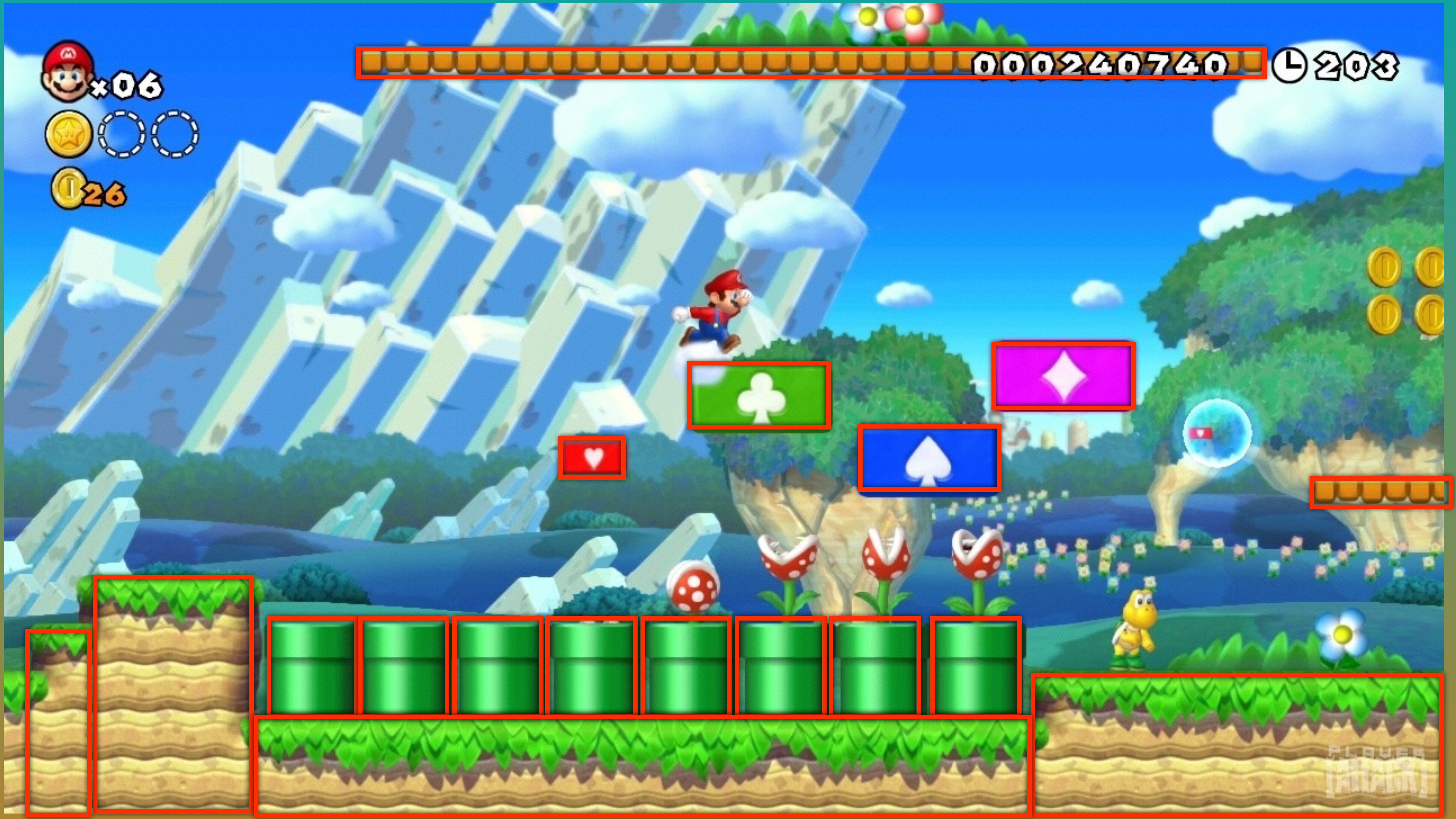
Platformer game physics.

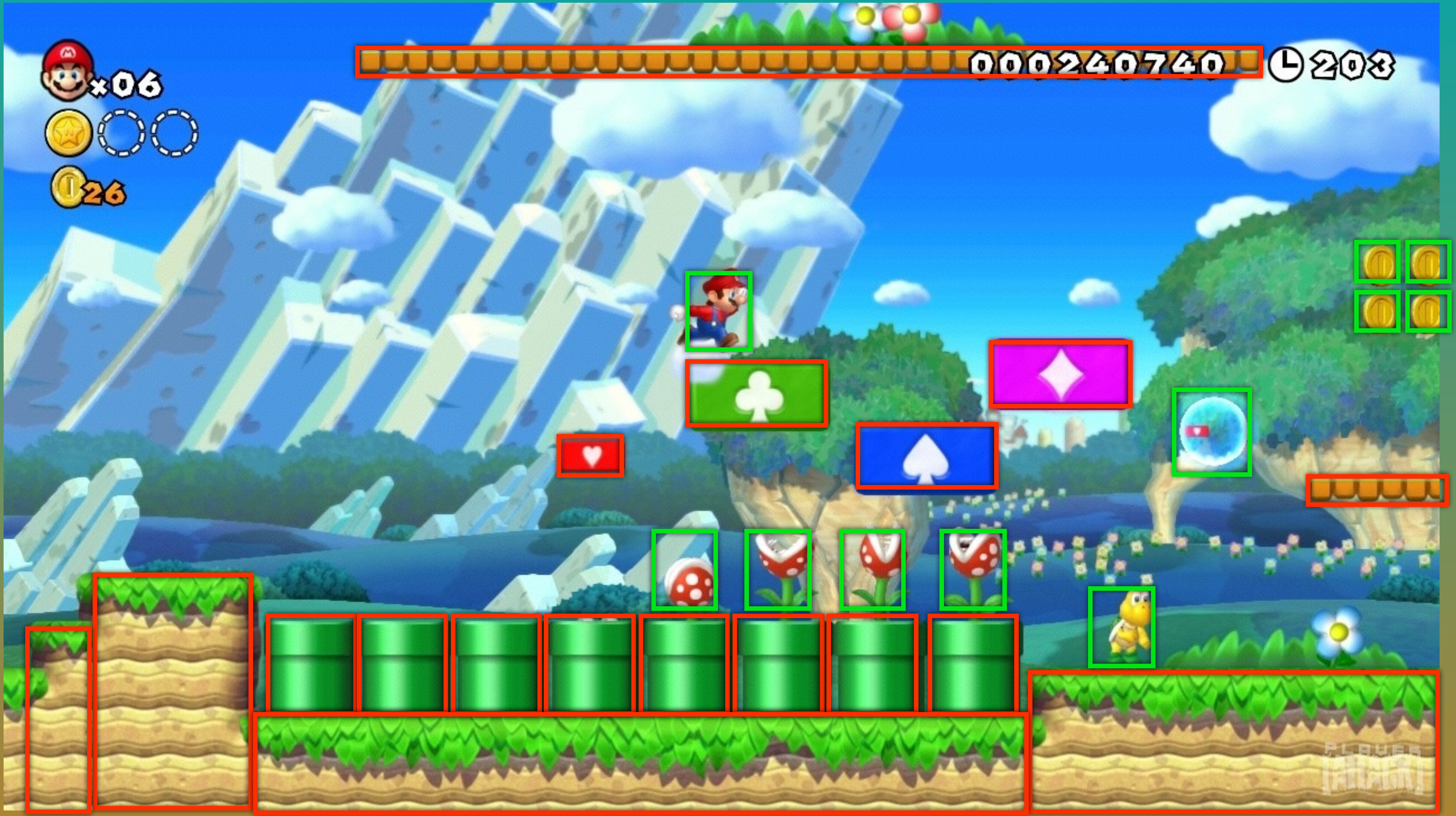
Platformer games overview.

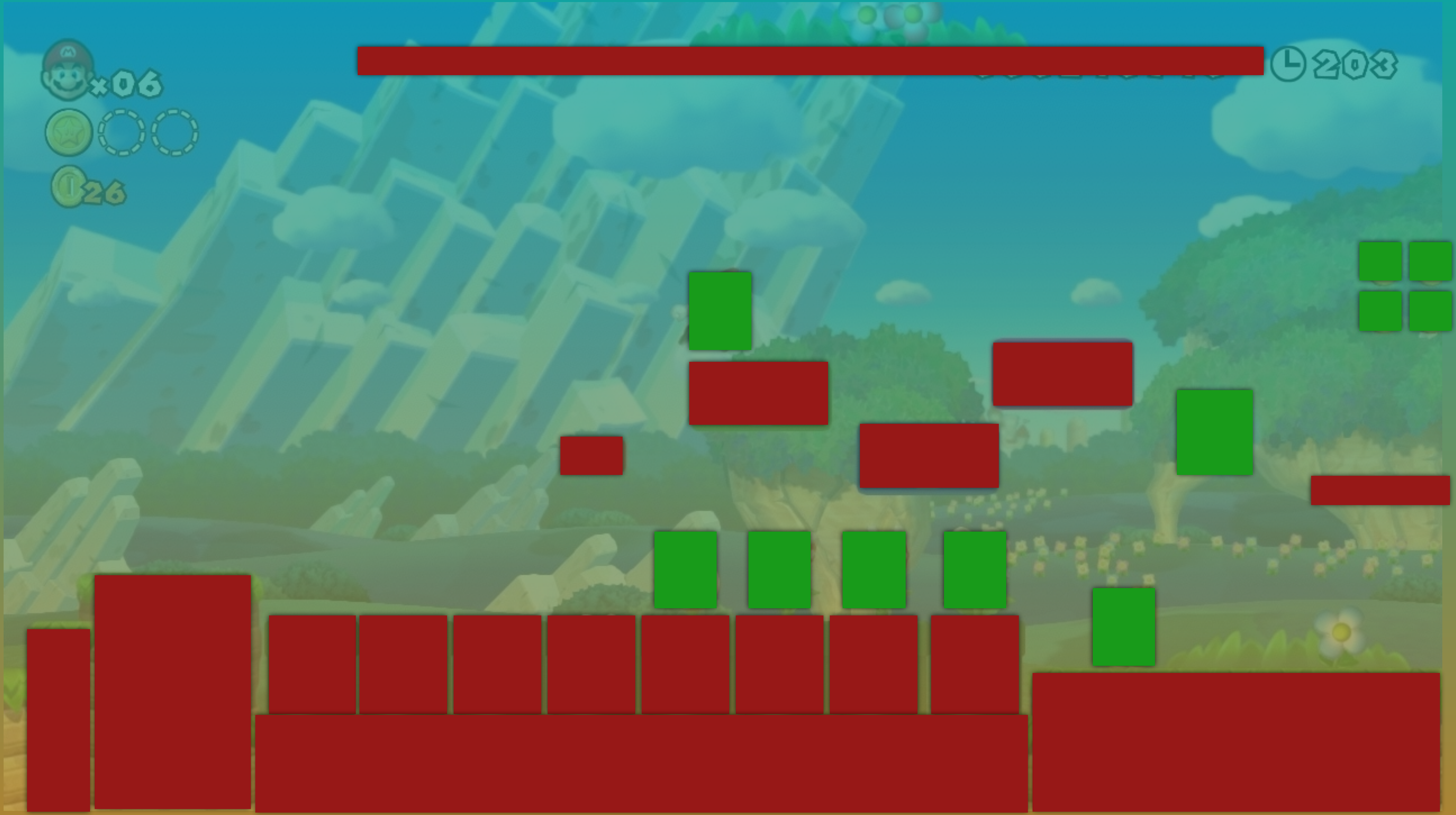
MARIO
004250
×05
WORLD
1-1
TIME
283
100

CITY

×06
26
000240740
203

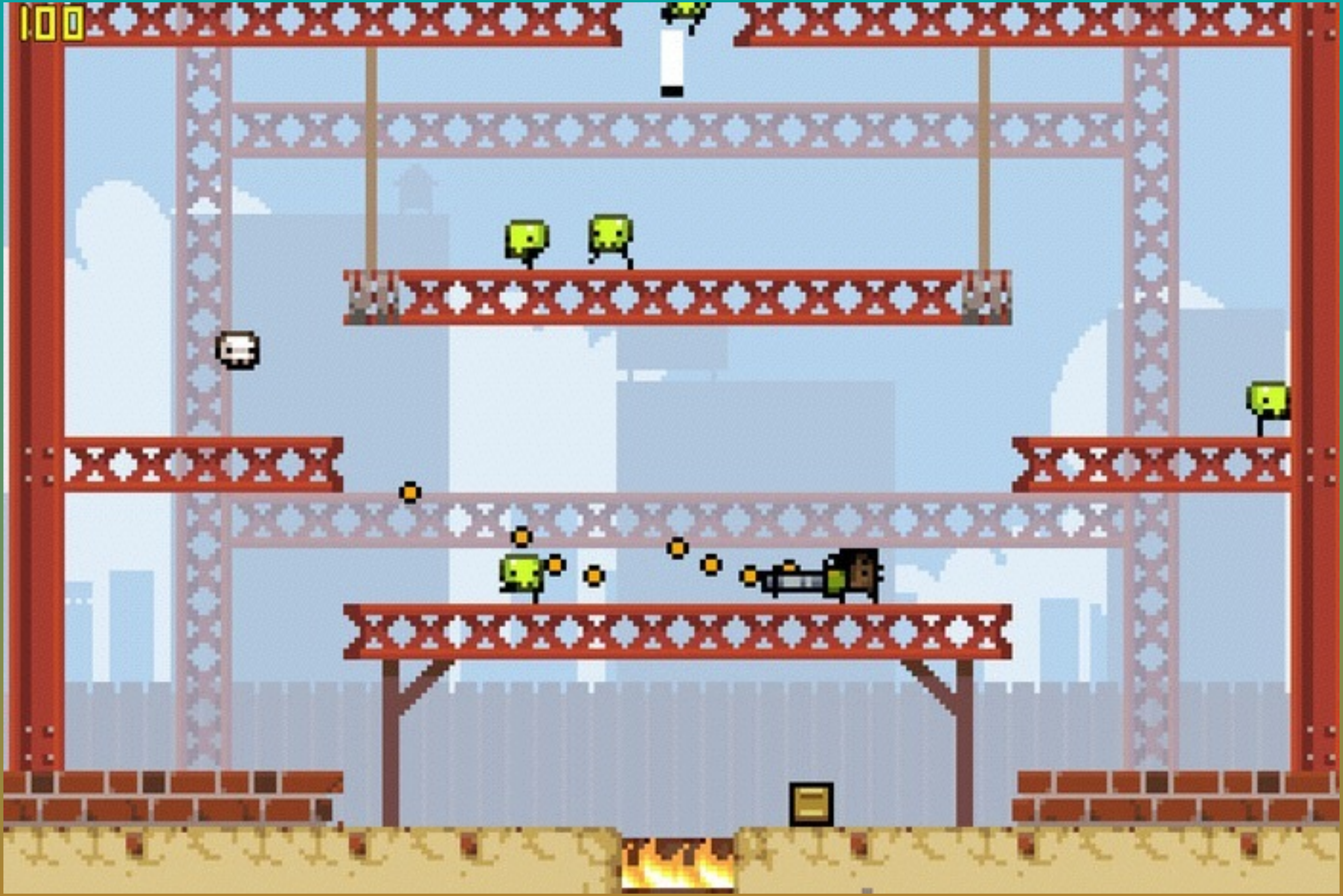

×06
26
000240740
203

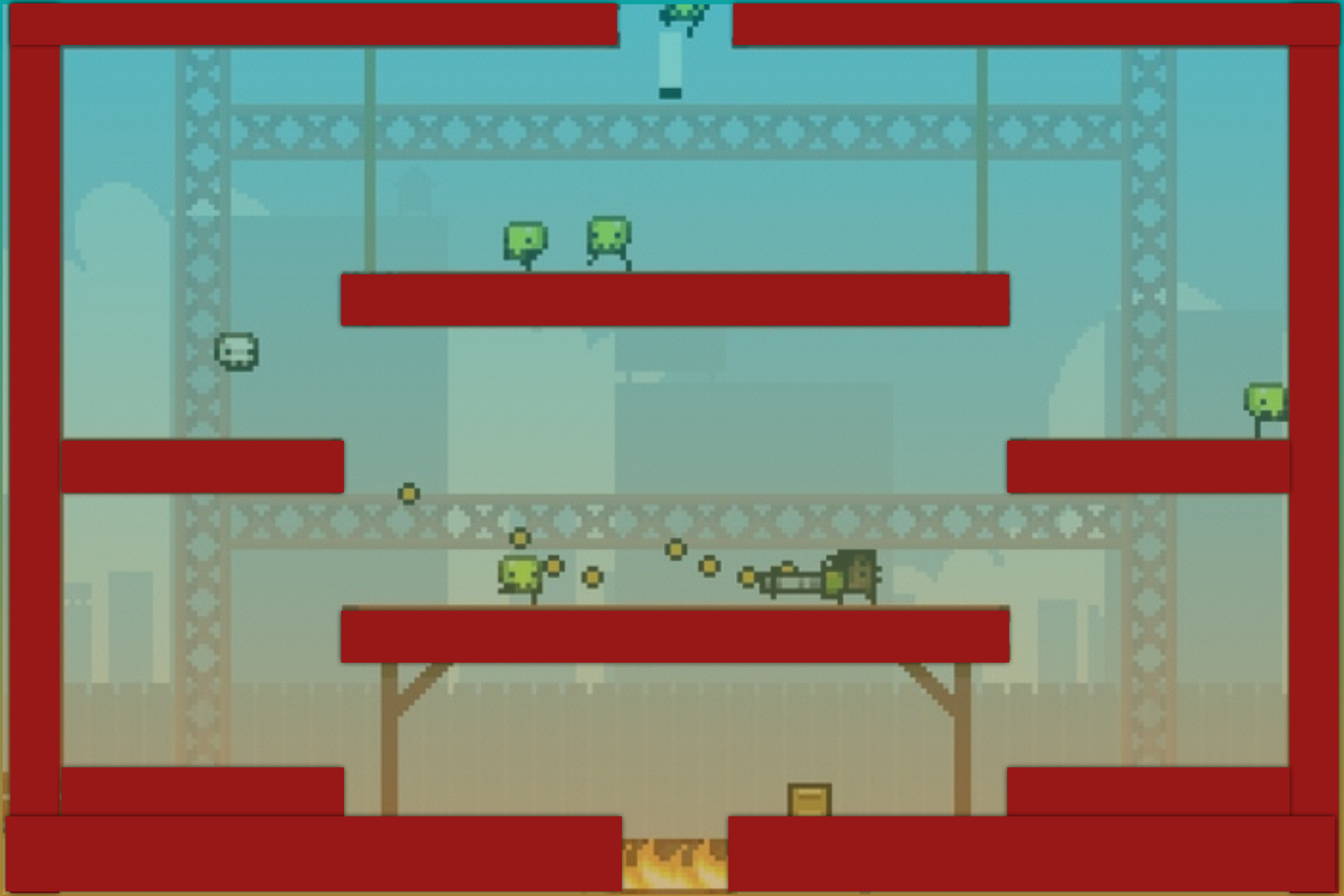

×06
26
000240740
203

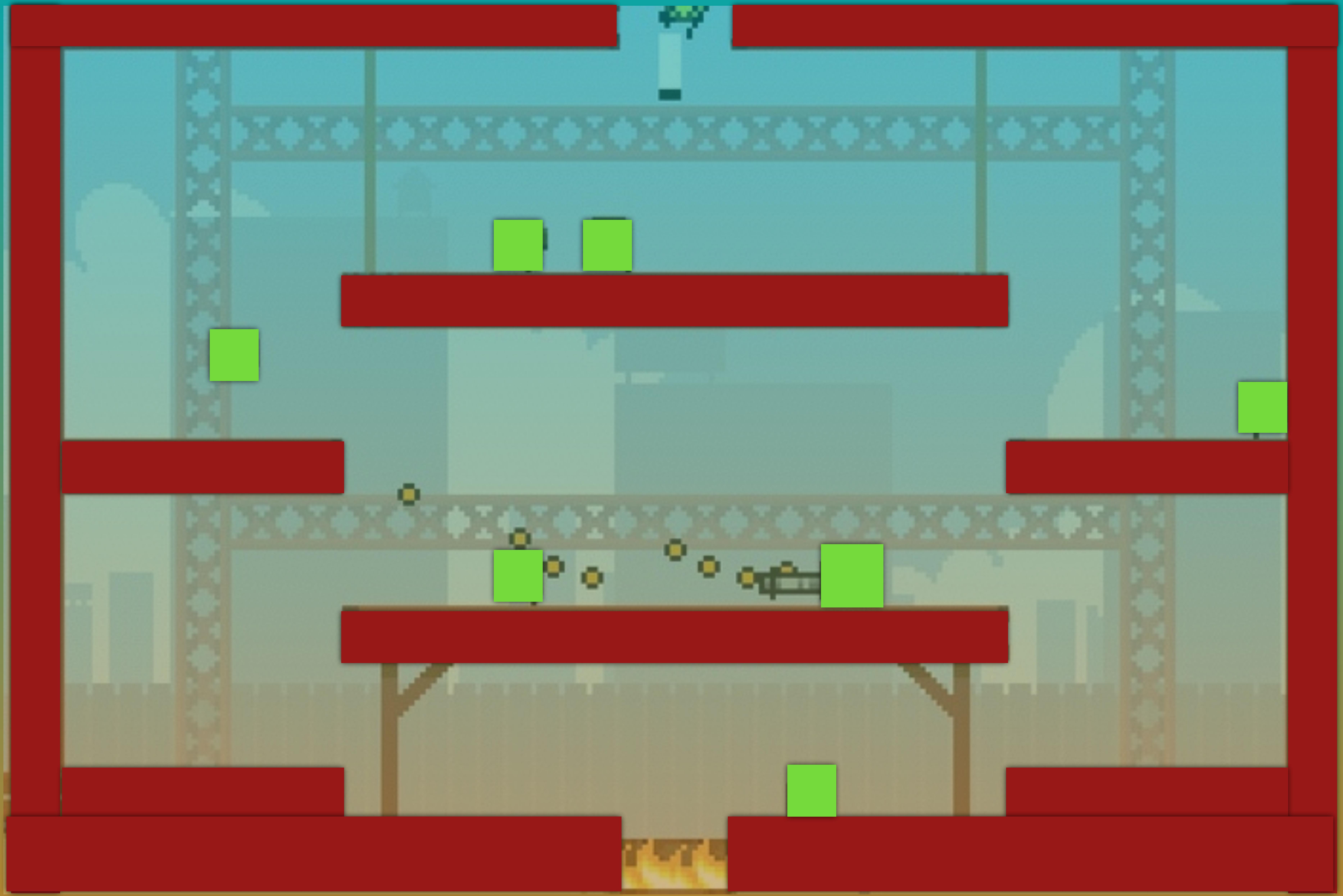

×06
26
203

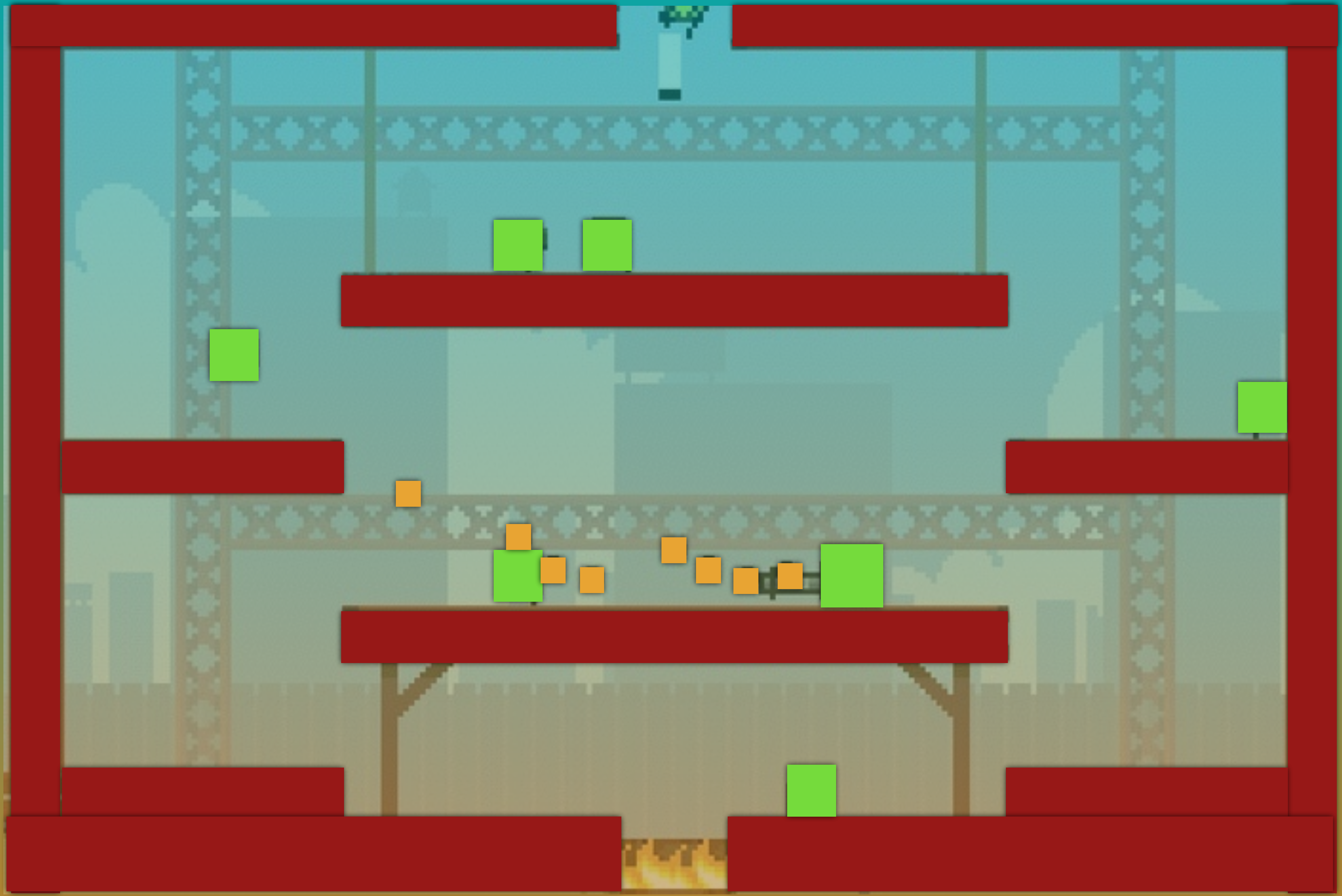
Building a single screen platformer.

Building a single screen platformer.

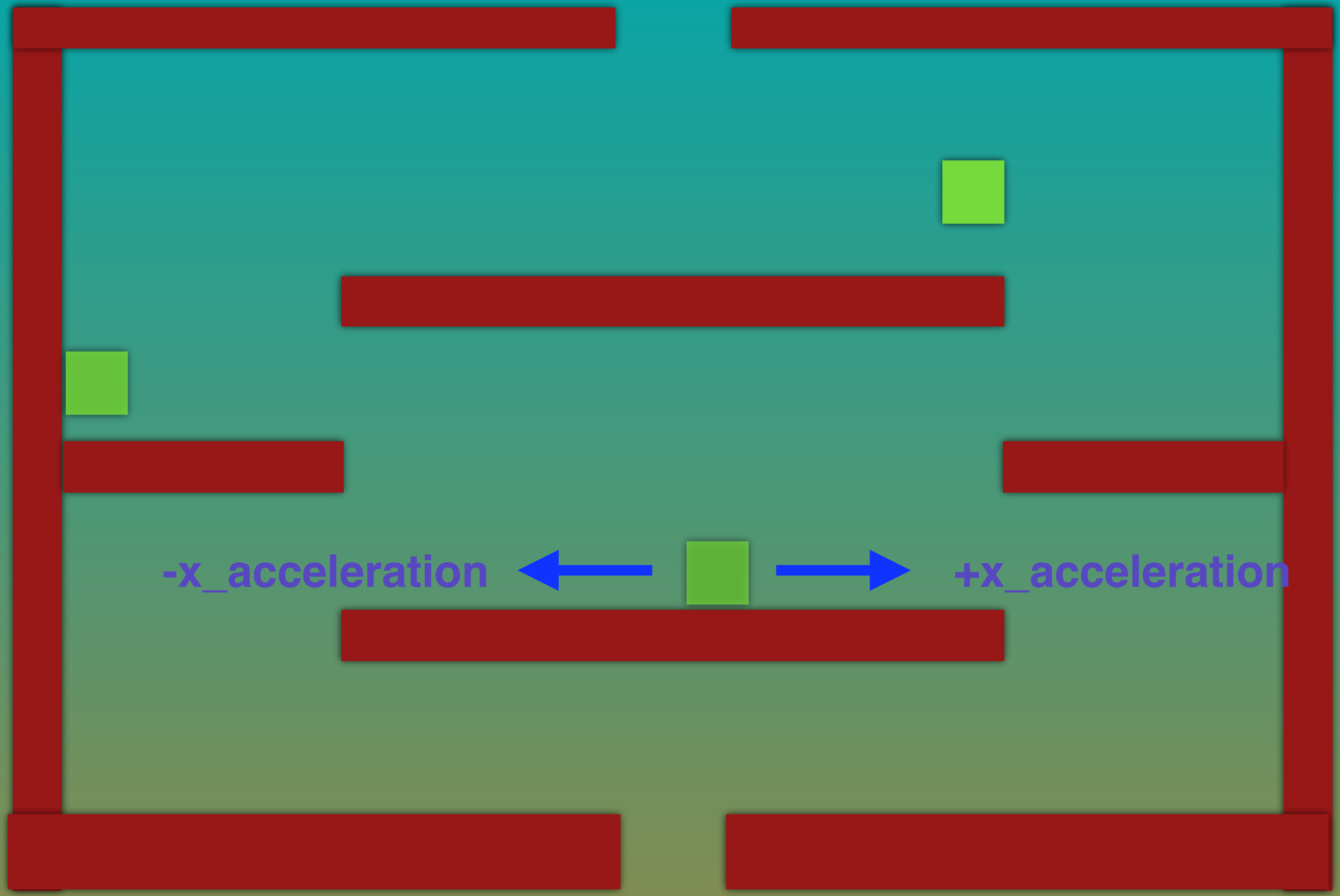

100





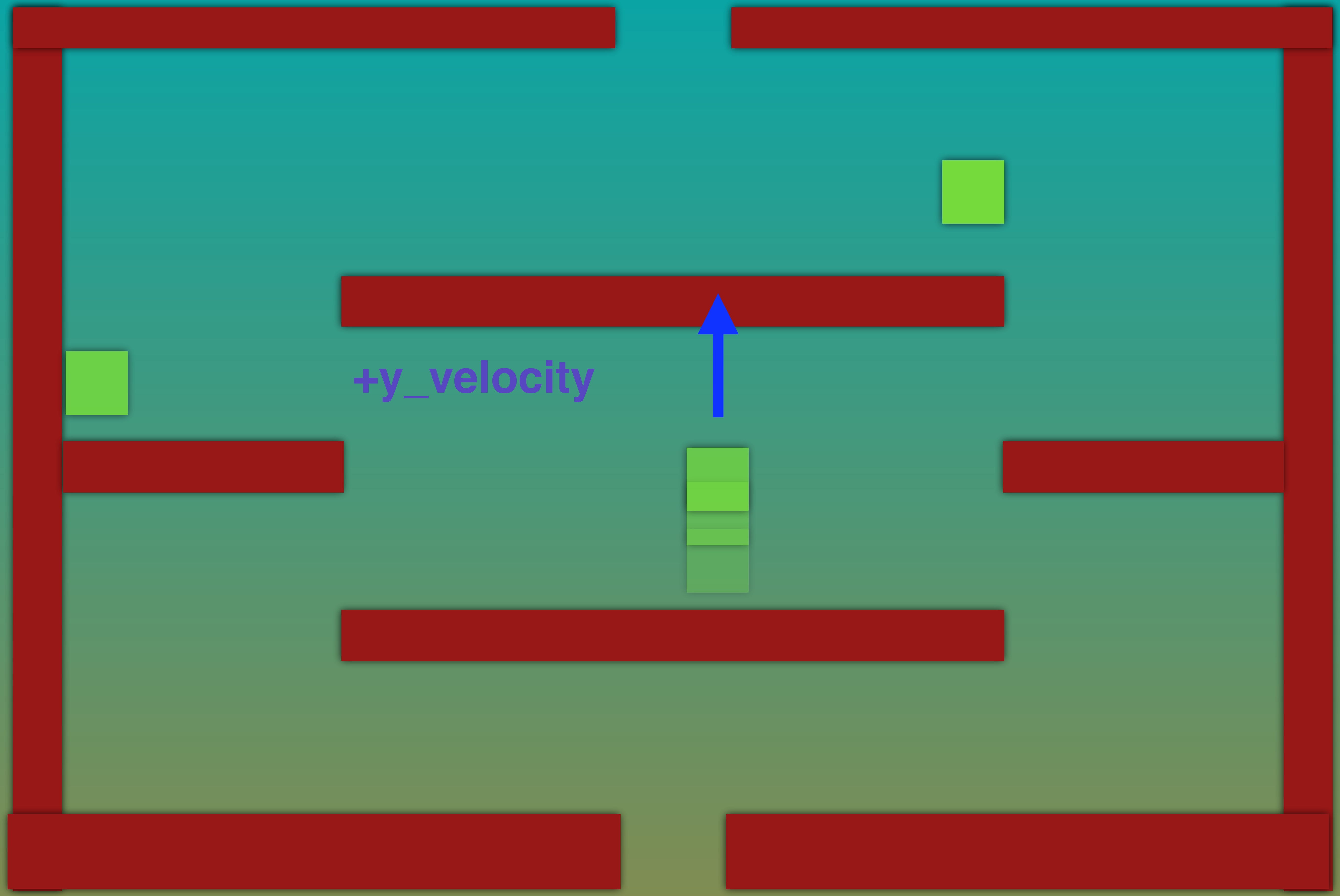


Movement.



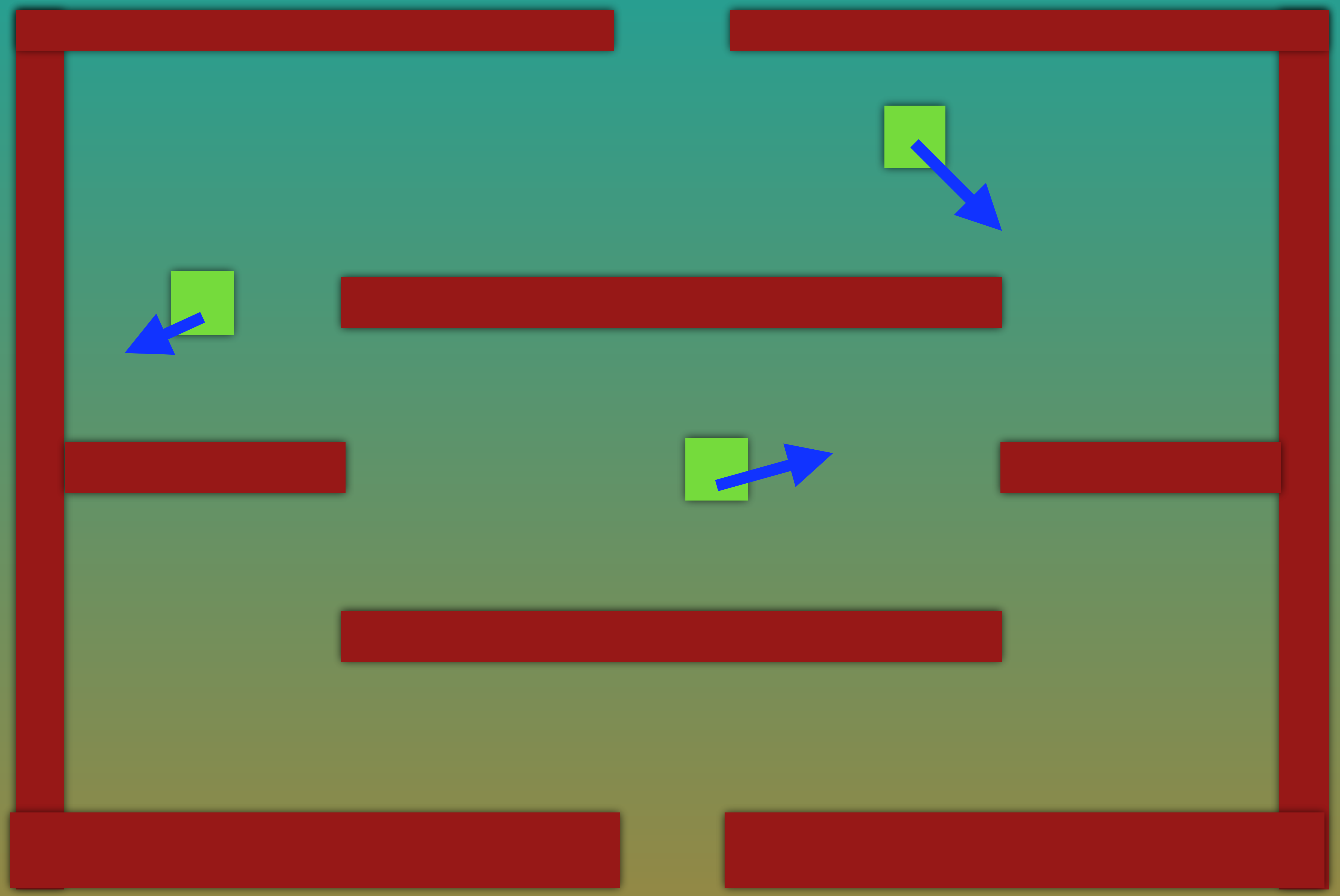
All left-right movement for dynamic entities is done by setting acceleration.

Jumping and forces.

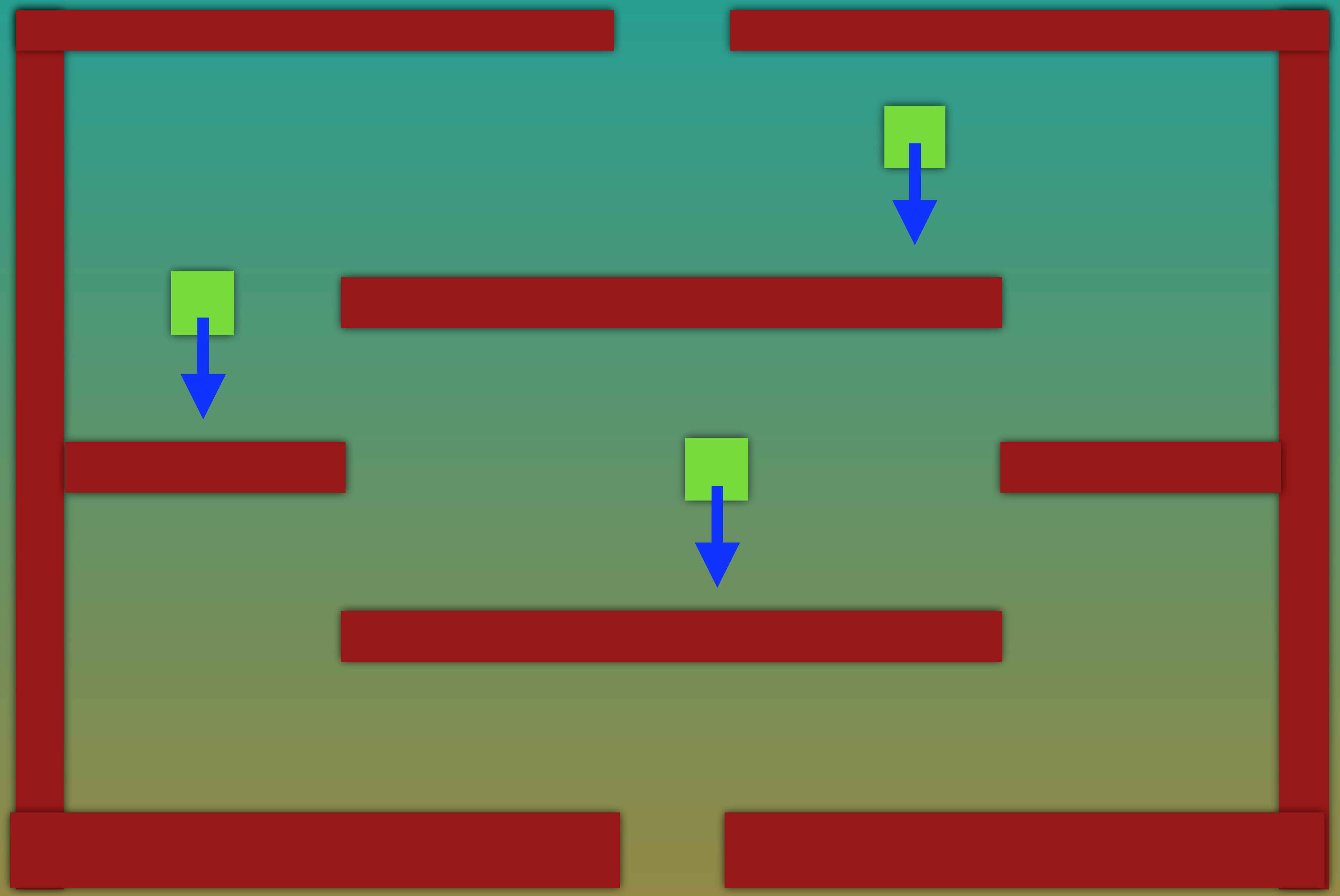


Jumping is done by setting velocity directly.

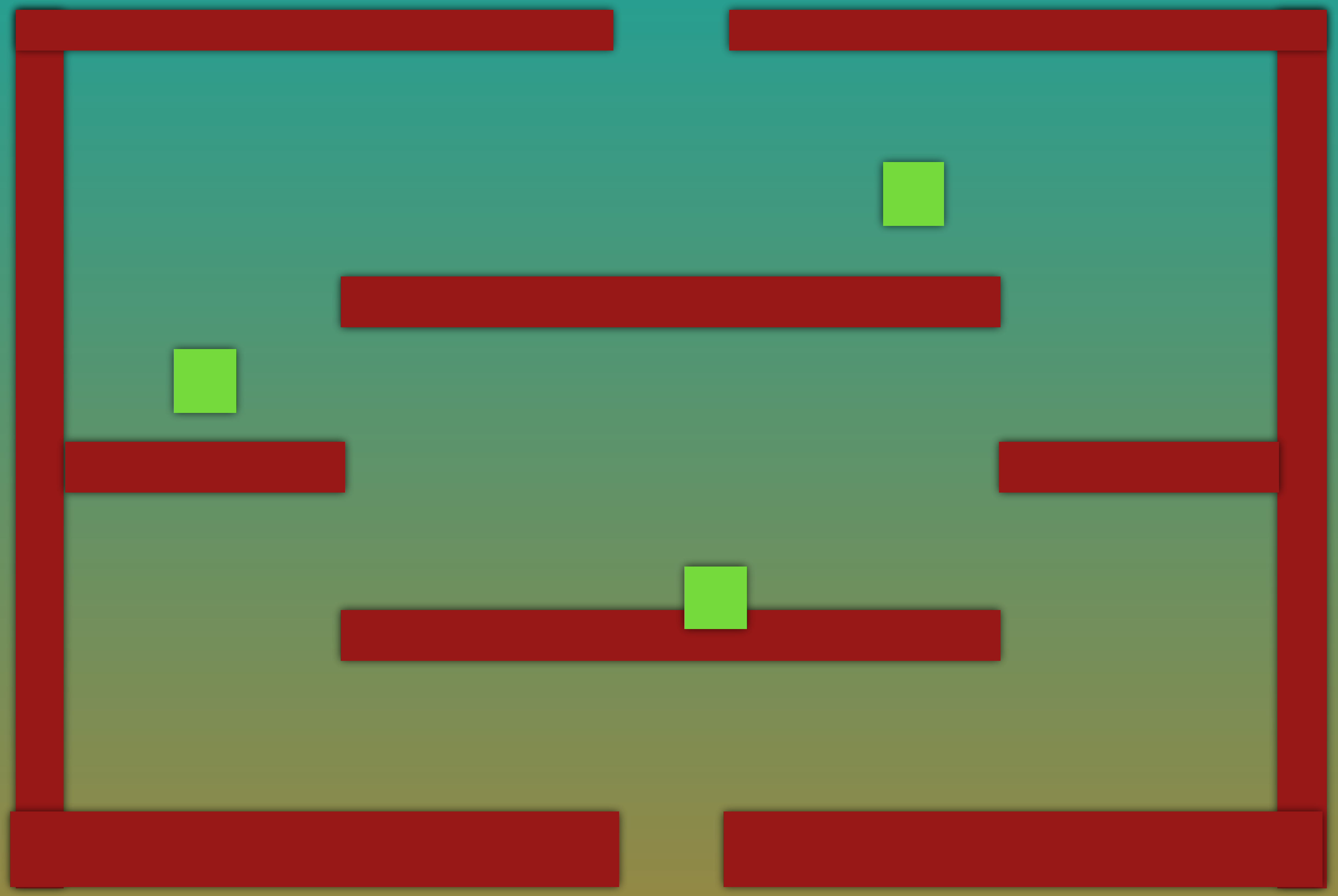
Physics steps.



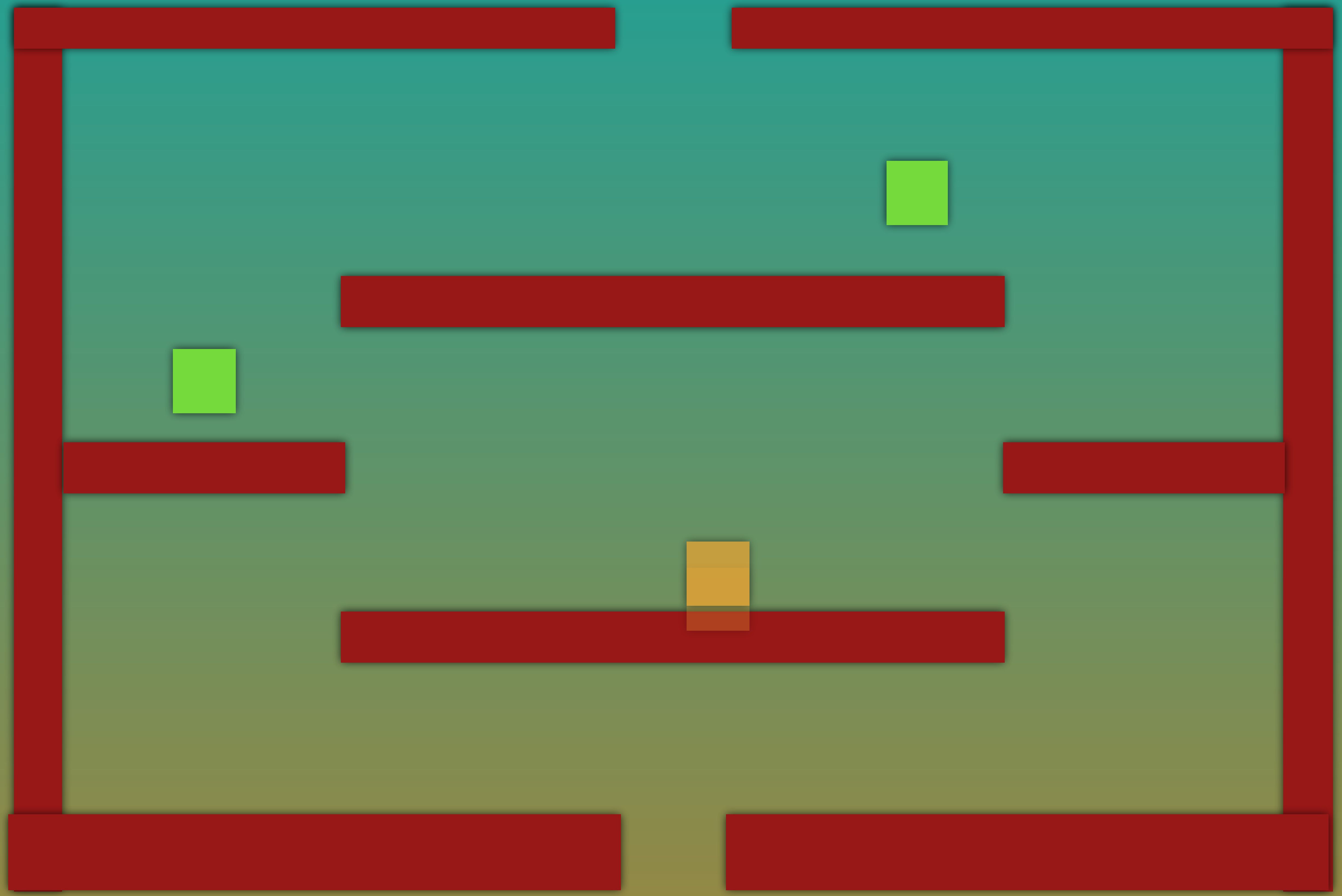
1. Apply acceleration and friction to dynamic entities.



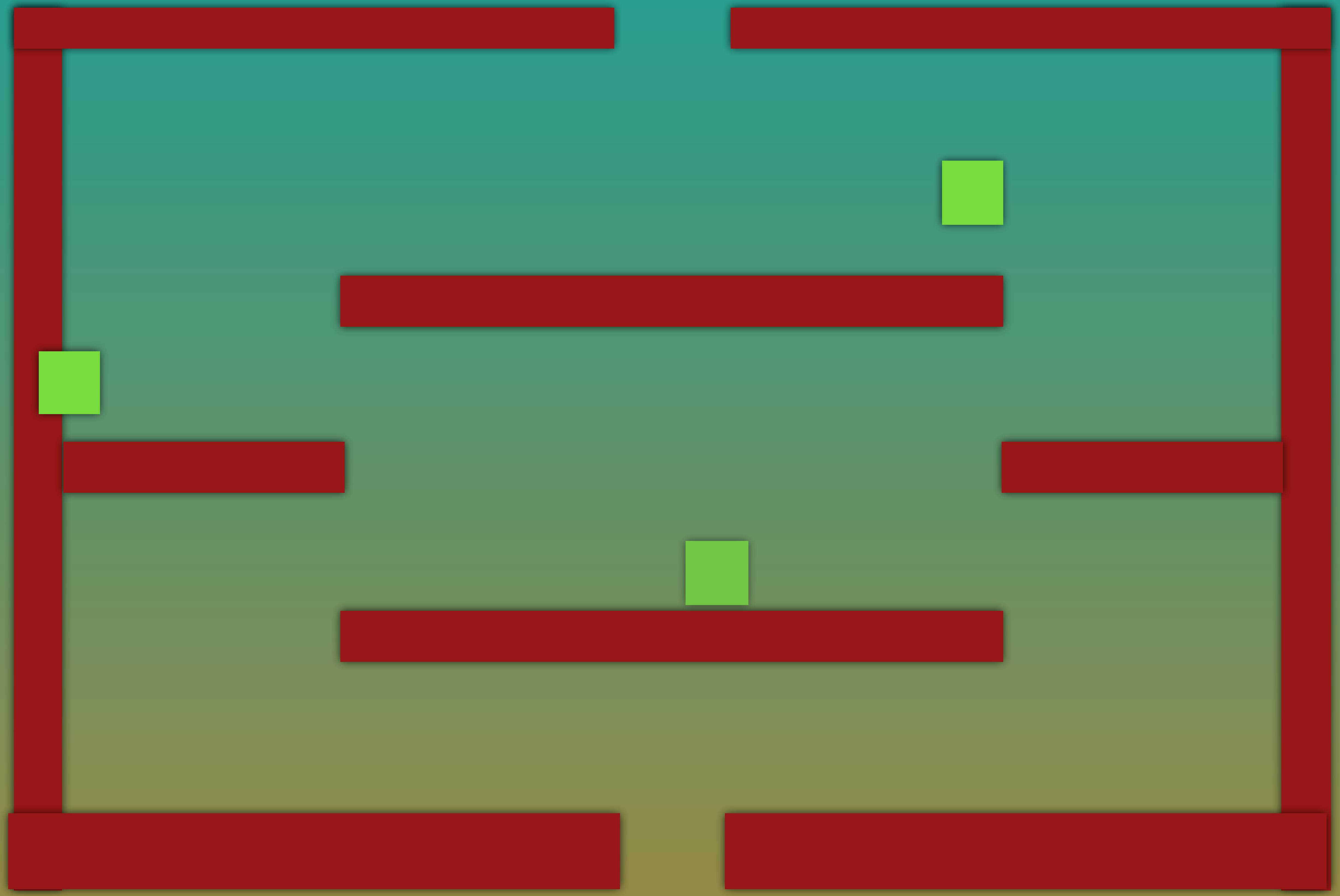
1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.



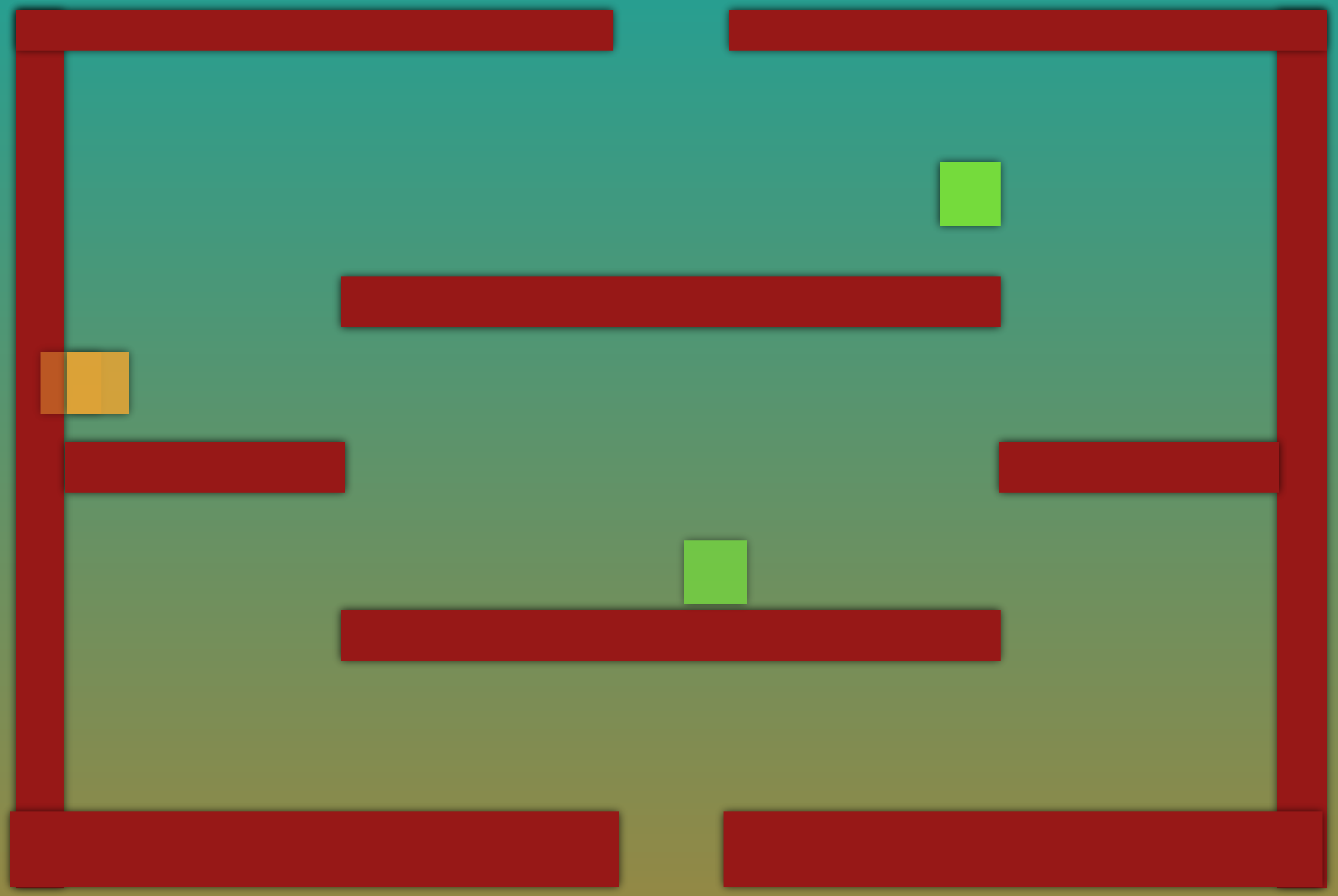
1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.
3. Apply Y-axis velocity to dynamic entity positions.



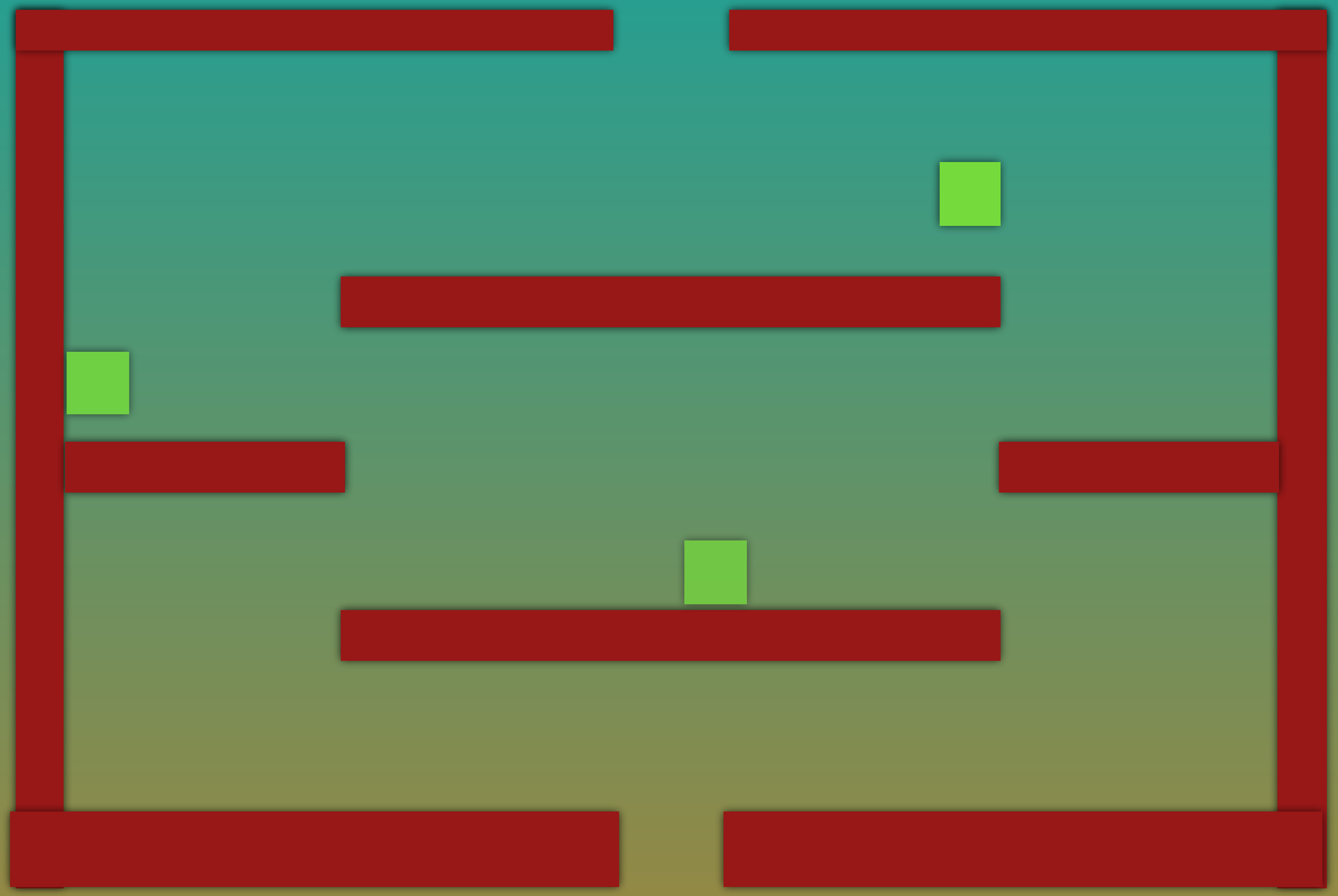
1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.
3. Apply Y-axis velocity to dynamic entity positions.
4. For each dynamic entity, check collisions with static entities and adjust Y-position to not collide. Set y-velocity to 0.



1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.
3. Apply Y-axis velocity to dynamic entity positions.
4. For each dynamic entity, check collisions with static entities and adjust Y-position to not collide. Set y-velocity to 0.
5. Apply X-axis velocity to dynamic entity positions.



1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.
3. Apply Y-axis velocity to dynamic entity positions.
4. For each dynamic entity, check collisions with static entities and adjust Y-position to not collide. Set y-velocity to 0.
5. Apply X-axis velocity to dynamic entity positions.
6. For each dynamic entity, check collisions with static entities and adjust X-position to not collide. Set x-velocity to 0.

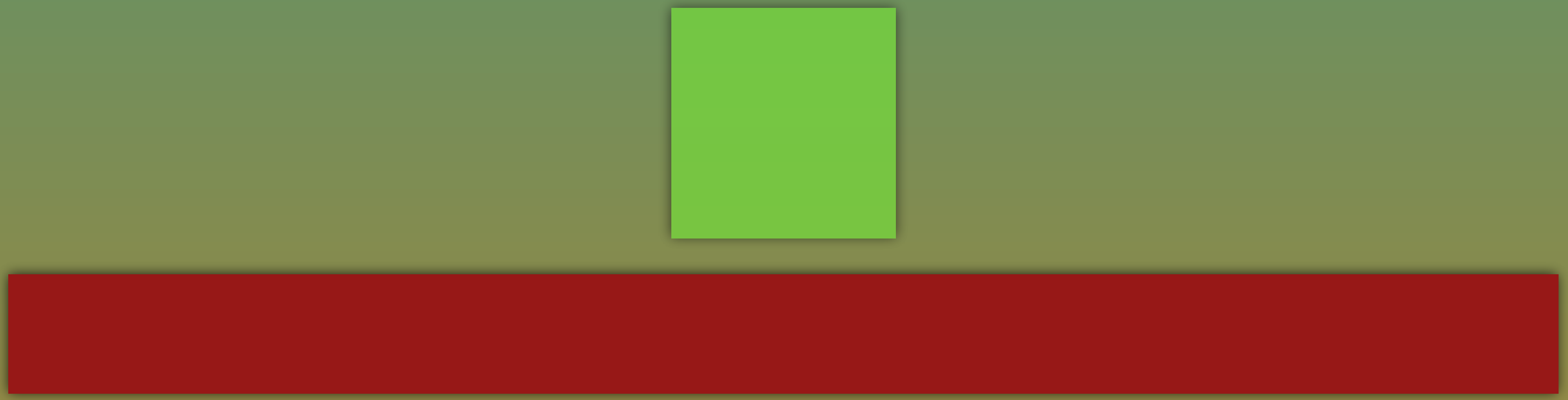


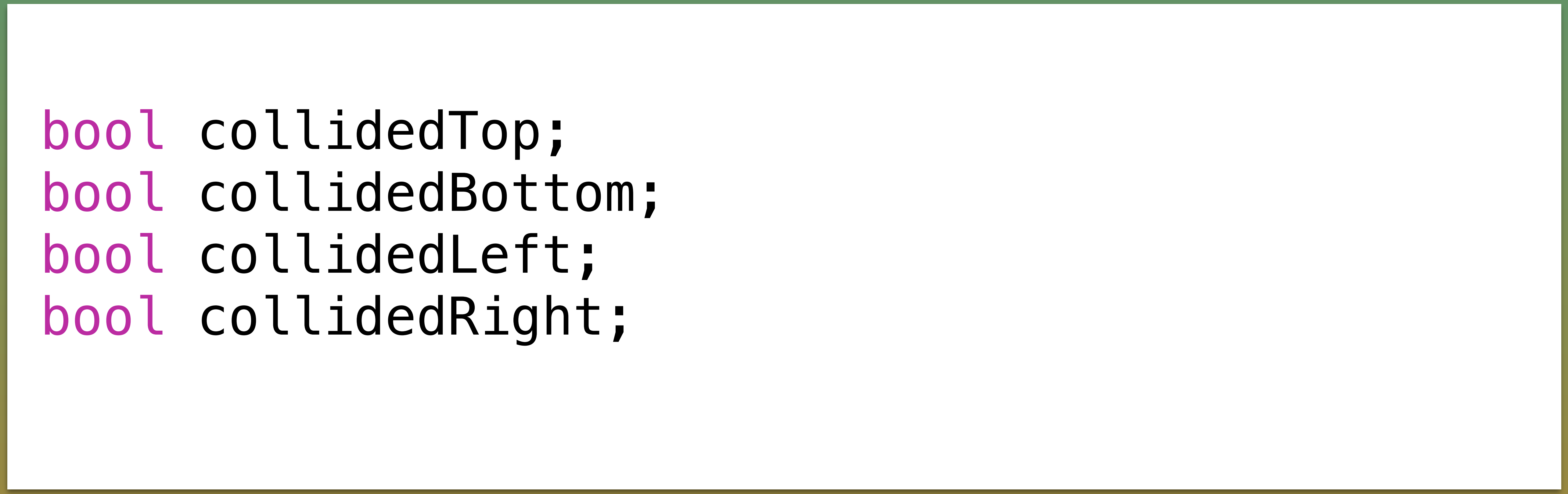
1. Apply acceleration and friction to dynamic entities.
2. Apply gravity to dynamic entities.
3. Apply Y-axis velocity to dynamic entity positions.
4. For each dynamic entity, check collisions with static entities and adjust Y-position to not collide. Set y-velocity to 0.
5. Apply X-axis velocity to dynamic entity positions.
6. For each dynamic entity, check collisions with static entities and adjust X-position to not collide. Set x-velocity to 0.

Contact detection.

Contact flags.

4 boolean flags, one for each side.

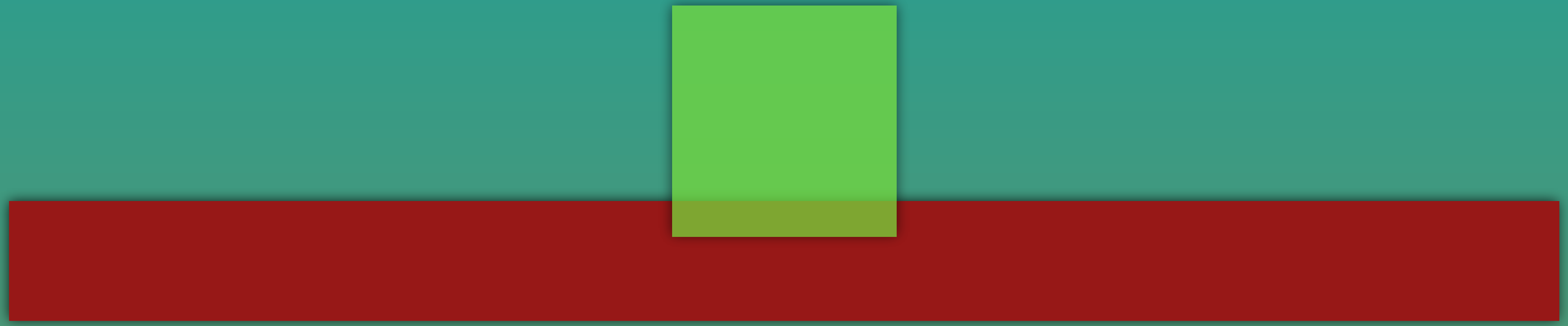




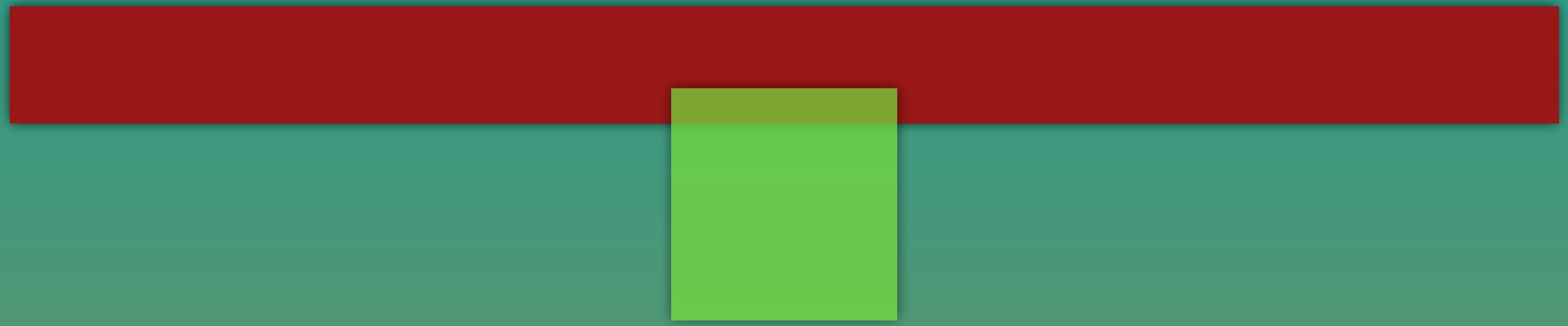
```
bool collidedTop;
bool collidedBottom;
bool collidedLeft;
bool collidedRight;
```

Clear on every frame

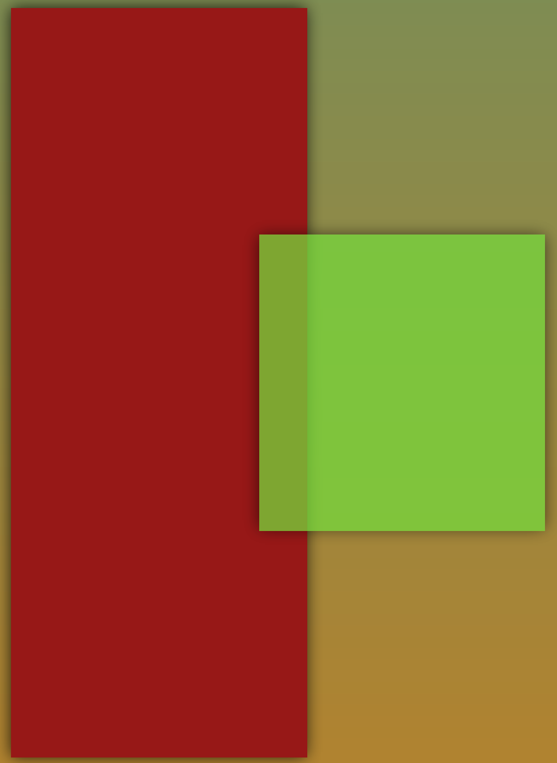
```
collidedTop = false;
collidedBottom = false;
collidedLeft = false;
collidedRight = false;
```



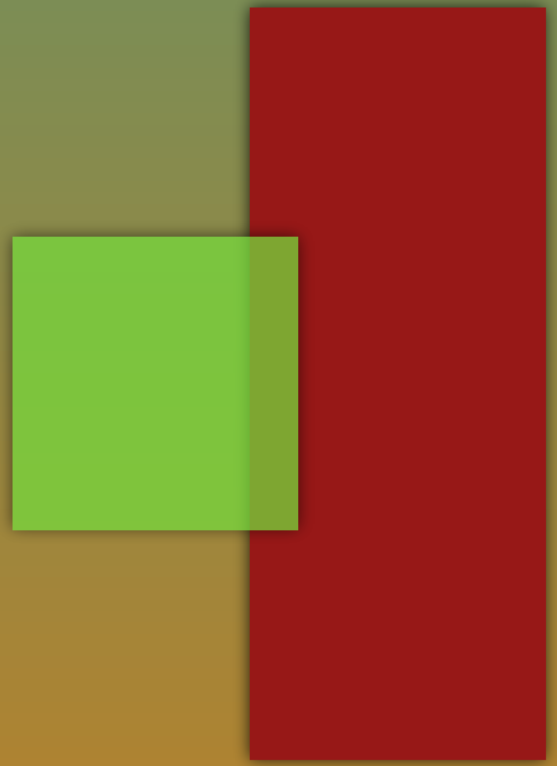
```
collidedBottom = true;
```



```
collidedTop = true;
```



```
collidedLeft = true;
```



```
collidedRight = true;
```

```
class Entity {
    public:

        Entity();
        void Update(float elapsed);
        void Render();

        bool collidesWith(Entity *entity);

        void FixedUpdate();

        SheetSprite sprite;
        float x;
        float y;

        float width;
        float height;
        float velocity_x;
        float velocity_y;
        float acceleration_x;
        float acceleration_y;
        float friction_x;
        float friction_y;
        float mass;
        bool isStatic;

        bool enableCollisions;

        bool collidedTop;
        bool collidedBottom;
        bool collidedLeft;
        bool collidedRight;
};
```

Let’s look at some code!

Home assignment…

Make a simple single screen platformer!

- Must use velocity, acceleration, friction and gravity.
- Must use fixed time step.
- Player must be able to jump, but only when standing on the ground.
- Player must interact with enemies, another player or collect items.
- Can use graphics or simple shapes.

- Look at Super Crate Box, Towerfall, Samurai Gunn, Bubble Bobble, Donkey Kong.