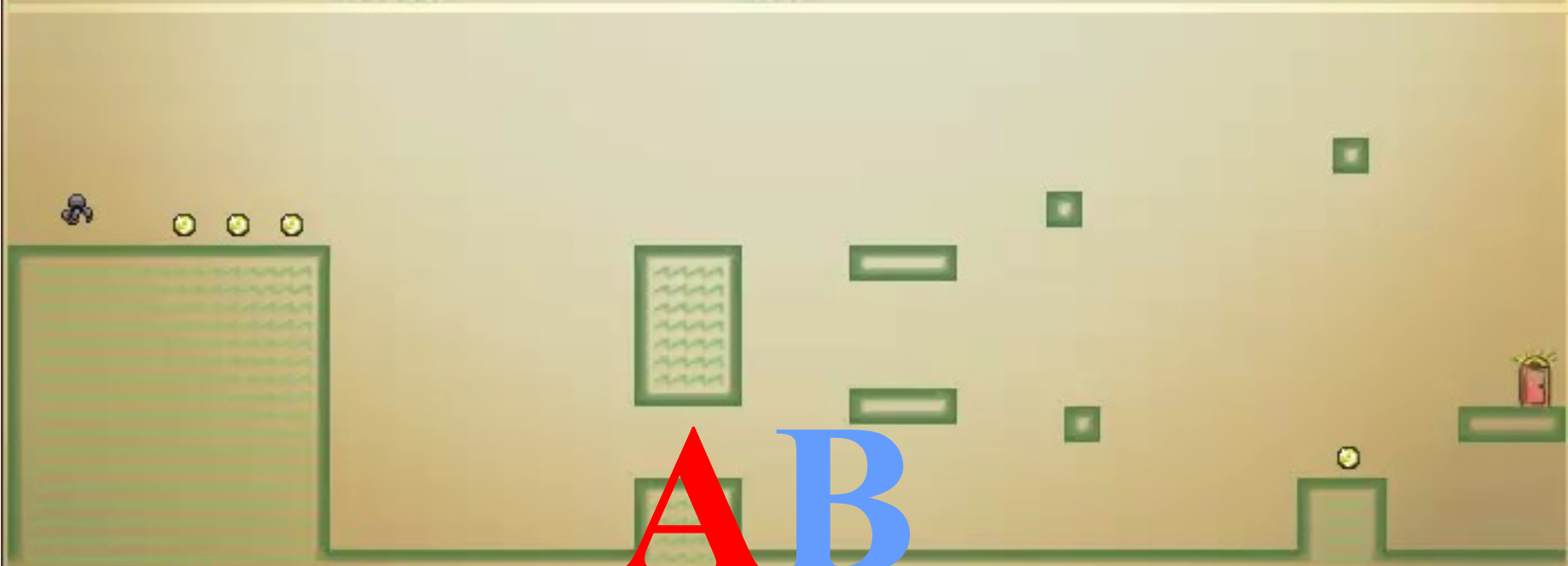
COMBO(C) REWIND(V)
QUIT RESTART(R)



TIME: 0
PAR TIME: 60
SPEED TIME: 15

GAMMA

COINS 0/6
STARS 6
POINTS: 573

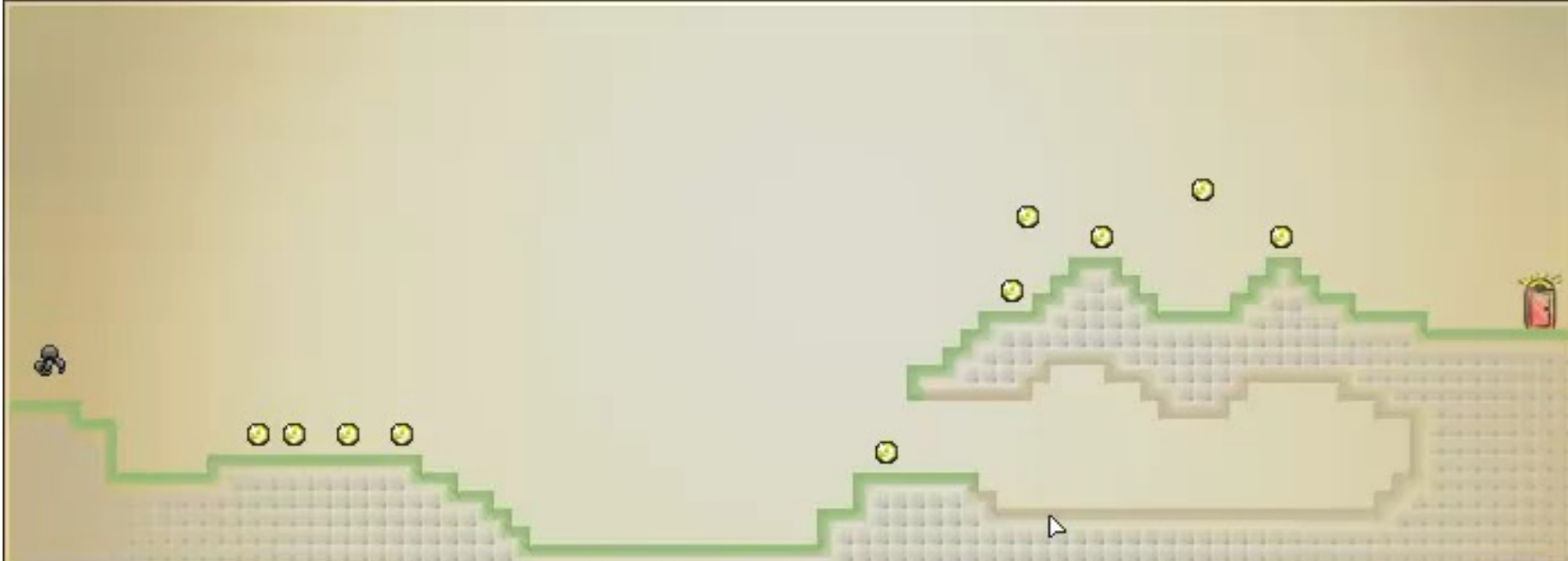


COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 0
PAR TIME: 60
SPEED TIME: 15

GAMMA

COINS 0/6
STARS 6
POINTS: 573



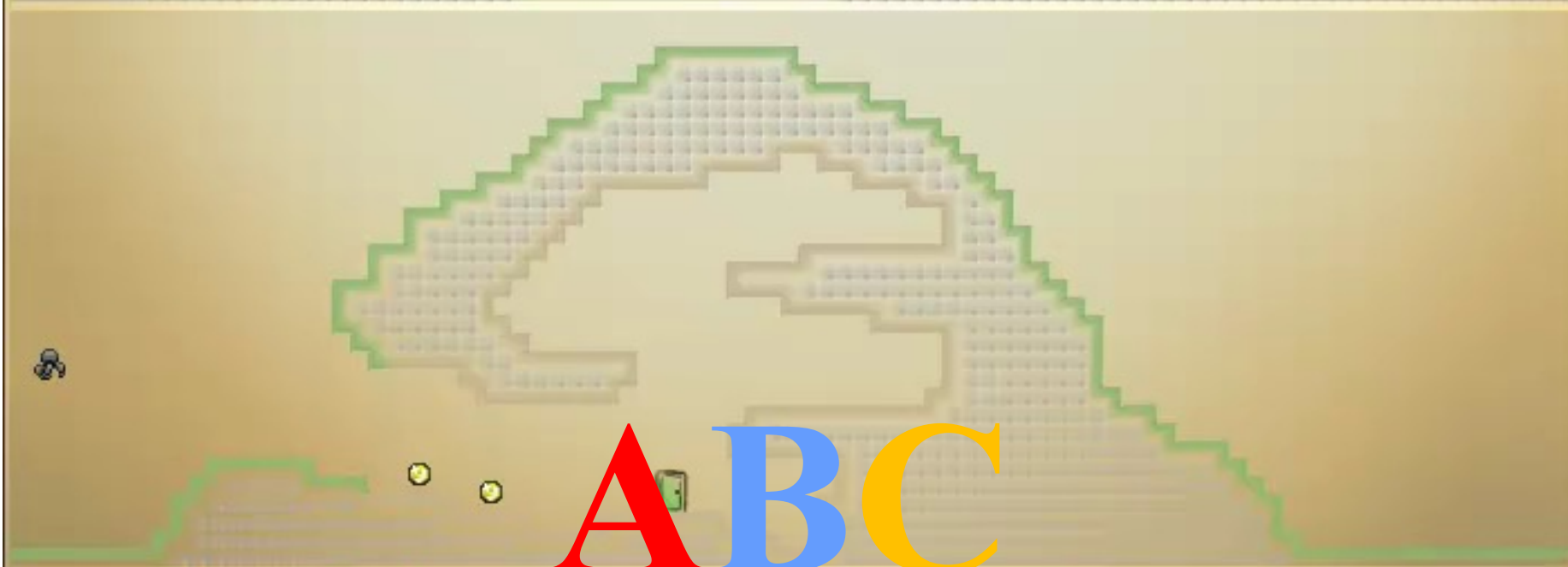
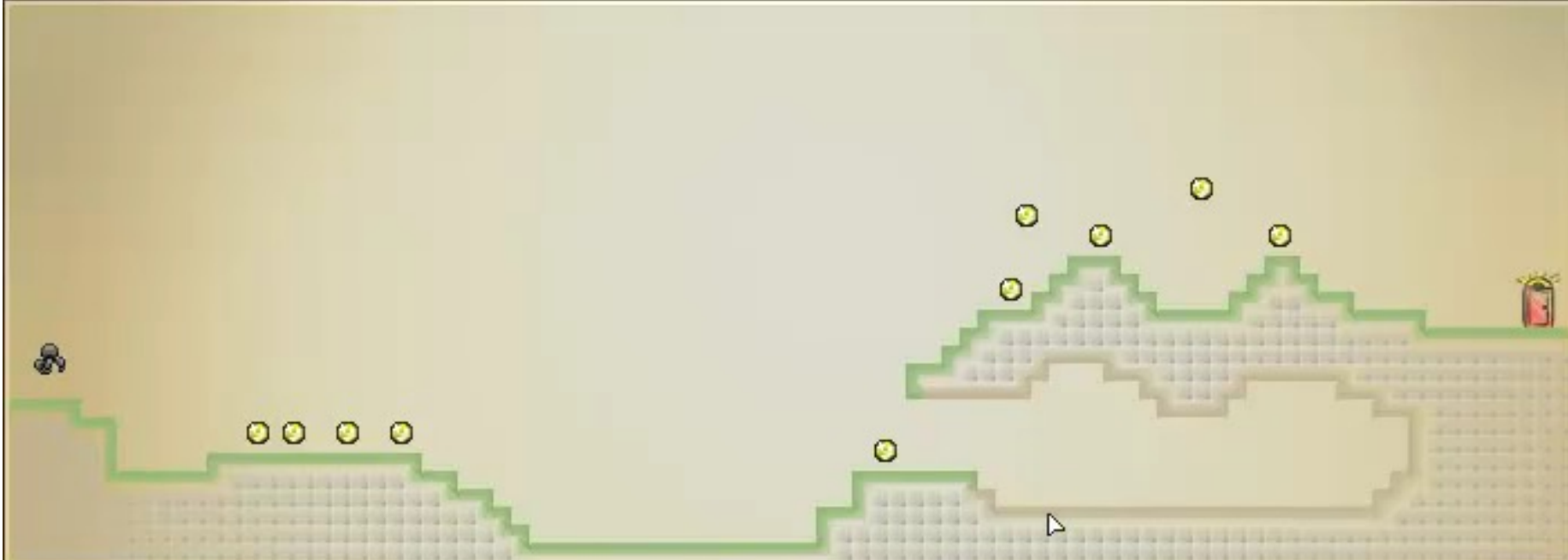
COMBO(C) REWIND(V)
QUIT RESTART(R)



TIME: 0
PAR TIME: 60
SPEED TIME: 26

MOUNTAINSIDE

COINS 0/12
STARS 9
POINTS: 879



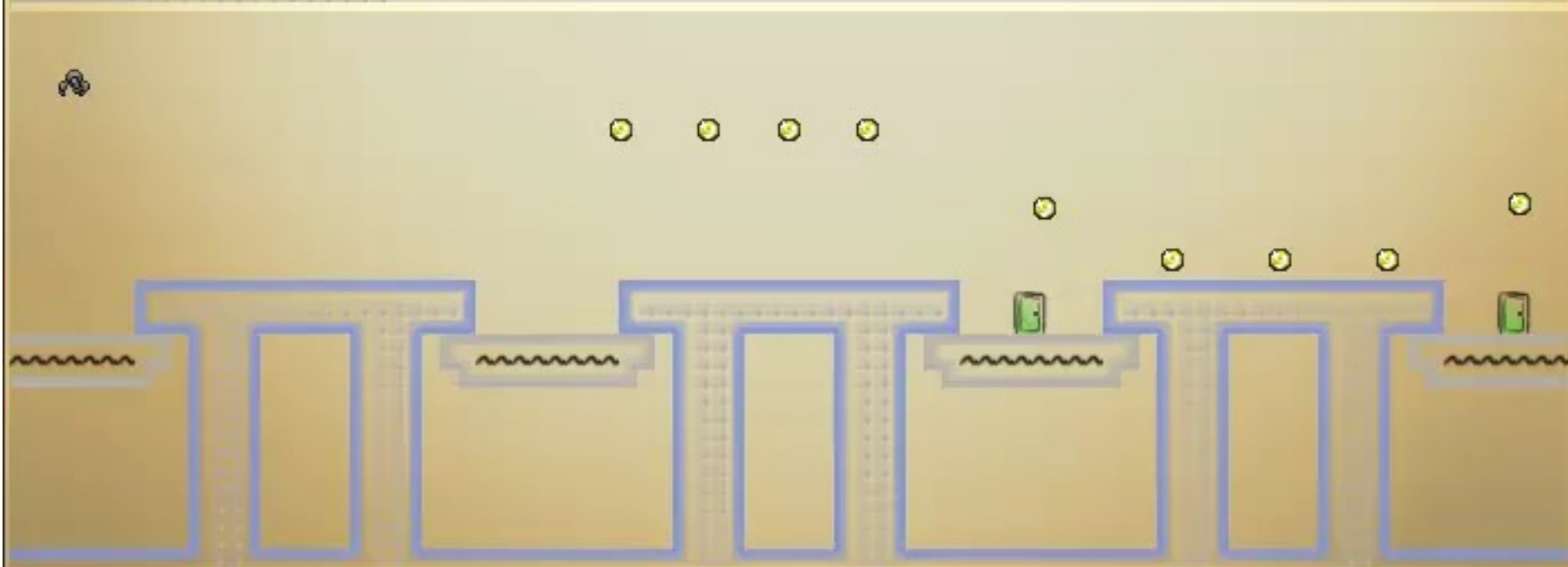
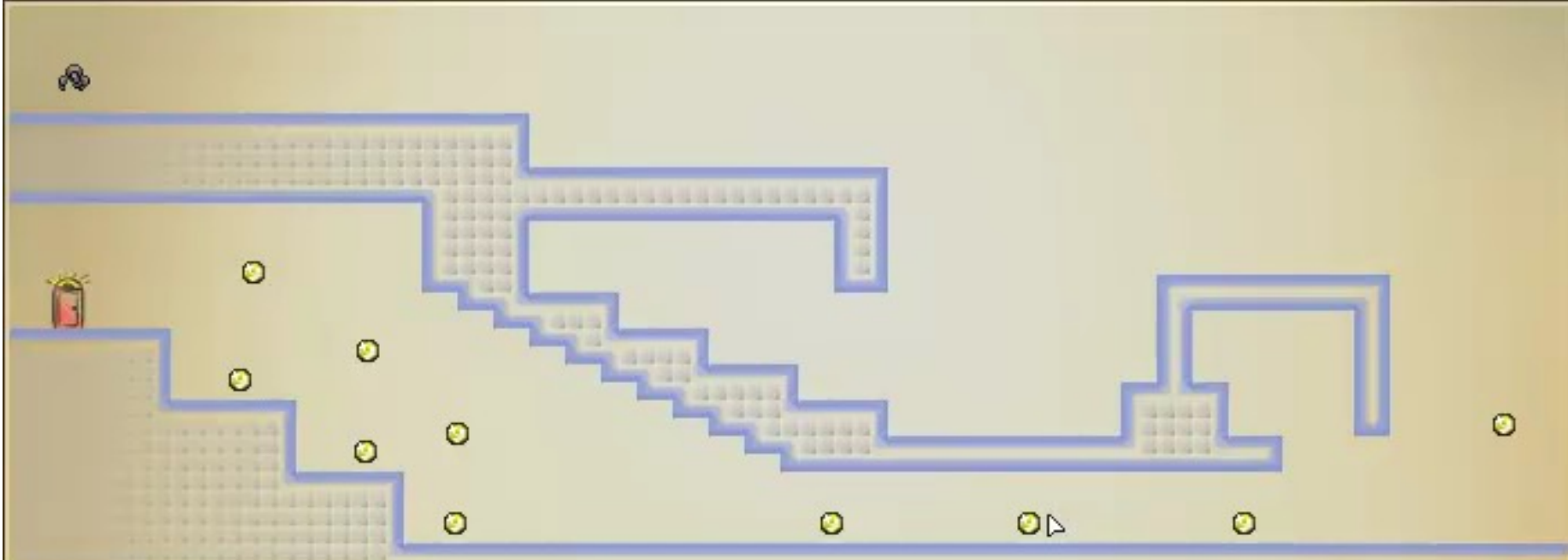
ABC

COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 0
PAR TIME: 60
SPEED TIME: 26

MOUNTAINSIDE

COINS 0/12
STARS 9
POINTS: 879



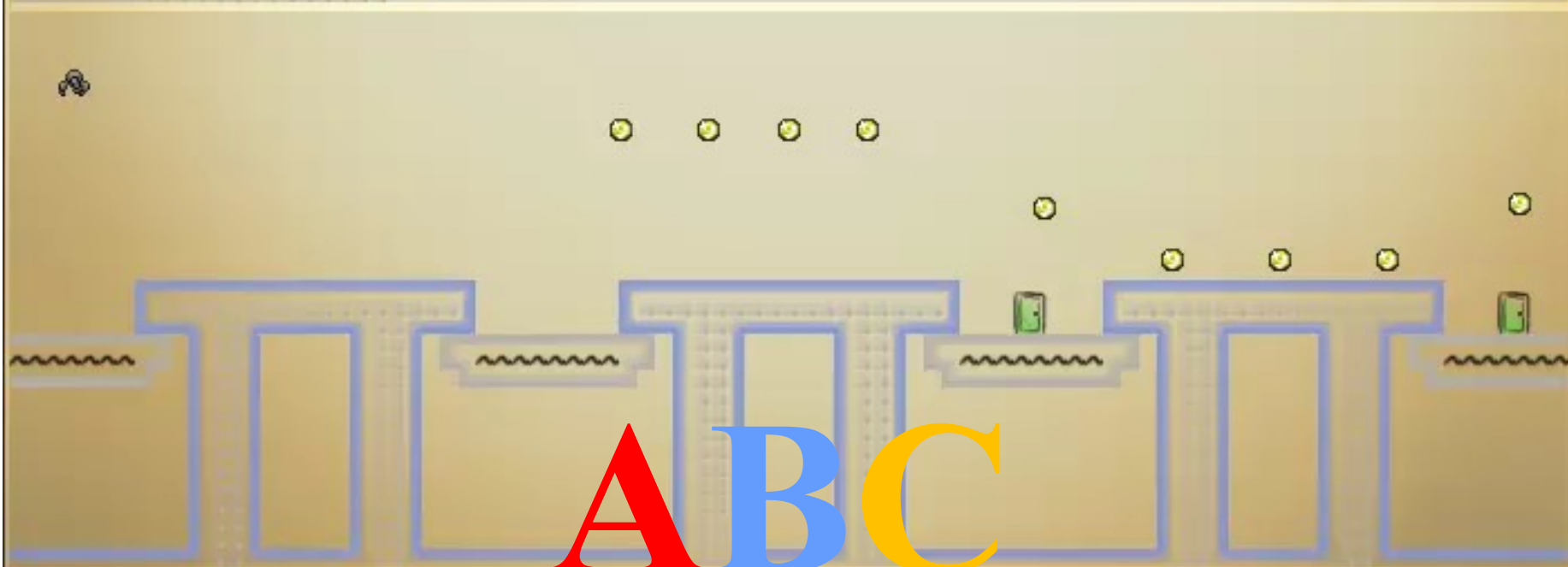
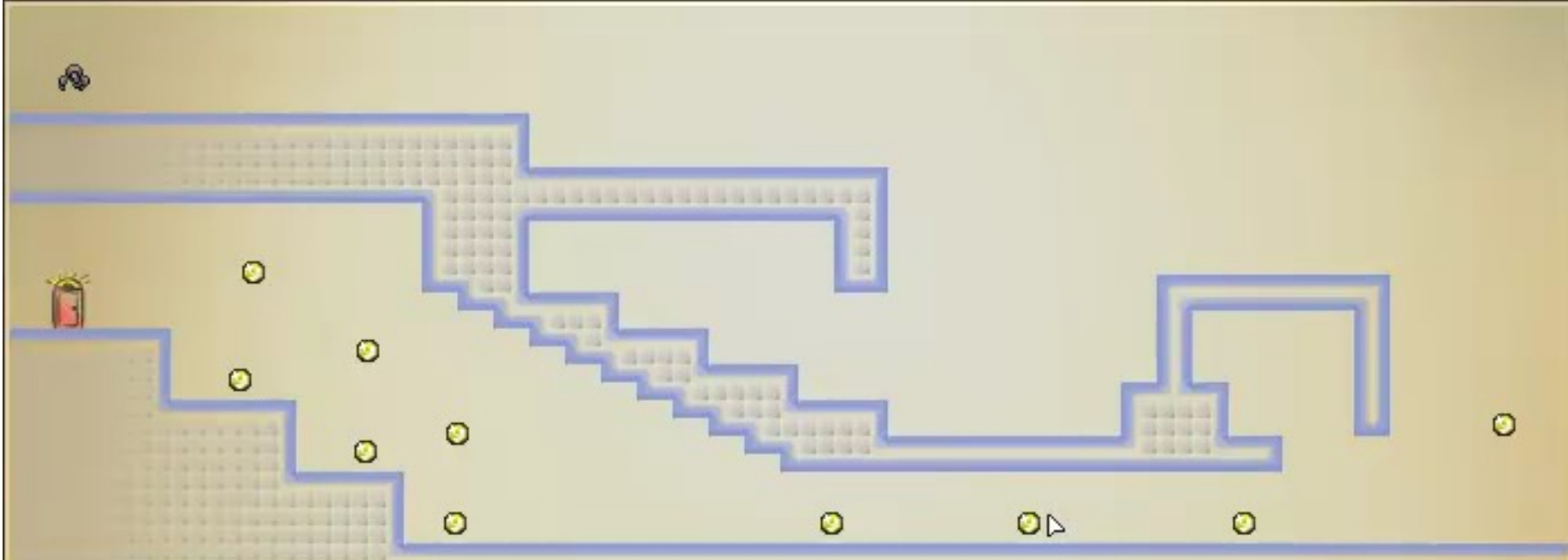
COMBO(C) REWIND(V)
QUIT RESTART(R)



TIME: 0
PAR TIME: 60
SPEED TIME: 30

PILLARS

COINS 0/19
STARS 12
POINTS: 1203



A B C

COMBO(C) REWIND(V)
QUIT RESTART(R)

TIME: 0
PAR TIME: 60
SPEED TIME: 30

PILLARS

COINS 0/19
STARS 12
POINTS: 1203

Hello Worlds

Mechanics:

A = move **B** = two worlds **C** = close world

A

AB

AB

ABC

ABC

Moderate reinforcement, high recombination

Starcraft



Starcraft

A AB ABC ABCD

Low reinforcement, high recombination

A B C D

A A A A

Summary

- Level design is always important
 - How keep your game different, lively?
 - How do you train your player?
- Level design uses **geographic constraints**
 - Create challenges by defining *design patterns*
 - Storyboard so player must go through challenges
- Level design uses **ludic constraints**
 - Do not introduce all of your capabilities at once
 - Leverage *reinforcement* and *recombination*