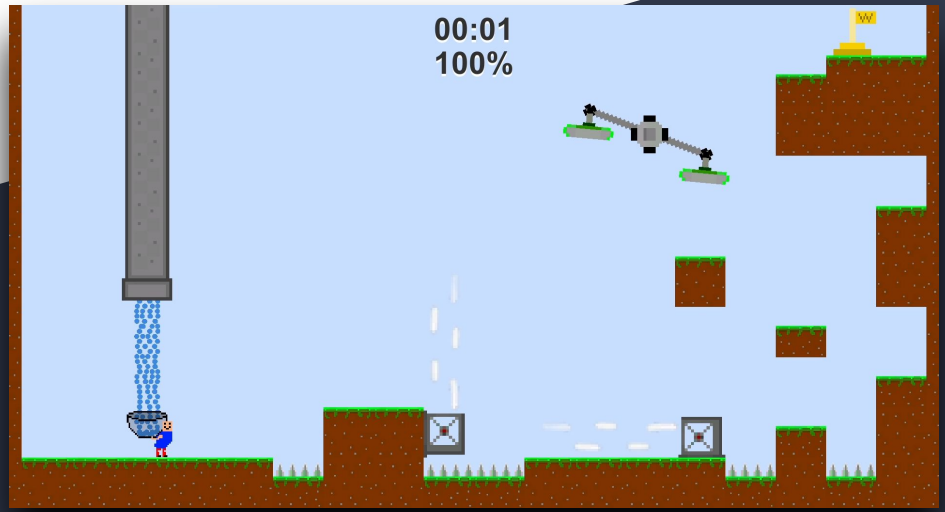
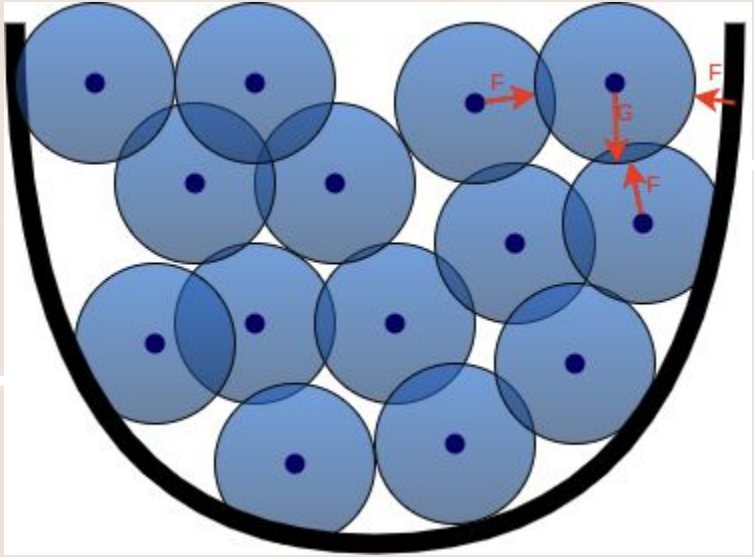
The Leak

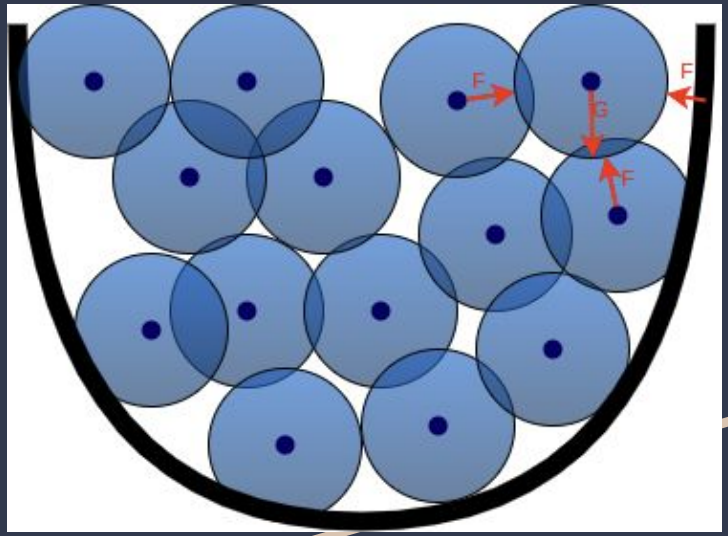
David Campos Rodríguez guscampda@gu.student.se
John Segerstedt sejohn@student.chalmers.se



Game Mechanic: Carrying Water

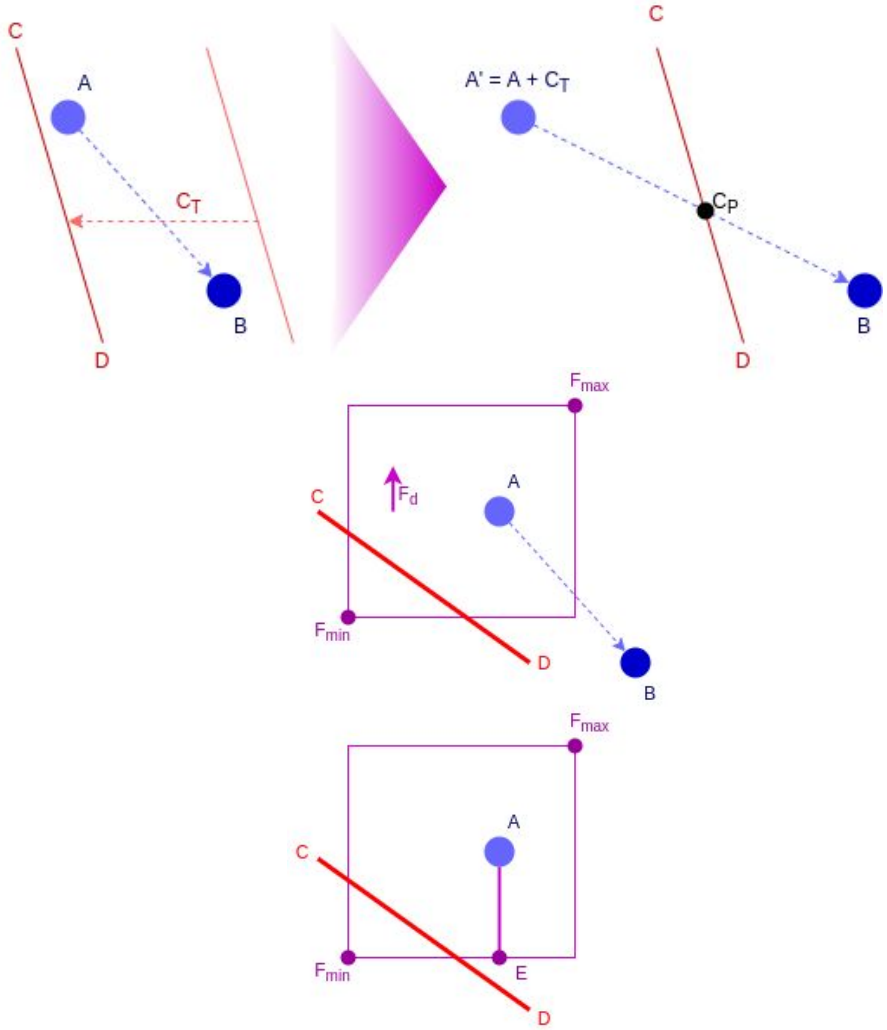


Implementation



- Based on blog post by Peeke Kuepers:
  Simulating blobs of fluid - source
- Unity Burst
- Uniform grid
- Interaction with glass, floor and fans

Collider and fan interaction

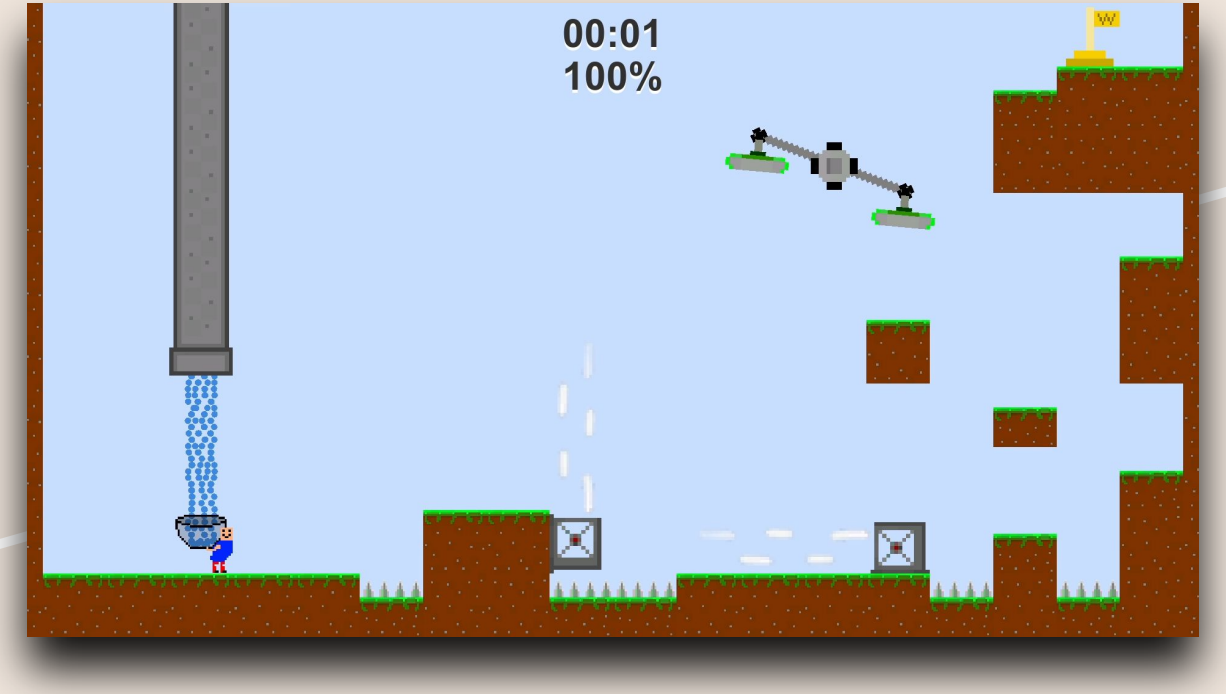


Problems and Limitations

- Water gets arbitrarily compressed when pushed by colliders
- The grid size is fixed and it needs quite a good space amount
- No spatial structure for colliders and fans
- Water can be better optimized
- Collider rotation / scale not fully supported
- Only axis-aligned fans allowed
- Would need more time

Carrying Water

Game Context: 2D Platformer



Gameplay Patterns

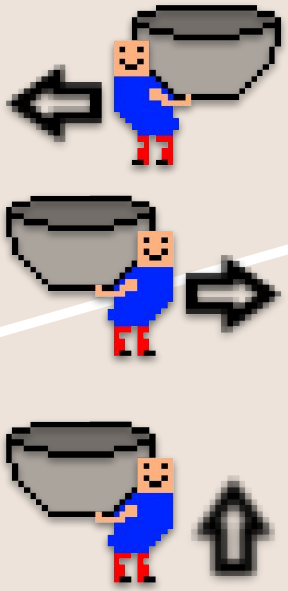
- Evade
- Game Over
- Game State Indicators
- Maneuvering
- Movement
- Obstacles
- Predefined Goals
- Resources
- Scores
- Single-Player Games
- Speedruns

Björk, S. (2018).Patterns. Accessed on 25 November 2020. url:http://virt10.itu.chalmers.se/index.php/Category:Patterns.

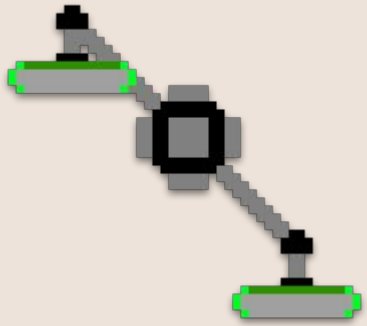
Implementation

User input:

- Left
- Right
- Jump



Implementation



User input:

- Left
- Right
- Jump

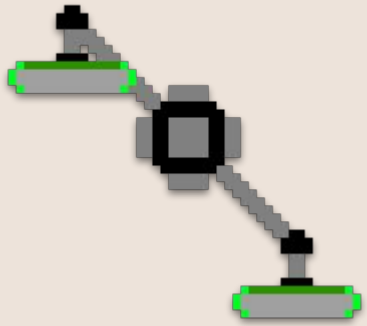
Obstacles:

- Carrousel
  - Core fixed in world space
  - Arms rotate 360 degrees
  - Hands fixed w/ hinge joints
  - Colliders marked in green

Implementation

User input:
- Left
- Right
- Jump

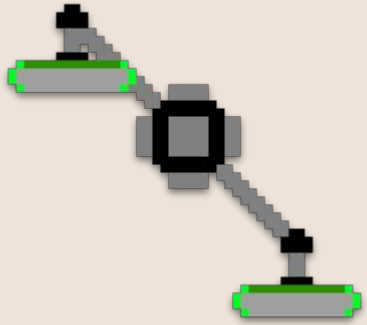
Obstacles:
- Carrousel
- Fans
  - Fan blades “animated”
  - Air funnel is trigger
  - Trigger applies “airforce”

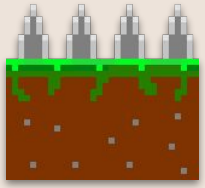


Implementation

User input:
- Left
- Right
- Jump

Obstacles:
- Carrousel
- Fans
- Spikes
  - Spike tips - Trigger *(kills player)*
  - Ground - Collider



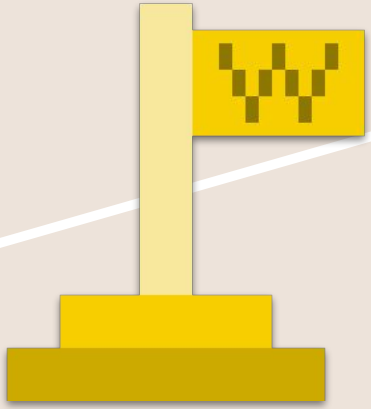


Implementation

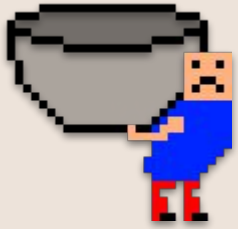
User input:
- Left
- Right
- Jump

Obstacles:
- Carrousel
- Fans
- Spikes

Win Condition:
- Reach Flag



Implementation



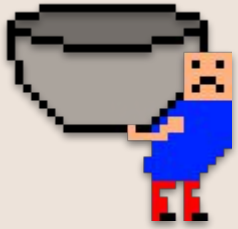
User input:
- Left
- Right
- Jump

Obstacles:
- Carrousel
- Fans
- Spikes

Win Condition:
- Reach Flag

Loss Conditions:
- Empty bowl

Implementation



User input:
- Left
- Right
- Jump

Obstacles:
- Carrousel
- Fans
- Spikes

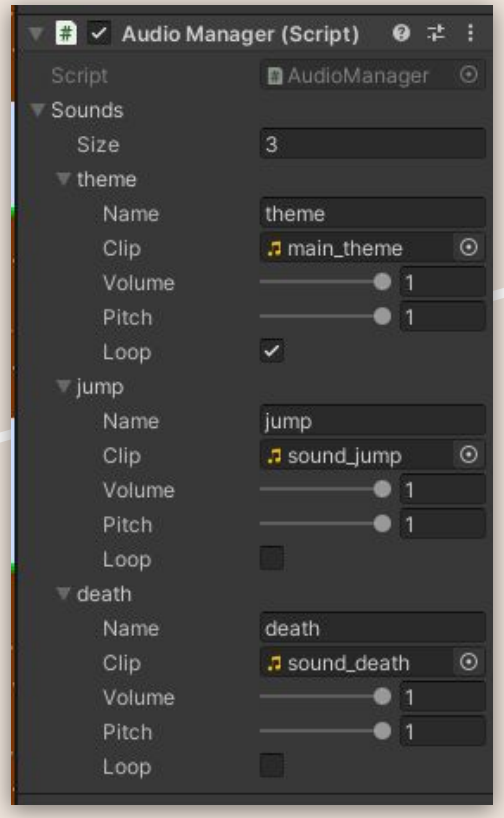
Win Condition:
- Reach Flag

Loss Conditions:
- Empty bowl
- Death

Implementation

Audio

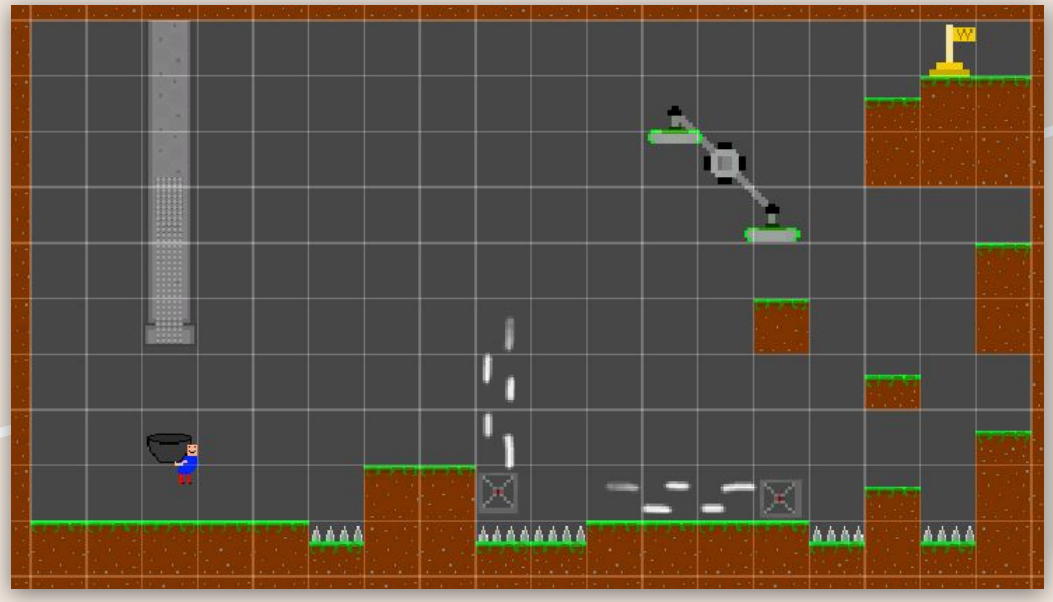
FindObjectOfType<AudioManager>().PlaySound("sound name");

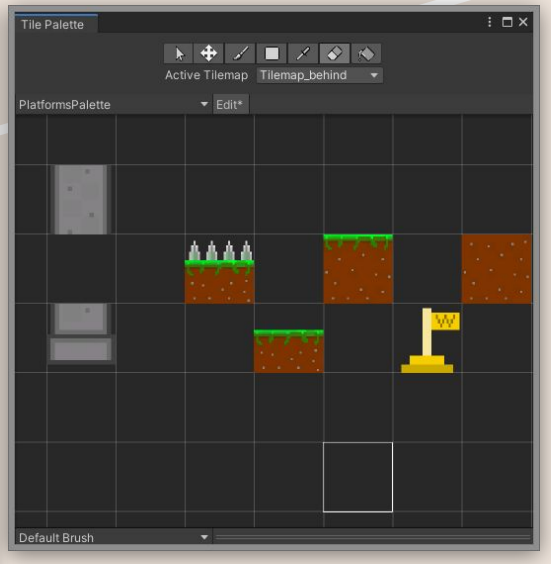


Implementation

Unity Tilemap

- Enables easy level-design
- Different sub-Tilemaps within the same Grid





Implementation

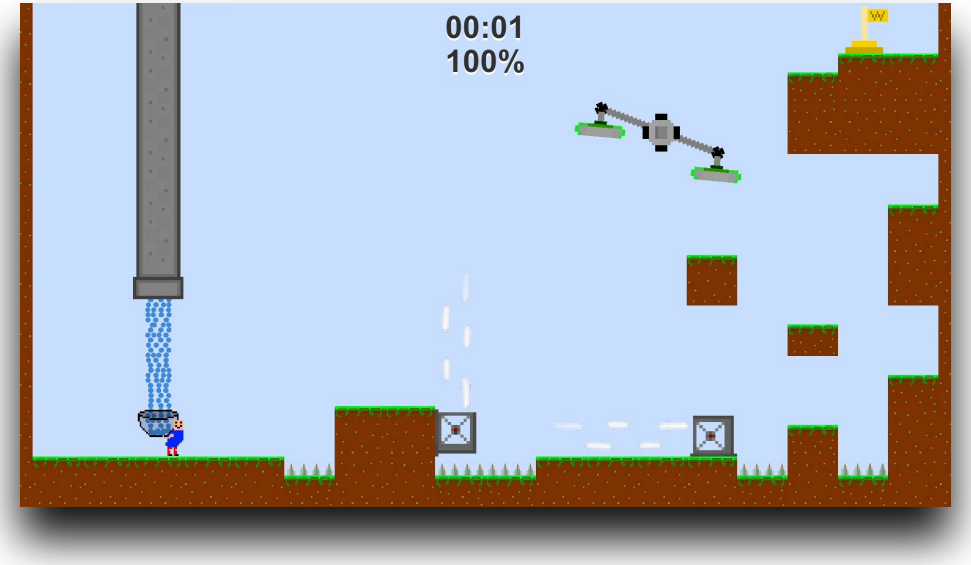
Separation of Concerns, MVC style:

- Player *(“model” - state)*
- PlayerSprites *(“view” - display state)*
- PlayerMovement *(“controller” updates state)*

Problems and Limitations

- Currently fixed camera *(heavily limits map size)*
- Many world objects fixed & un-interactable
- Needs a different Tilemap for each type of trigger/collider
- Tilemaps - useful but inflexible
- Single level

Results - Demo



The Leak

David Campos Rodríguez guscampda@gu.student.se
John Segerstedt sejohn@student.chalmers.se

Questions?