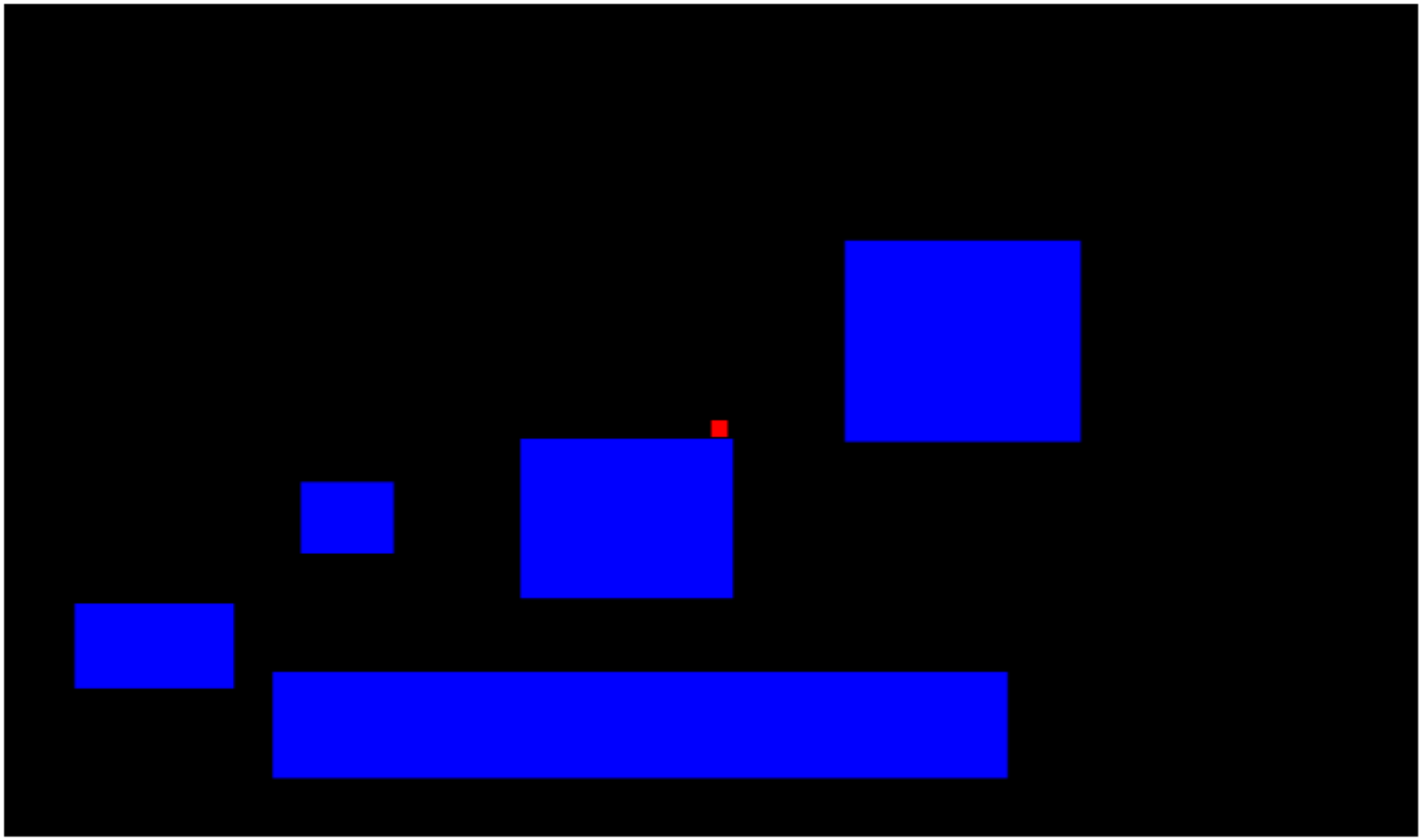
UW GAME DEV CLUB

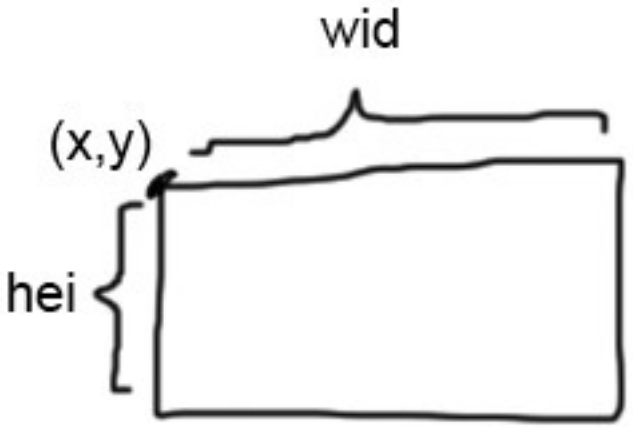
Making a simple platformer with physics

Using stuff from last time (loading levels, player-following camera)



The core: rect-rect hit detection

We have two rectangles defined by {x,y,wid,hei}.



How can you tell if they're intersecting?

UW GAME DEV CLUB

The core: rect-rect hit detection

```
//given two objects with fields {x,y,wid,hei}, returns if the two rectangles are touching
function hitrect_touch(a,b) {
	if (a.wid == 0 || a.hei == 0 || b.wid == 0 || b.hei == 0) return false;
	var r1x1 = a.x;
	var r1x2 = a.x + a.wid;
	var r1y1 = a.y;
	var r1y2 = a.y + a.hei;

	var r2x1 = b.x;
	var r2x2 = b.x + b.wid;
	var r2y1 = b.y;
	var r2y2 = b.y + b.hei;
	return !(r1x1 > r2x2 || r2x1 > r1x2 || r1y1 > r2y2 || r2y1 > r1y2);
}
```

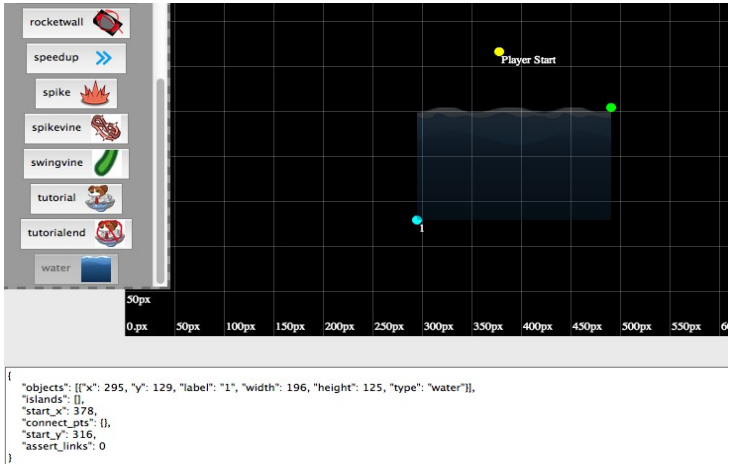
```
hitrect_touch({x:0,y:0,wid:100,hei:100},{x:10,y:10,wid:50,hei:50}) //true

hitrect_touch({x:0,y:0,wid:100,hei:100},{x:-10,y:-10,wid:5,hei:5}) //false
```

UW GAME DEV CLUB

Setting up our world

Generate our worlds using the same “SpeedyPups” tool, http://spotcos.com/gogodoggy/editor.html




```
{
    "objects": [
        {"x": 295, "y": 129, "label": "1", "width": 196, "height": 125, "type": "water"}
    ],
    "islands": [],
    "start_x": 378,
    "connect_pts": {},
    "start_y": 316,
    "assert_links": 0
}
```

*w+click to set player spawn position,

*Select water then click+escape to drop a green point. q+click to the bottom-left of the green point to make a square

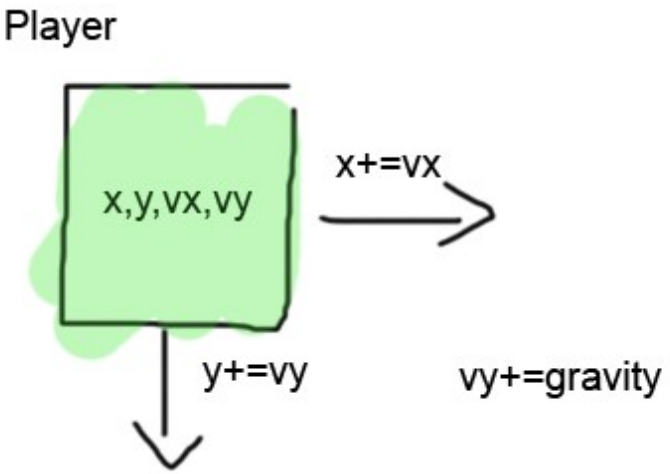
Since the y coords in the editor are flipped, load like this:

```
_player.x = LEVEL.start_x;
_player.y = -LEVEL.start_y;

for (var i_obj = 0; i_obj < LEVEL.objects.length; i_obj++) {
      var itr_obj = LEVEL.objects[i_obj];
      _gameobjs.push({
            x : itr_obj.x,
            y : -itr_obj.y-itr_obj.height,
            wid : itr_obj.width,
            hei : itr_obj.height
      });
}
```

UW GAME DEV CLUB

Platformer (the concept)

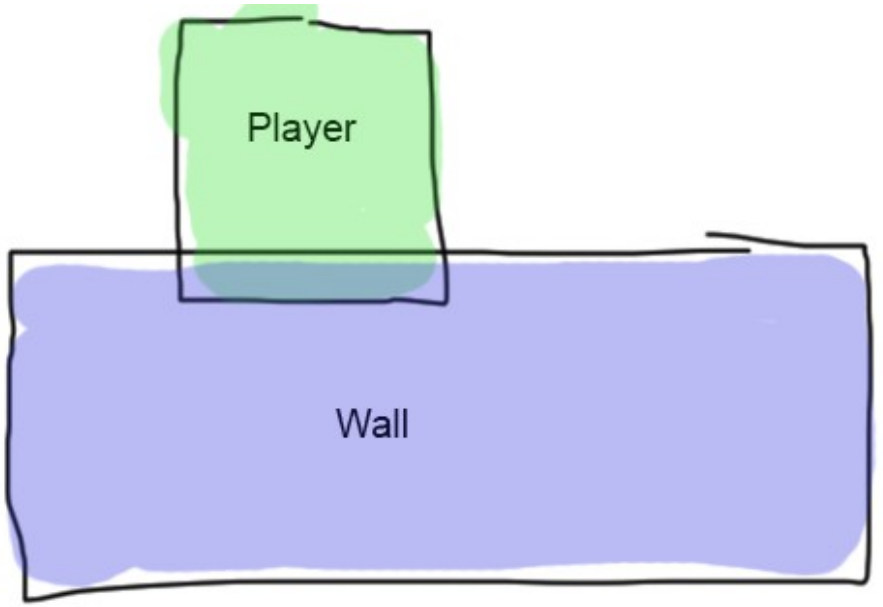


Have a player with x,y,vx,vy fields.

On update,
Update x by vx
Update y by vy
Update vy by gravity (constant)

If the player intersects a wall (using the rect-rect intersection)...

(it gets a little hairy from here)



UW GAME DEV CLUB

Platformer logic: in the air

We want to be able to know if we're on the ground
(only can jump when on the ground)

Use this global variable:

```
var _on_ground = false;
```

And in our update, two conditions:

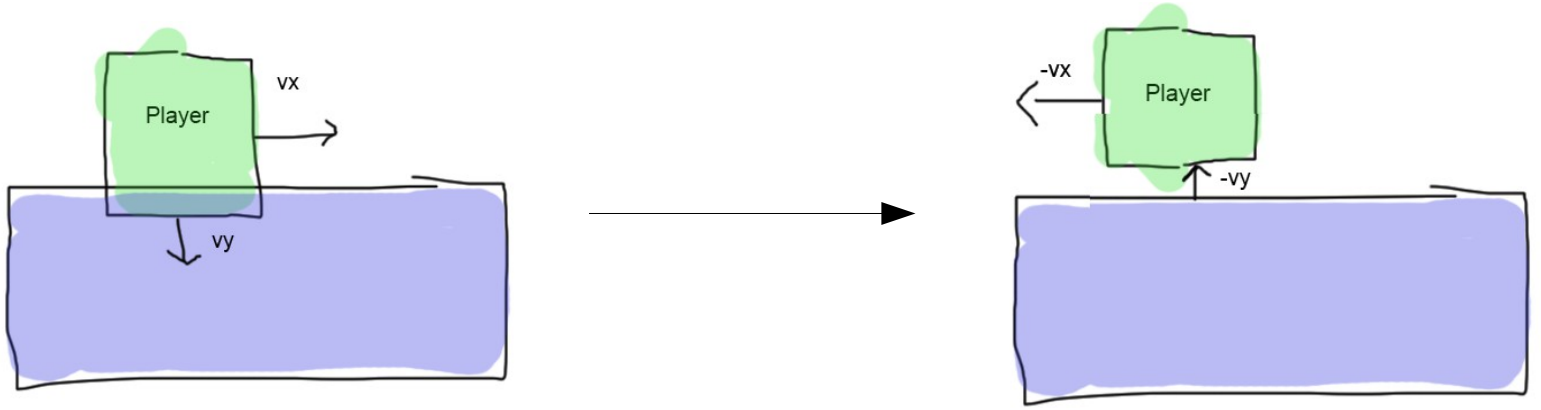
```
function update()

    if (_on_ground) {
        //...

    } else {
        _player.vy += GRAVITY;
        _player.x += _player.vx;
        _player.y += _player.vy;

        for (wall in all_walls) {
            if (hitrect_touch(player,wall)) {
                //...
            }
        }
    }
```

Platformer logic: in the air



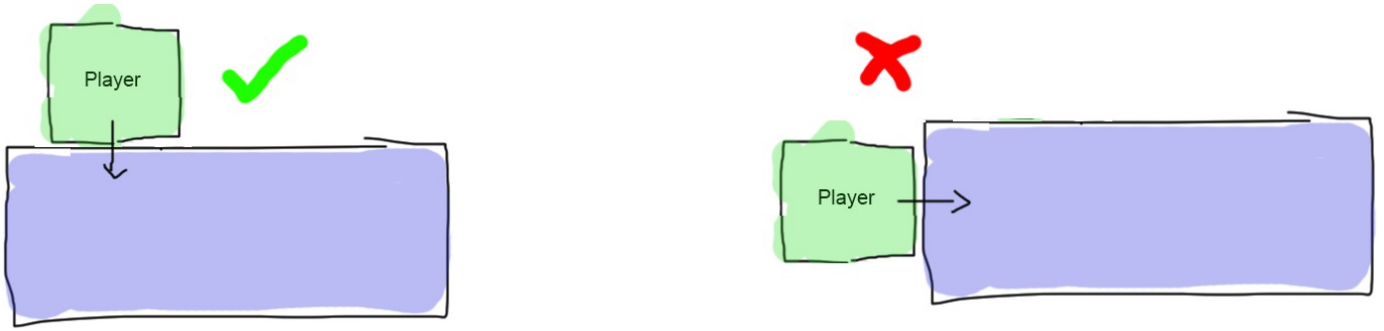
If the player is colliding with a wall, move him out in the reverse direction (-vx,-vy)
Until he's no longer colliding.

```
if (hitrect_touch(_player,itr_obj)) {
      while (hitrect_touch(_player,itr_obj)) {
            _player.x -= signum(_player.vx);
            _player.y -= signum(_player.vy);
      }

      //...
}
```

Note signum is: `number != 0 ? (number < 0 ? -1 : 1) : 0;`
(We want to move him out in increments of 1)

Platformer logic: in the air



If the player is directly above the wall (that he just moved out of), we're on the ground.

Enter ground mode.

```
_player.y += 1;

if (hitrect_touch(_player,itr_obj)) {
      _on_ground = true;
}

_player.y -= 1;
```

UW GAME DEV CLUB

Platformer logic: on the ground

```
function update()

	if (_on_ground) {
		_player.x += _player.vx;
		_player.vy = 0;

		_player.y += 1; //test if under the player is a wall
			var did_hit = false;
			for (var i_obj = 0; i_obj < _gameobjs.length; i_obj++) {
				var itr_obj = _gameobjs[i_obj];
				if (hitrect_touch(_player,itr_obj)) {
					did_hit = true;
					break;
				}
			}
			if (!did_hit) { //if there wasn't a wall, not on the ground anymore
				_on_ground = false;
			}
		_player.y -= 1; //unset the test

	} else {
		//...
	}
```

-Move the player in the x direction (but not in y)

-If there is NOT a wall underneath the player (test in the same way as before), You're in the air now

UW GAME DEV CLUB

Misc

```
if (KEYS_DOWN[KEYBOARD.RIGHT]) {
      _player.vx = 10;

} else if (KEYS_DOWN[KEYBOARD.LEFT]) {
      _player.vx = -10;

} else {
      _player.vx *= 0.3;
      if (_player.vx < 0.1) _player.vx = 0;
}
```

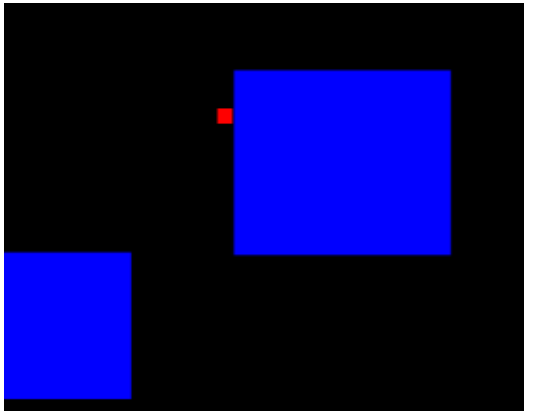
To get the most “natural” movement, add a friction slowdown when no keys are pressed (as opposed to immediately stopping)

```
if (KEYS_DOWN[KEYBOARD.SPACE] && _on_ground) {
      _player.vy = -15;
      _on_ground = false;
}
```

Add this to jump
(what would happen if we didn't put in the change to air mode?)

UW GAME DEV CLUB

Some (unsolved) problems



You can climb up walls (bug or feature?)

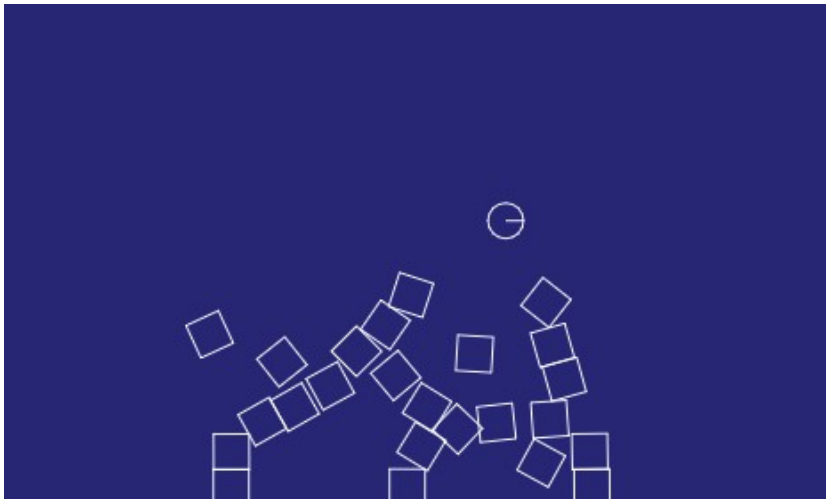
If you land while moving fast (in the x direction), you'll get a little glitch-hiccup. Why is this?

Are there any conditions that could cause this physics engine to infinite loop? (There are)

How well do you think this engine works if the walls move around? (surprisingly well, though a couple of edge cases)

UW GAME DEV CLUB

Further resources



Want more complicated physics interactions (without having to code them yourself?)

http://box2d-js.sourceforge.net/index2.html

(this is what angry birds uses)