

Code User Story
Roel

Tutorial gevolgd en juiste gravity gemaakt

Tutorial volgen voor het plaatsen van platforms

in lijst Done

LEDEN

D

JK

R

+

2 dagen

+

Labels

Omschrijving

Bewerken

Iedereen moet de video's kijken dus 3-4 uur pp

Episodes

Verberg voltooide items

Verwijderen

58%

☒

1

☒

2

☒

3

☒

4

☒

5

☒

6

☒

7

☐

8

☐

9

☐

10

☐

12

☐

12

Een item toevoegen

Activiteit

Details weergeven

TOEVOEGEN AAN KAART

Leden

Labels

Checklist

Vervaldatum

Bijlage

Omslag

POWER-UPS

+ Power-Ups toevoe...

Upgrade voor een onbeperkt aantal Power-Ups per bord.

Upgrade team

BUTLER

NIEUW

+ Knop Kaart toevo...

ACTIES

→ Verplaats

Kopiëren

Sjabloon maken

Volgen

Archiveren

Delen

Gravity en jump speed optimaliseren

in lijst Done

LEDEN

D

JK

R

+

Omschrijving

Bewerken

Jump speed en Gravity goed instellen, zodat het natuurlijk aanvoelt



Code character.cpp

```
1 #include "Character.hpp" //Character.cpp
2
3 #include "DEFINITIONS.hpp"
4
5 namespace engine{
6
7     //Definitions for Character
8     Character::Character(const sf::Vector2f &position, const sf::Texture &texture, const sf::Texture &texture_flip) :
9         position(position),
10         texture(texture),
11         texture_flip(texture_flip)
12     {
13         sprite.setTexture(texture);
14         sprite.setPosition(position);
15         width = sprite.getGlobalBounds().width;
16         height = sprite.getGlobalBounds().height;
17     }
18
19     Character::Character():
20         position(0,0)
21     {}
22
23     void Character::flipTexture() {
24         if(flip){
25             sprite.setTexture(texture);
26         }else{
27             sprite.setTexture(texture_flip);
28         }
29         flip = !flip;
30     }
31
32     void Character::setTexture(sf::Texture &texture_, sf::Texture &texture_flip_) {
33         texture = texture_;
34         texture_flip = texture_flip_;
35         sprite.setTexture(texture);
36         width = sprite.getGlobalBounds().width;
37         height = sprite.getGlobalBounds().height;
38     }
39
40     void Character::setPosition(const sf::Vector2f &position_) {
41         position = position_;
42         sprite.setPosition(position);
43     }
44
45     void Character::setPosition(const float &x, const float &y) {
46         position.x = x;
47         position.y = y;
48         sprite.setPosition(position);
49     }
50
51     sf::Vector2f Character::getPosition(){
52         return position;
53     }
54
55     void Character::draw(sf::RenderWindow& renderWindow) {
56         renderWindow.draw(sprite);
57     }
58
59     void Character::move(sf::Vector2f movement) {
60         position += movement;
61         if(movement.x < 0 && !flip){
62             flipTexture();
63         }else if(movement.x > 0 && flip){
64             flipTexture();
65         }
66         sprite.setPosition(position);
67     }
68
69     void Character::move(const float &x, const float &y) {
70         position.x += x;
71         position.y += y;
72         sprite.setPosition(position);
73     }
74
75     sf::Vector2f Character::nextPosition(sf::Vector2f movement) {
76         return position + movement;
77     }
78
79     sf::Sprite Character::nextSprite(sf::Vector2f movement) {
80         sf::Sprite temp;
81         temp.setTextureRect(sprite.getTextureRect());
82         temp.setPosition(position + movement);
83         return temp;
84     }
85
86     sf::Sprite& Character::getSprite() {
87         return sprite;
88     }
89
90     bool Character::objectCollisionAndFalling(const sf::Sprite &object, const float& dt) {
91         //Als de volgende move van de player in het object zit
92
93         sf::Sprite temp = nextSprite(velocity * dt);
94
95         if(temp.getGlobalBounds().intersects(object.getGlobalBounds())){
96
97             //rechts
98             if(getPosition().x> object.getPosition().x + object.getGlobalBounds().width){
99                 velocity.x = 0;
100             }
101             //links
102             else if(getPosition().x + width < object.getPosition().x ){
103                 velocity.x = 0;
104             }
105             //onderkant
106             else if(getPosition().y > object.getPosition().y + object.getGlobalBounds().height){
107                 velocity.y = 0;
108             }
109             //bovenkant
110             else{
111                 velocity.y = 0;
112                 jump = 0;
113                 on_ground = true;
114             }
115             return true;
116             //Als de player niet het object raakt
117             else{
118                 return false;
119             }
120             //delete temp;
121         }
122     }
123
124     void Character::updateVelocity(const float &dt) {
125         //Velocity update
126         if(velocity != sf::Vector2f{0,0}){
127             move(velocity * dt);
128         }
129     }
130
131     void Character::respawn(sf::Vector2f spawn) {
132         setPosition(spawn);
133         velocity.x = 0;
134     }
135
136 }
```





Code character.hpp

```
1 #pragma once //Character.hpp
2
3 #include <SFML/Graphics.hpp>
4
5 namespace engine{
6
7     class Character{
8     private:
9         sf::Sprite sprite;
10        sf::Vector2f position;
11        sf::Texture texture;
12        sf::Texture texture_flip;
13
14        bool flip = false;
15
16        void flipTexture();
17
18    public:
19        sf::Vector2f velocity = {0,0};
20        float speed = 400.0f;
21        float jump_speed = 600.0f;
22        float slow_down = 0.0f, slow_down_ground = 20.0f, slow_down_air = 5.0f;
23        float speed_up = 0.0f, speed_up_ground = 40.0f, speed_up_air = 30.0f;
24        int height, width, max_jump = 2, jump = 0;
25        bool jump_done = false, on_ground = false;
26
27        Character(const sf::Vector2f &position, const sf::Texture &texture, const sf::Texture &texture_flip);
28
29        Character();
30
31        void setTexture(sf::Texture& texture_, sf::Texture &texture_flip_);
32
33        void setPosition(const sf::Vector2f& position_);
34
35        void setPosition(const float& x, const float& y);
36
37        sf::Vector2f getPosition();
38
39        void draw(sf::RenderWindow& renderWindow);
40
41        void move(sf::Vector2f movement);
42
43        void move(const float& x, const float& y);
44
45        sf::Vector2f nextPosition(sf::Vector2f movement);
46
47        sf::Sprite nextSprite(sf::Vector2f movement);
48
49        sf::Sprite& getSprite();
50
51        bool objectCollisionAndFalling(const sf::Sprite &object, const float& dt);
52
53        void updateVelocity(const float& dt);
54
55        void respawn(sf::Vector2f spawn);
56    };
57
58 }
59
```

Platforms gemaakt

 **platforms**
in lijst [Done](#) 

LEDEN LABELS

Omschrijving

Voeg een meer gedetailleerde beschrijving toe...

Checklist

Verberg voltooide items

Verwijderen

100%

- ☒ ~~Class voor platforms~~
- ☒ ~~Collision op x-as~~
- ☒ ~~Collision op y-as~~

Een item toevoegen

Activiteit

Details weergeven

 Maak een opmerking...

 **daankerbush** gisteren om 12:08

Probleem bij collision als ze allemaal achter elkaar staan

 - Beantwoord

Code

```
Platform.cpp
Platform.hpp
+

1 #include "Platform.hpp"
2
3 namespace engine
4 {
5     Platform::Platform(GameDataRef data) : _data(data){}
6
7     void Platform::addPlatform(sf::Texture& texture, const sf::Vector2f& position) {
8         sf::Sprite platform;
9         platform.setTexture(texture);
10        platform.setPosition(position);
11        platformSprites.push_back(platform);
12    }
13
14    std::vector<sf::Sprite>& Platform::getPlatforms() {
15        return platformSprites;
16    }
17
18    void Platform::draw(){
19        for (sf::Sprite platform : platformSprites){
20            _data->renderWindow.draw(platform);
21        }
22    }
23 }
```

```
Platform.cpp
Platform.hpp
X +

1 #pragma once
2
3 #include <SFML/Graphics.hpp>
4 #include "Game.hpp"
5 #include <vector>
6
7 namespace engine {
8
9     class Platform {
10     public:
11         Platform(GameDataRef data);
12
13         void addPlatform(sf::Texture& texture, const sf::Vector2f& position);
14
15         std::vector<sf::Sprite>& getPlatforms();
16
17         void draw();
18
19     private:
20         GameDataRef _data;
21         std::vector<sf::Sprite> platformSprites;
22
23     };
24
25 }
```

Extra taken

- Ik heb de achtergrond gemaakt
- En daan + joris geholpen bij het maken van testlevel