UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Teaching physics and simulations with Godot

Leszek Nowak

Teaching Assistant

Department of Information and Technologies

Jagiellonian University in Kraków

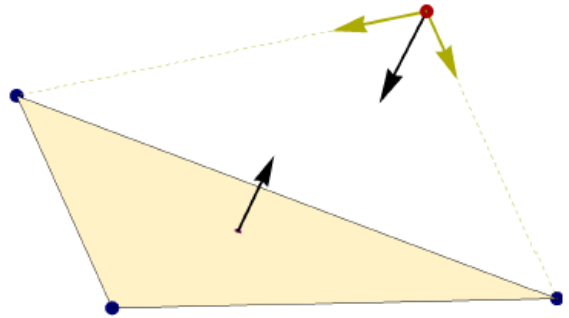
GodotCon Poznań, 10.10.2018

Teaching geometry

- We have been through this.

1. Dany jest trójkąt Δ_{ABC}:

$$\Delta_{ABC} = \begin{pmatrix} 0 & 0 & 3\sqrt{3} \\ 0 & 6\sqrt{3} & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$



(*i*) **(1 pkt)** Wyznacz normalną $\hat{\mathbf{n}}$ do trójkąta Δ_{ABC}.

Dany jest położony w $\mathbf{R} = 4\sqrt{3}\ \hat{\mathbf{n}} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ reflektor świecący światłem w kolorze $(\ 1,\ 1,\ 1\ )$, z wersorem kierunkowym snopa światła (jak na rysunku powyżej) $\hat{\mathbf{s}} = -\hat{\mathbf{n}}$, parametrem skupienia wiązki $p = 2$ i współczynnikami osłabienia $k_c = 1$, $k_l = \frac{1}{2\sqrt{6}}$ i $k_q = \frac{1}{96}$.

Mówimy, że trójkąt jest oświetlony, jeżeli oświetlony jest przynajmniej jeden z jego wierzchołków.

(*ii*) **(2 pkt.)** Sprawdź czy trójkąt Δ_{ABC} jest oświetlony przez opisany powyżej reflektor.

(*iii*) **(3 pkt.)** Wylicz kolor światła w punktach:

$$\mathbf{B} = \begin{pmatrix} 0 \\ 6\sqrt{3} \\ 0 \end{pmatrix} \text{ i } \mathbf{C} = \begin{pmatrix} 3\sqrt{3} \\ 0 \\ 3 \end{pmatrix}$$

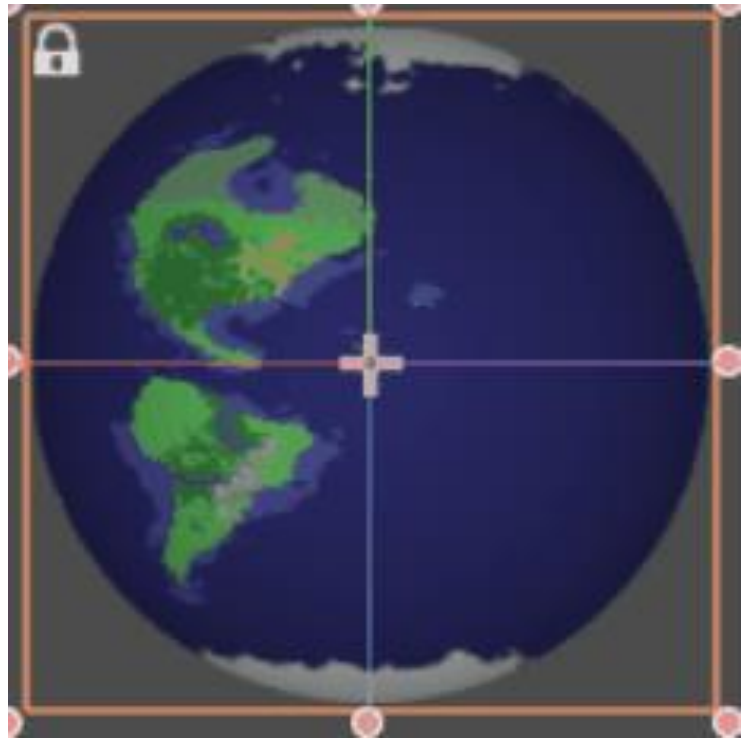
Writing your own engine

- Why you should not do it.

```
public:
    GameObject();
    ~GameObject();

    void OnExecute();
    void OnThink();

    void InitSDLWindow();

    void InitOpenglSupport();
    void InitOpenglDefines();
    void InitOpengl();

    void OpenglResizeScreen();
    #if DEBUG_MODE
        void Dumy();
    #endif
private:
    int FramePerSecond;

    bool done;

    SDL_Event event;

    Uint32 SomeTimeControlVariable;
    Uint32 thisTime;
    Uint32 lastTime;
    Uint32 deltaTime;
    Uint32 frameTime;
```

Moving to Godot

```
uniform texture continents;
float scroll = TIME * 0.05;
vec2 normal  = tex ( TEXTURE, UV ).rg - vec2( 0.5 );
vec2 cont_uv = vec2(  scroll - UV.x * 0.2, UV.y * 0.65 + 0.175 );
vec3 c = tex( continents, cont_uv - normal * 0.2 ).rgb;
COLOR.rgb  = c * ( 1.0 - length( normal ) );
```



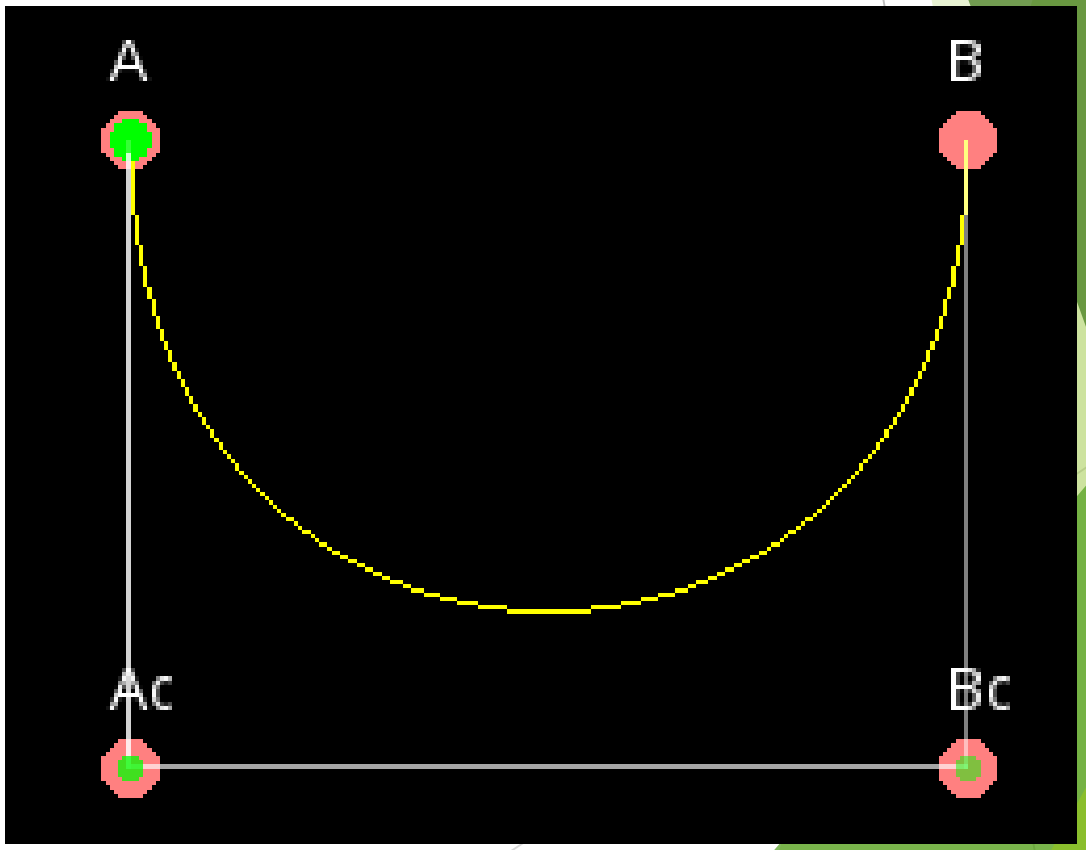
Simple examples

Learn basic vector operations by programming steering in 2d

Getting from equation to code

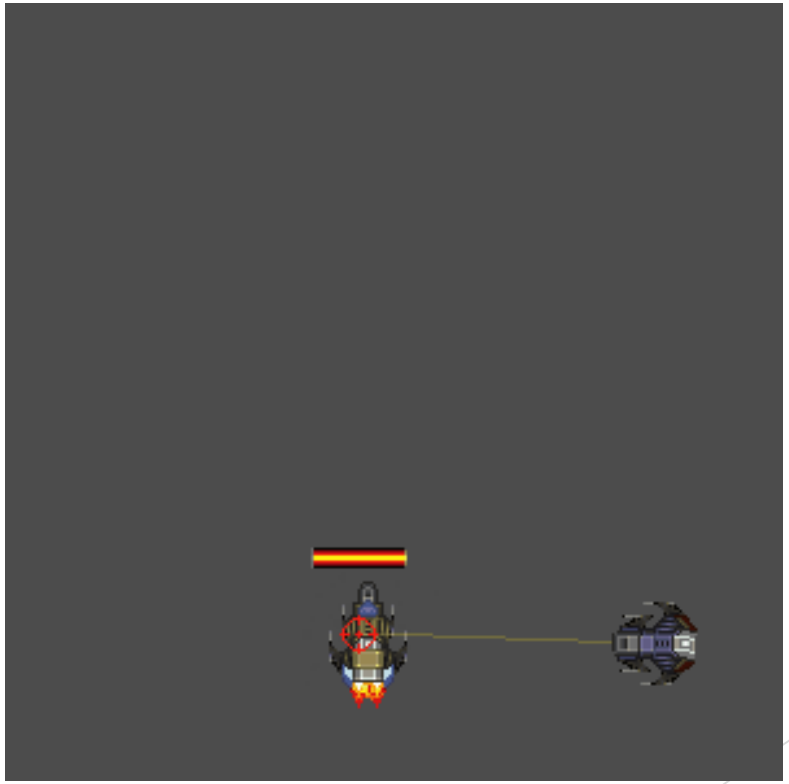
$$\mathbf{B}(t) = \sum_{k=0}^{3} B_{3,k}(t)\mathbf{P}_k = (1-t)^3\mathbf{P}_0 + 3t(1-t)^2\mathbf{P}_1 + 3t^2(1-t)\mathbf{P}_2 + t^3\mathbf{P}_3$$

```
func get_bezier_1(step, A, B ):
    var t = float(step) / STEPS
    return (1 - t) * A + t * B

func get_bezier_2(step, A, B, C):
    var t = float(step) / STEPS
    var t_1 = 1 - t
    return pow( t_1, 2 ) * A + 2 * t * t_1 * B + pow( t, 2 ) * C
```



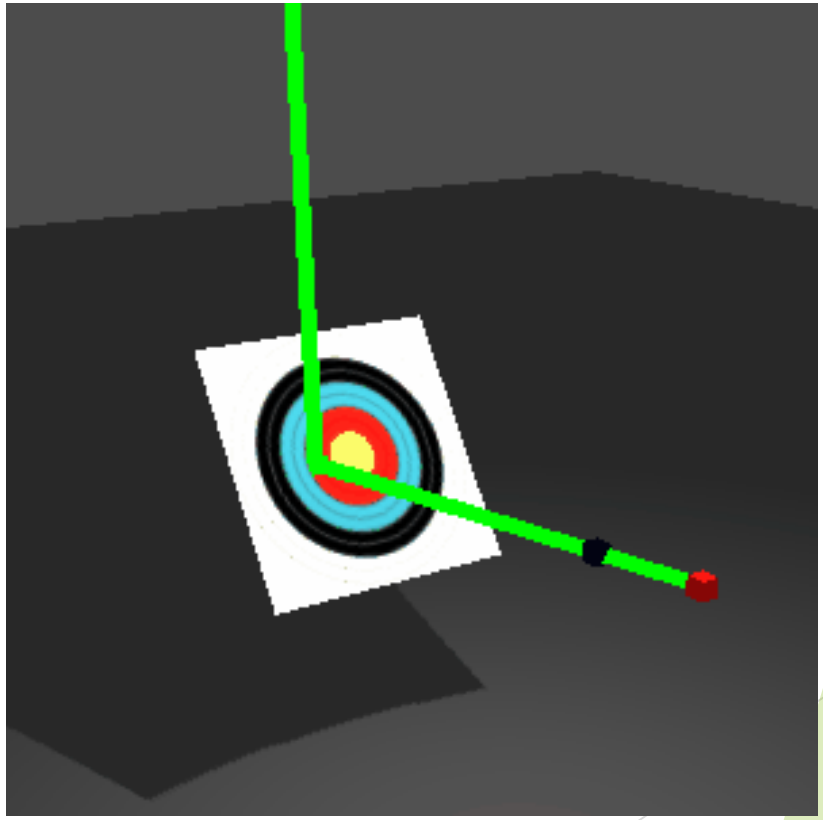
Having some fun in the process

- Solve interesting problems



And there is 3d

```
func test_triangle_hit(P1,P2,P3):
»   var surface_Normal = ( P2 - P1 ).cross( P3 - P1 ).normalized()
»   a = -surface_Normal.dot(P1)
»   t = -( surface_Normal.dot(Source) + a ) / ( surface_Normal.dot(Ray) )
»   var surface_Hit = Source + t * Ray
»   if ( (P2 - P1).cross(surface_Hit - P1) ).dot(surface_Normal) > 0:
»   »   if ( (P3 - P2).cross(surface_Hit - P2) ).dot(surface_Normal) > 0:
»   »   »   if ( (P1 - P3).cross(surface_Hit - P3) ).dot(surface_Normal) > 0:
»   »   »   »   Hit    = surface_Hit
»   »   »   »   Normal = surface_Normal
»   »   »   »   return true
»   return false
```

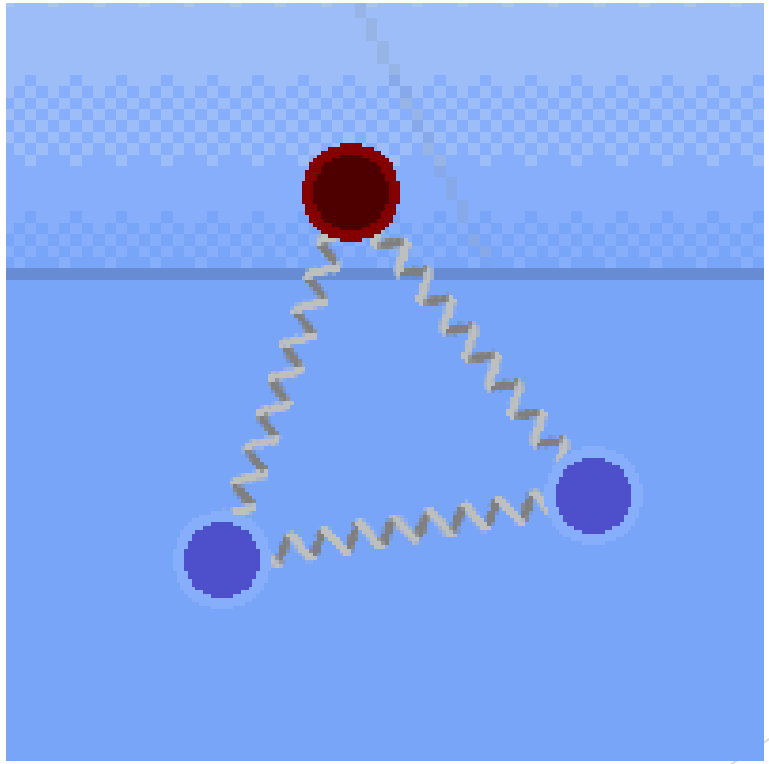


Into simulations and physics

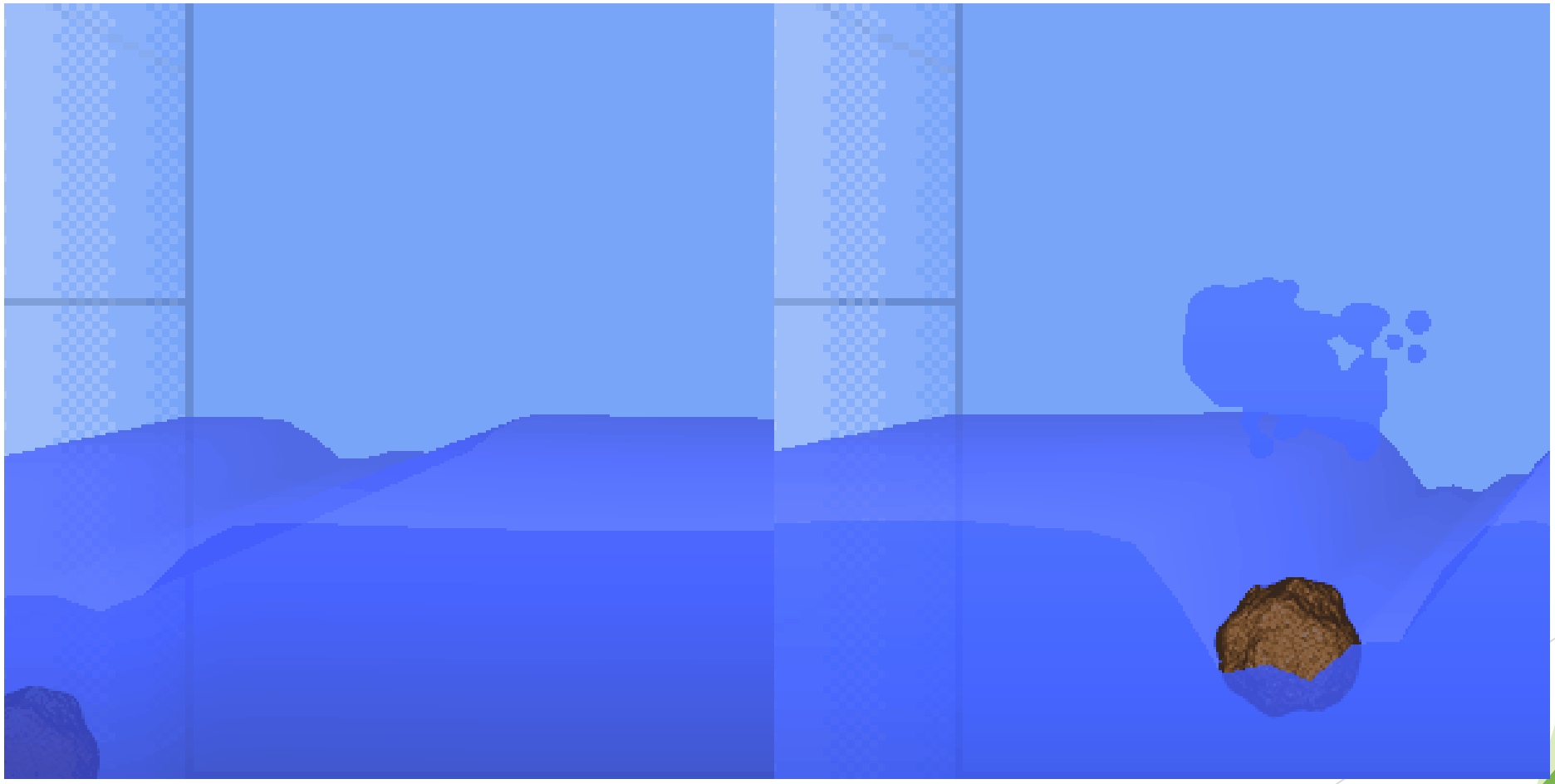
- Last year testing

Hooke’s Law

- Teaching this way proved to be fine



Wave Propagation



The project

- Length 2 years
- 5+ faculty members
- Additional teaching materials from Geometry, Physics and Simulation

The scope

Geometry

- 90 topics

e.g.:

- Matrix operations
- Principle Component Analysis
- Curves (Bezier, Catmull-Rom, Hermite, B-Splines)
- Shading (Gouraud, Phong, BRDF)
- Separate Axis Theorem

Physics and Simulation

- Over 130 topics

e.g.:

- Forces Integrators (Velocity Verlet, Leap-frog, Runge-Kutta, Midpoint)
- Flocking systems
- Collisions
- Mass-point systems
- Computational Fluid Dynamics
- Vehicle aerodynamics

What can we expect

Video for each of the topics containing:

- Lecture
- Example in Godot
- Math solution

Contact

leszek.nowak@uj.edu.pl

http://www.doit.fais.uj.edu.pl/leszek-nowak

https://github.com/johnmeadow1

https://itch.io/profile/johnmeadow