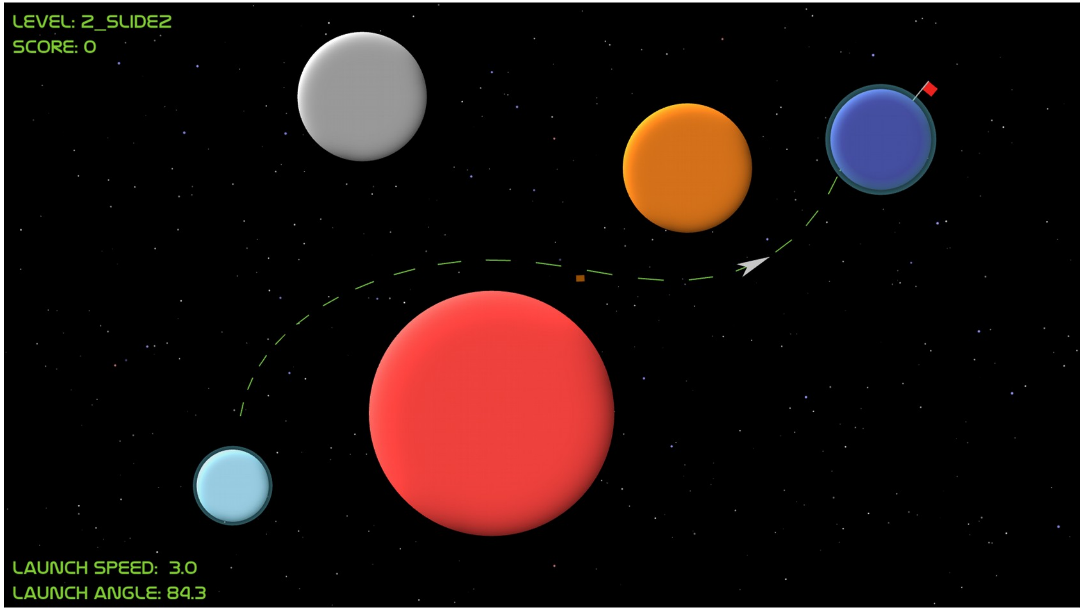
AstroFlight (2D Physics Game)

A beginner‘s practical by
Christopher Brückner

AstroFlight

- Concept of the game
- Celestial mechanics
- Rendering games
- Visualizing scalar fields
- Outlook
- Demonstration



Gravity in most videogames

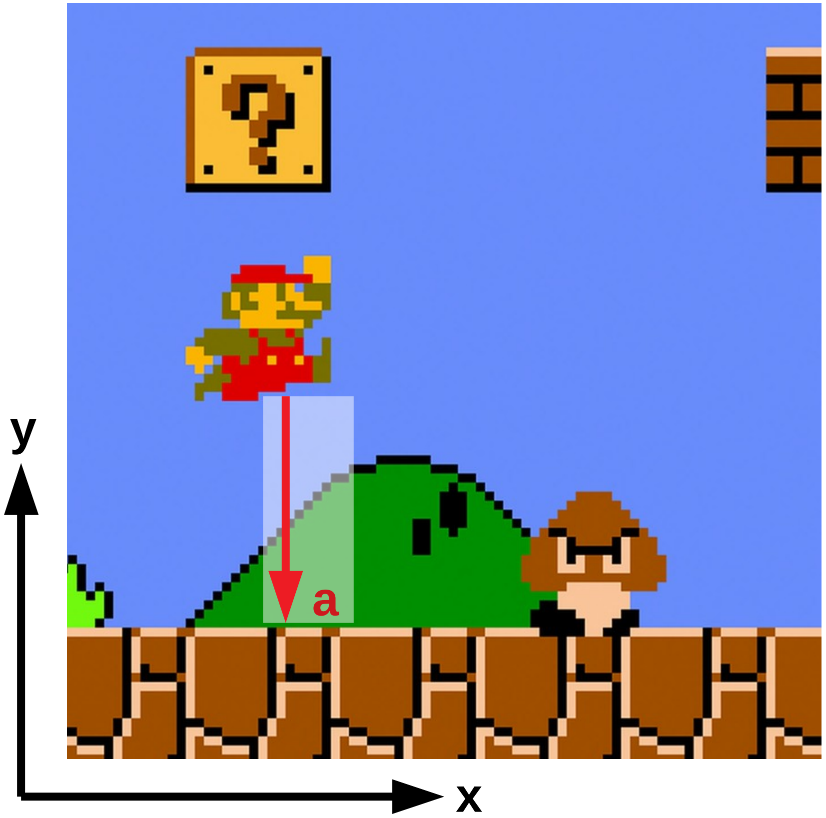


Image source:
https://emulatoronline.com/nes-games/super-mario-bros/

Gravity on a bigger scale

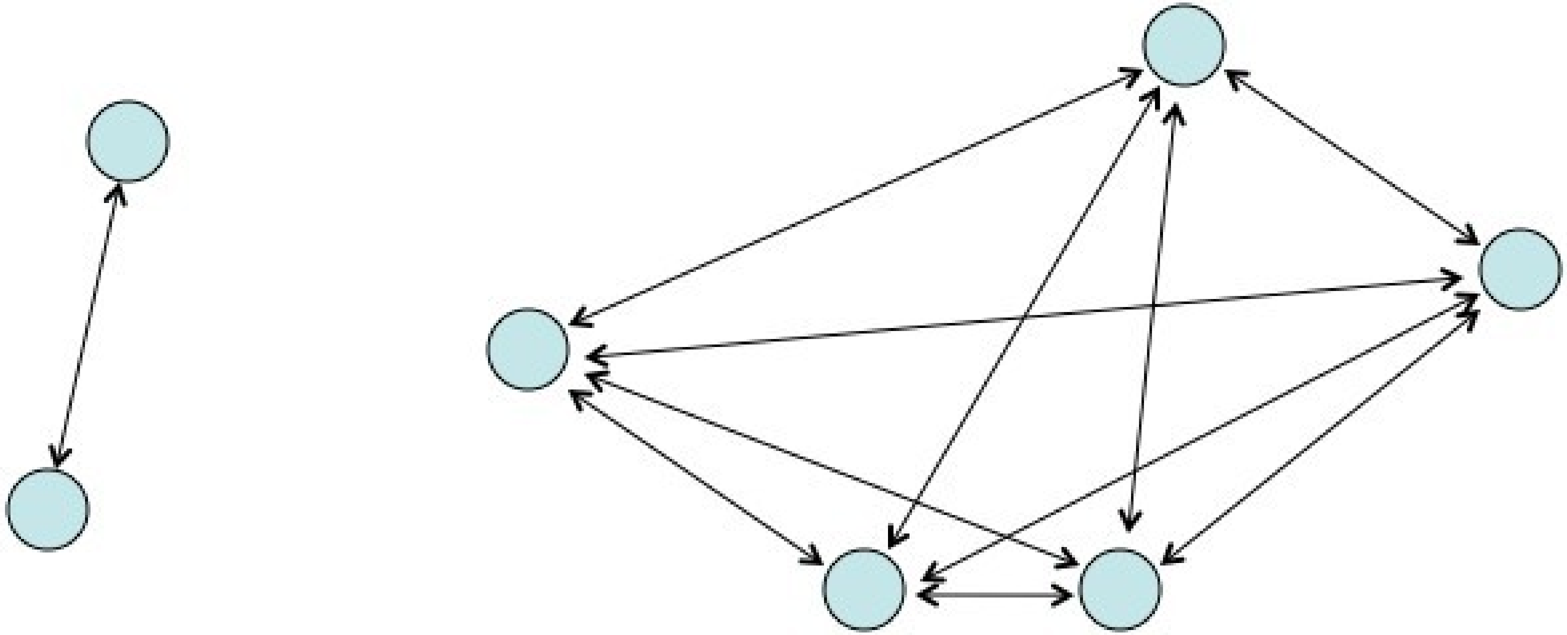
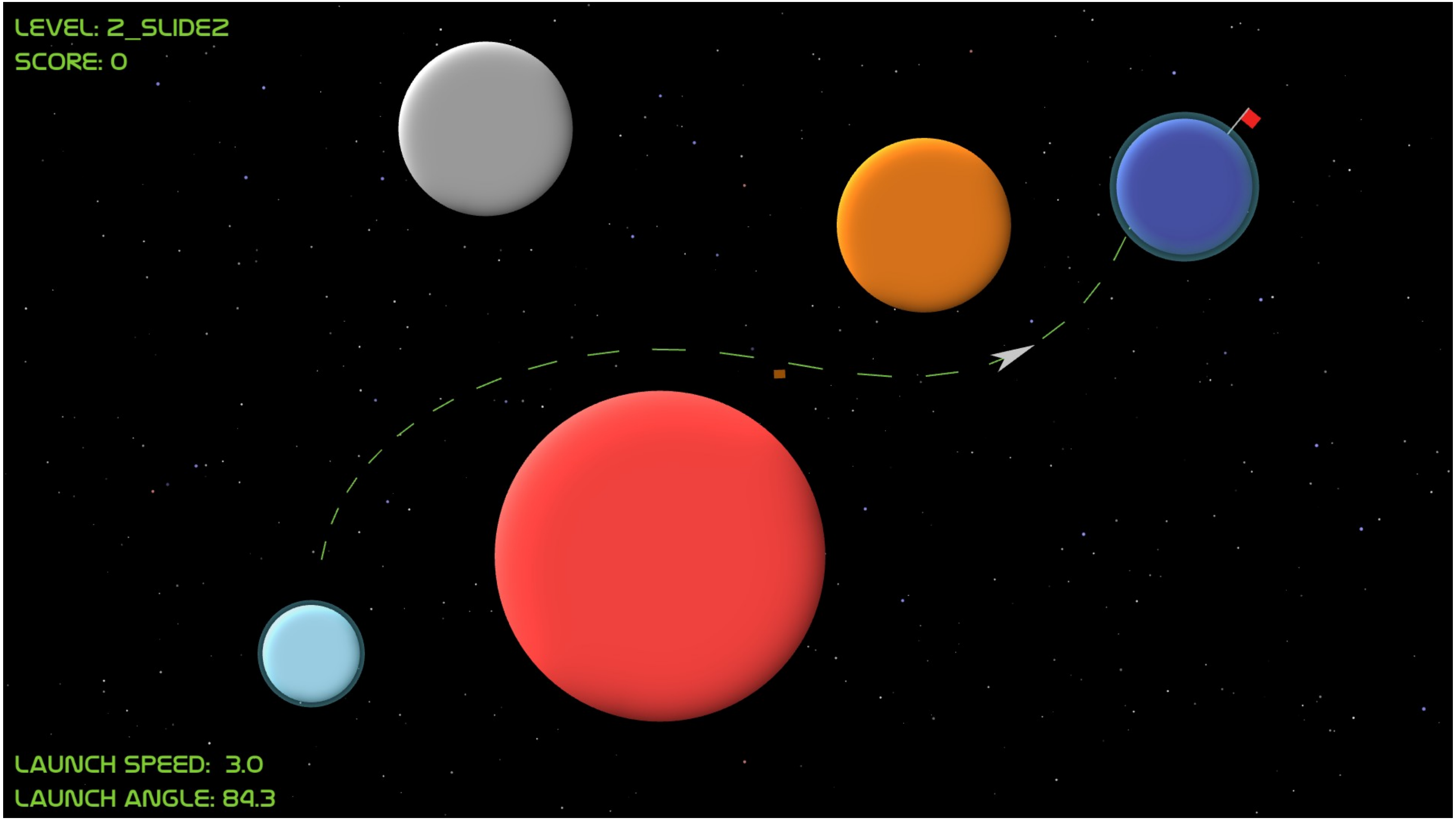


Image source:
https://www.askamathematician.com/2018/05/q-what-is-the-most-complicated-equation/


LEVEL: 2_SLIDE2
SCORE: 0
LAUNCH SPEED: 3.0
LAUNCH ANGLE: 84.3

Gravity on a bigger scale

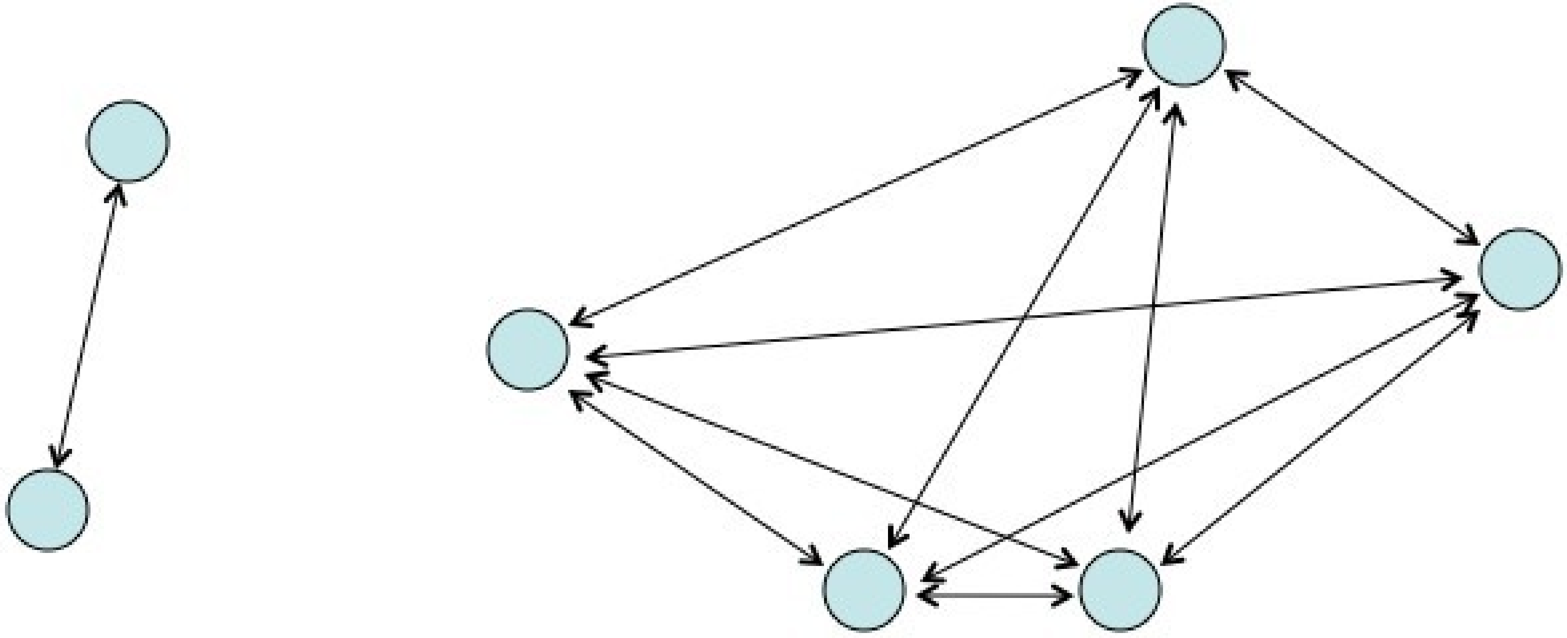
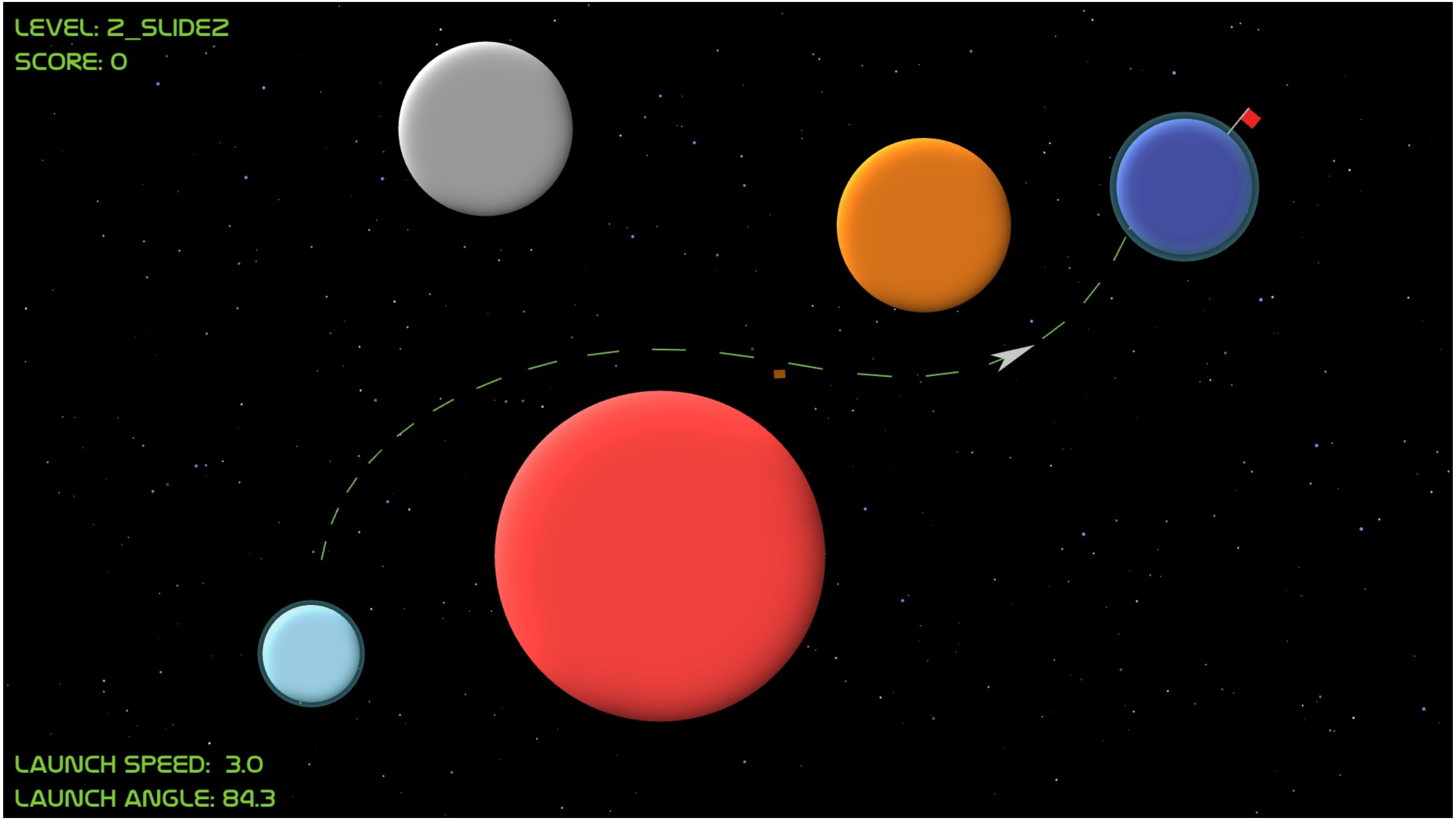
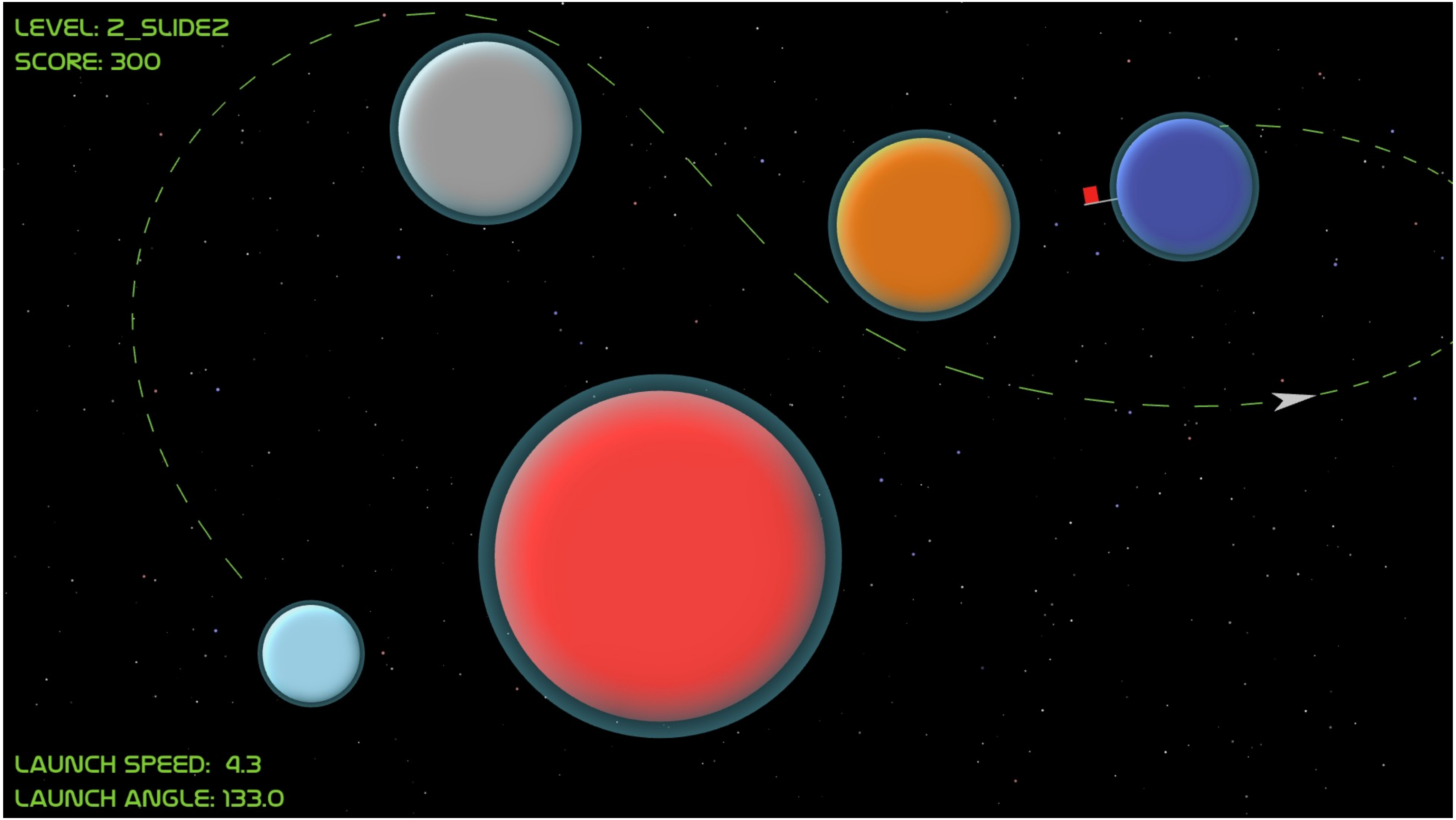
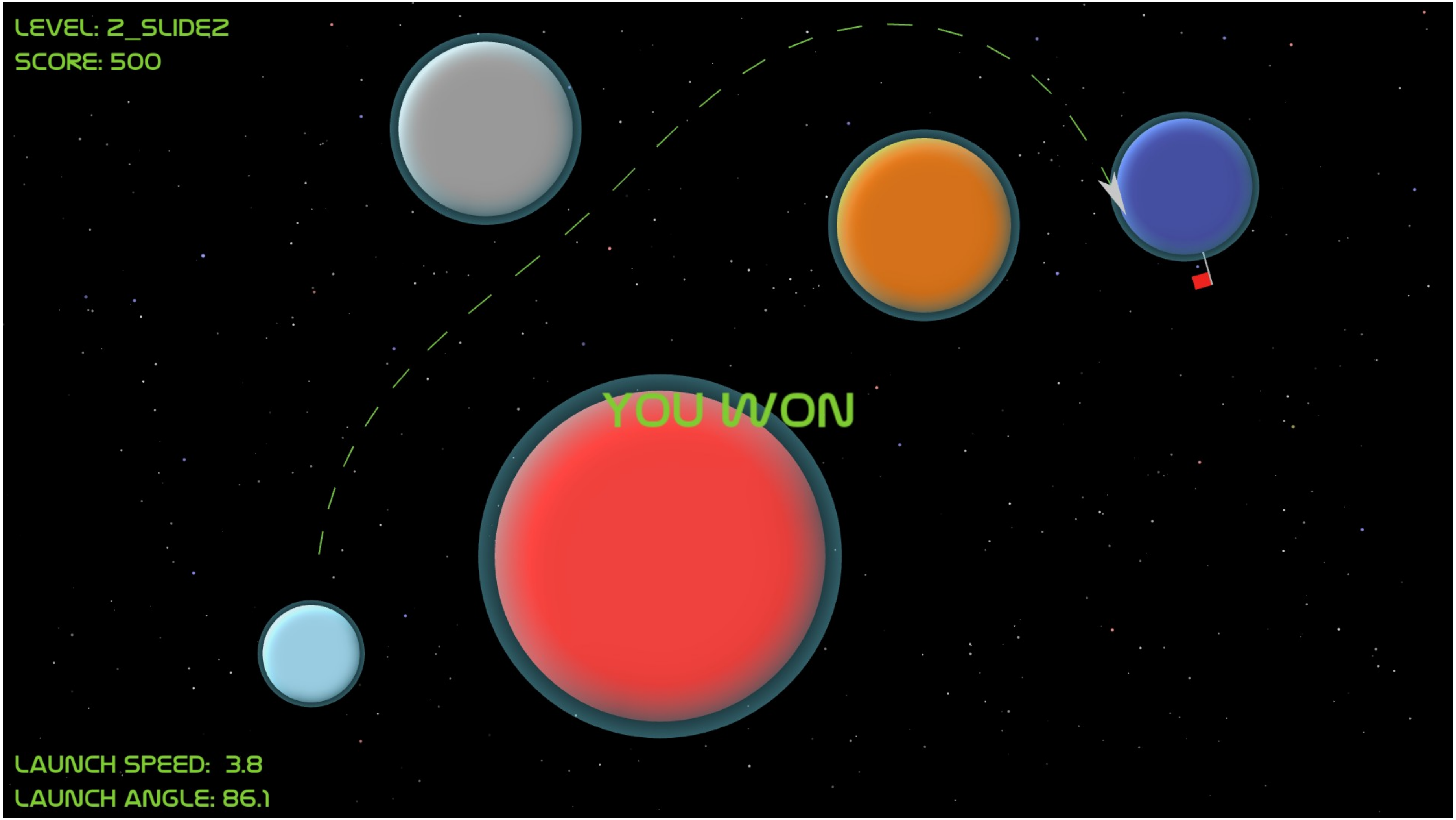
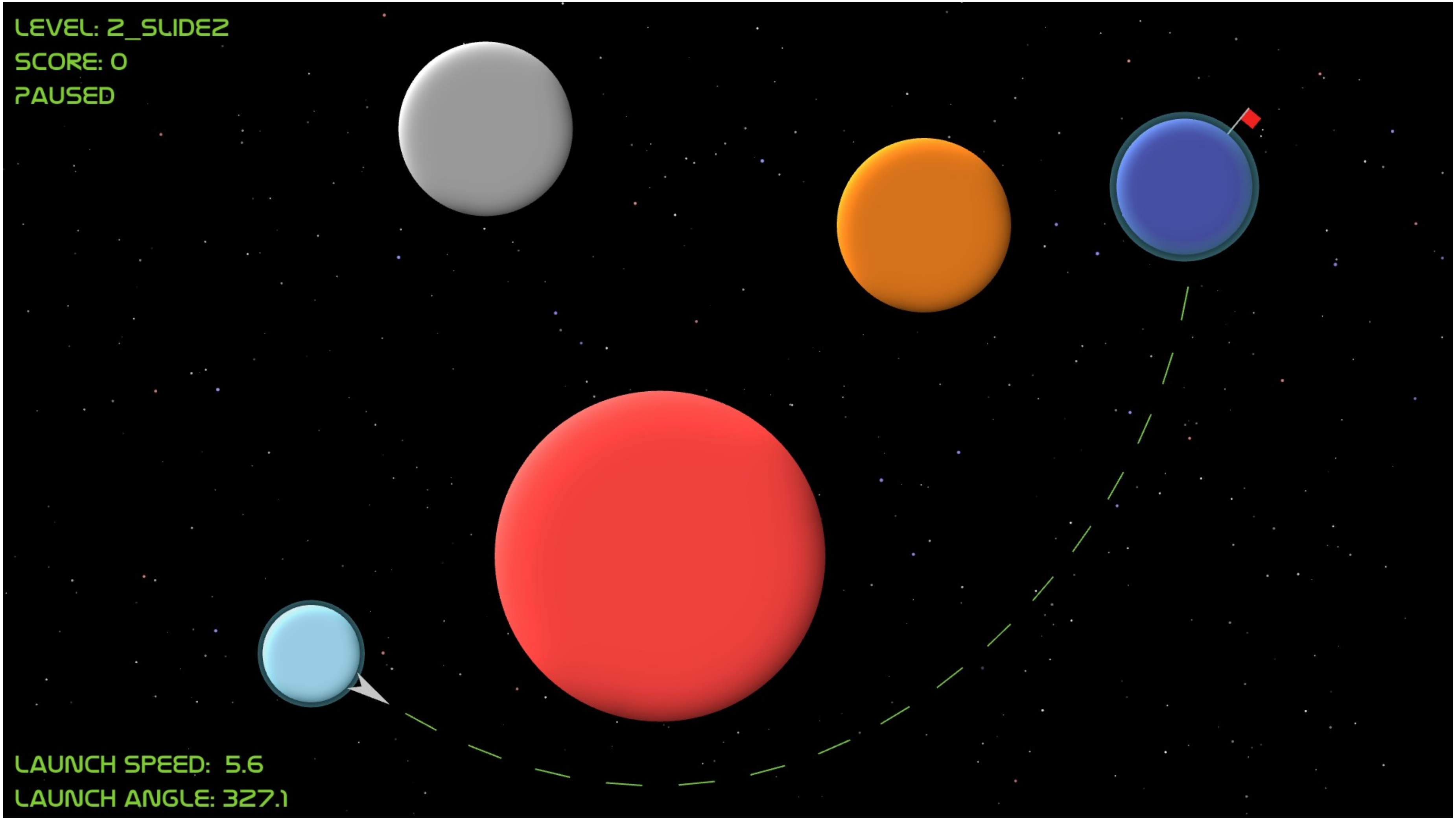


Image source:
https://www.askamathematician.com/2018/05/q-what-is-the-most-complicated-equation/


LEVEL: 2_SLIDE2
SCORE: 0
LAUNCH SPEED: 3.0
LAUNCH ANGLE: 84.3


LEVEL: 2_SLIDE2
SCORE: 300
LAUNCH SPEED: 4.3
LAUNCH ANGLE: 133.0


LEVEL: 2_SLIDE2
SCORE: 500
YOU WON
LAUNCH SPEED: 3.8
LAUNCH ANGLE: 86.1


LEVEL: 2_SLIDE2
SCORE: 0
PAUSED
LAUNCH SPEED: 5.6
LAUNCH ANGLE: 327.1

Concept of the game: Summary

- Simplified physics model
- Multiple forces acting upon the player
- Player has no control after launch

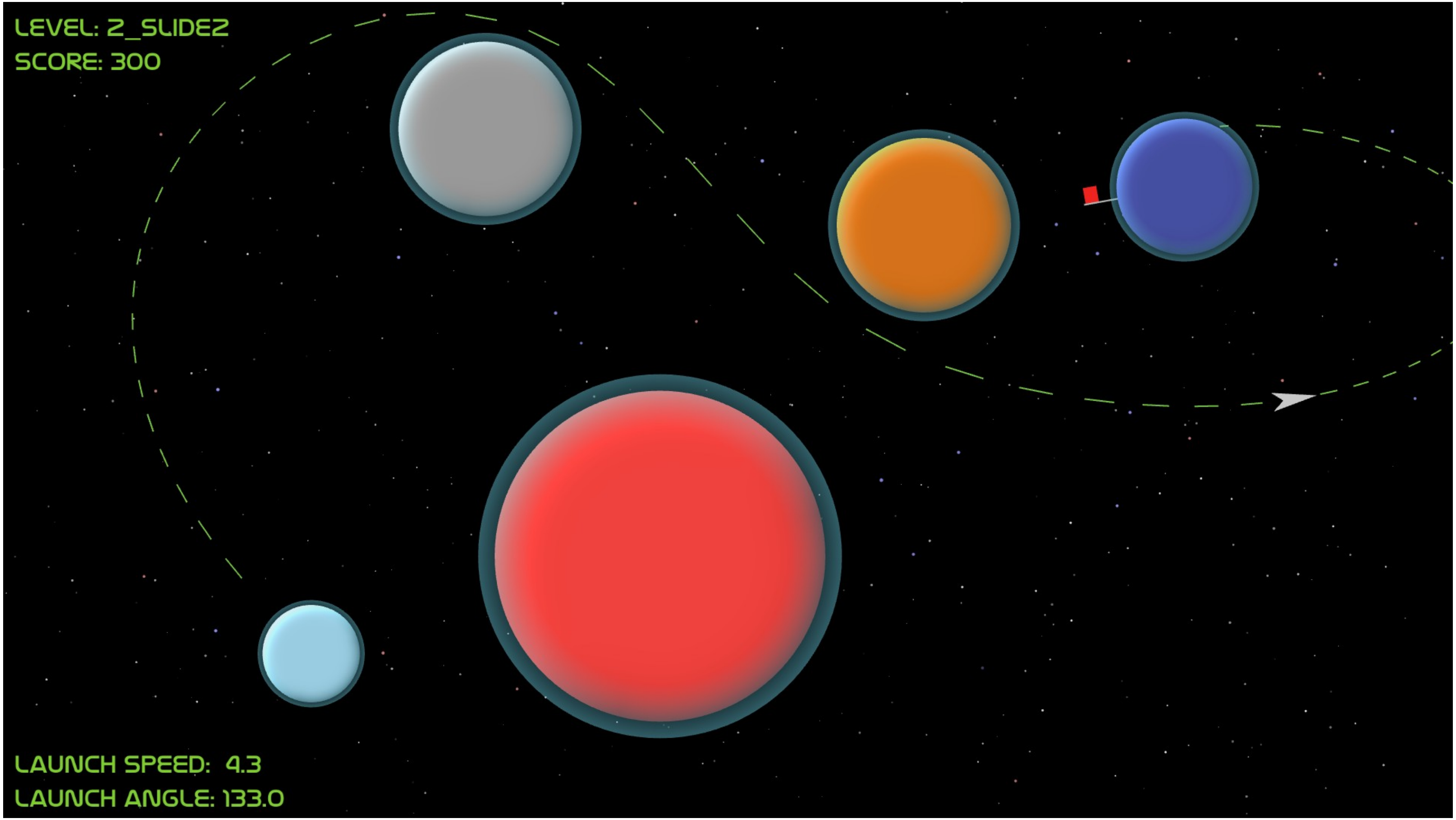
  → Forces are not his enemy, but his tool
- Terraform planets to earn bonus points

Goals

Main Goal: Reach the planet marked with a flag

Bonus goals → Exploitation of multiple possible solutions!

1. Terraform all planets
2. Collect a satellite
3. Land directly on the flag
4. Find the shortest / quickest path
5. Reach the highest score


LEVEL: 2_SLIDE2
SCORE: 300
LAUNCH SPEED: 4.3
LAUNCH ANGLE: 133.0

Movement

Equations of motion:

$$Acceleration\ \vec{a}(t)$$

$$Velocity\ \vec{v}(t) = \vec{v}_0 + \int \vec{a}(t)\,dt$$

$$Displacement\ \vec{x}(t) = \vec{x}_0 + \int \vec{v}(t)\,dt$$

$$Force\ \vec{F}(t) = m\vec{a}(t)$$

Equations in AstroFlight:

$$m_s \vec{a}(t) = \sum_i \frac{G m_s m_i}{\|\vec{r}_{si}(t)\|^3} \vec{r}_{si}(t)$$

$$\vec{v}(t) = \vec{v}(t-1) + \vec{a}(t)$$

$$\vec{x}(t) = \vec{x}(t-1) + \vec{v}(t)$$

$$G = 6.6743\left(\cdot 10^{-11} \frac{m^3}{kg\,s^2}\right)$$

Movement

1 second = 60 ticks

1 tick = 1 evaluation of all motion equations

→ constant game speed with 60 FPS

→ can be problematic in more complex games

Movement

1 second = 60 ticks

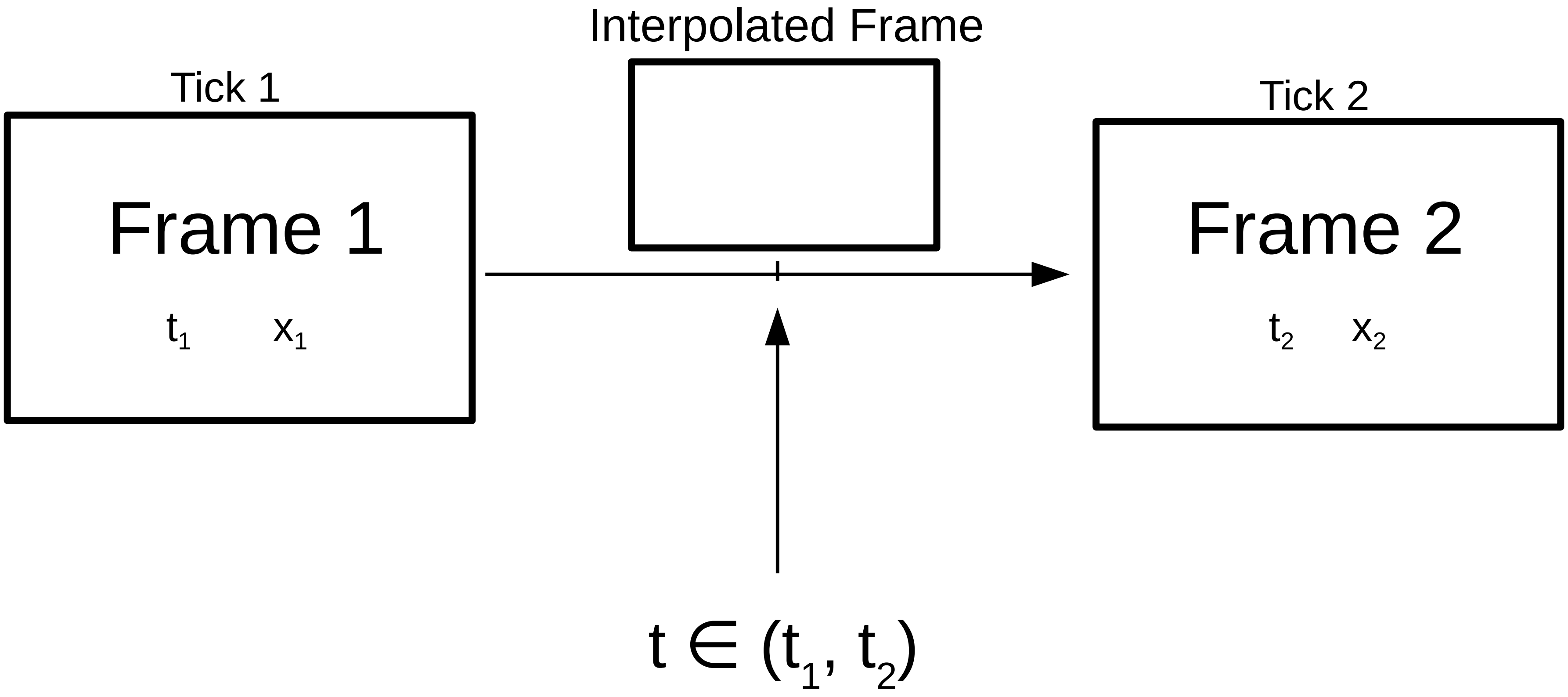
1 tick = 1 evaluation of all motion equations

→ constant game speed with 60 FPS

→ can be problematic in more complex games

→ possible solution: variable FPS $\vec{a} = \boxed{\Delta t} \sum_i \frac{G m_i}{\|\vec{r}_{si}\|^3} \vec{r}_{si}$

Interpolation



Interpolation

Naive approach: Wait 1 tick and interpolate directly

Compute x_1 at t_1

Compute x_2 at t_2

Render frame 1

While waiting for t_3:

- Interpolate x_t at $t \in (t_1, t_2)$ and render frame t

Compute x_3 at t_3

Render frame 2

Interpolation

Better approach: Prediction

Compute x_1 at t_1 and render frame 1

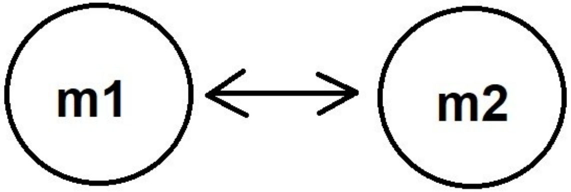
While waiting for t_2:

Predict x_t at $t \in (t_1, t_2)$ and render frame t

$$\boxed{x_t = x_1 + \Delta t\, v_1 + \frac{\Delta t^2}{2} a_1} \qquad \Delta t = t - t_1$$

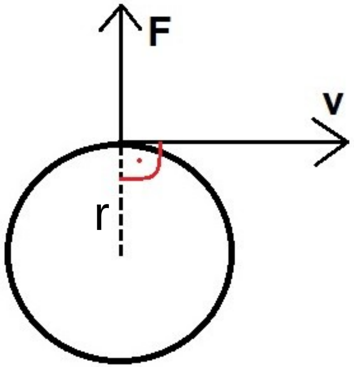
Important forces

Gravitational force $\frac{G m_1 m_2}{\|\vec{r}_{12}\|^2}\hat{r}_{12}$

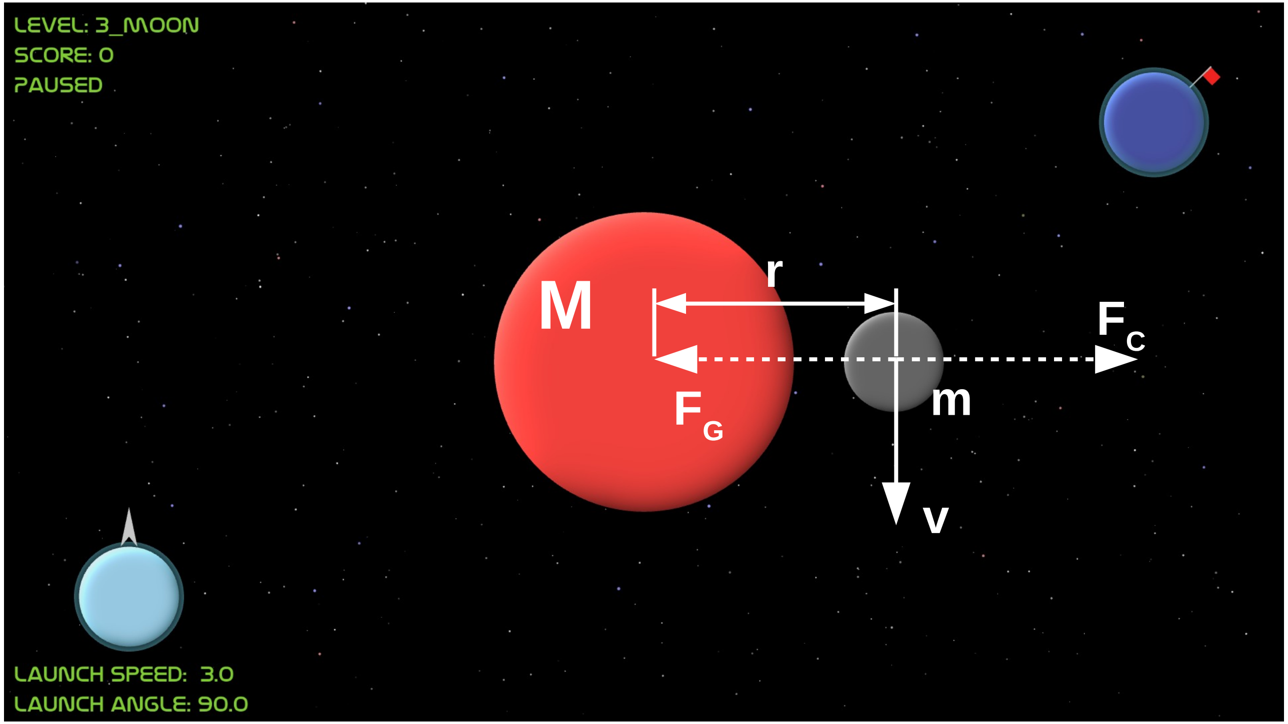


$$G = 6.6743\left(\cdot 10^{-11}\frac{m^3}{kg\,s^2}\right)$$

Centrifugal force $m\frac{v^2}{r}\hat{r}$



Celestial mechanics

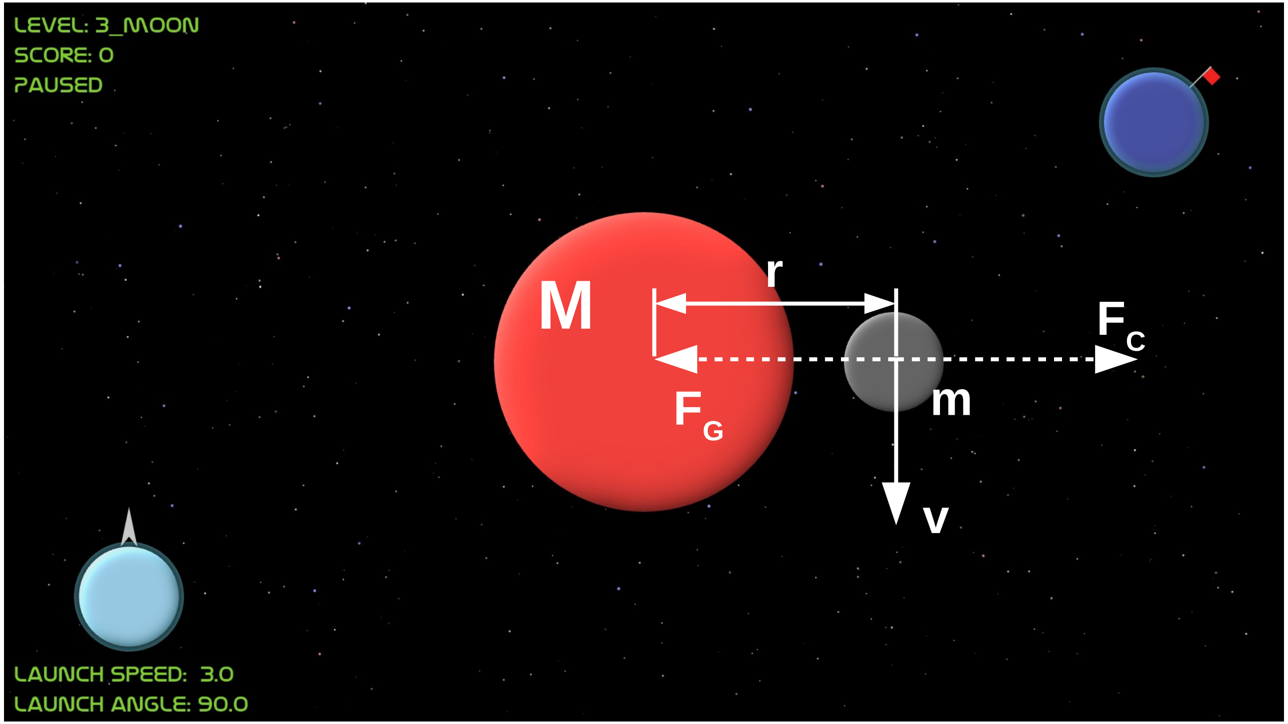


Celestial mechanics

$$|F_{Centrifugal}| = |F_{Gravity}|$$

$$m\frac{v^2}{r} = G\frac{Mm}{r^2}$$

$$v = \sqrt{\frac{GM}{r}}$$

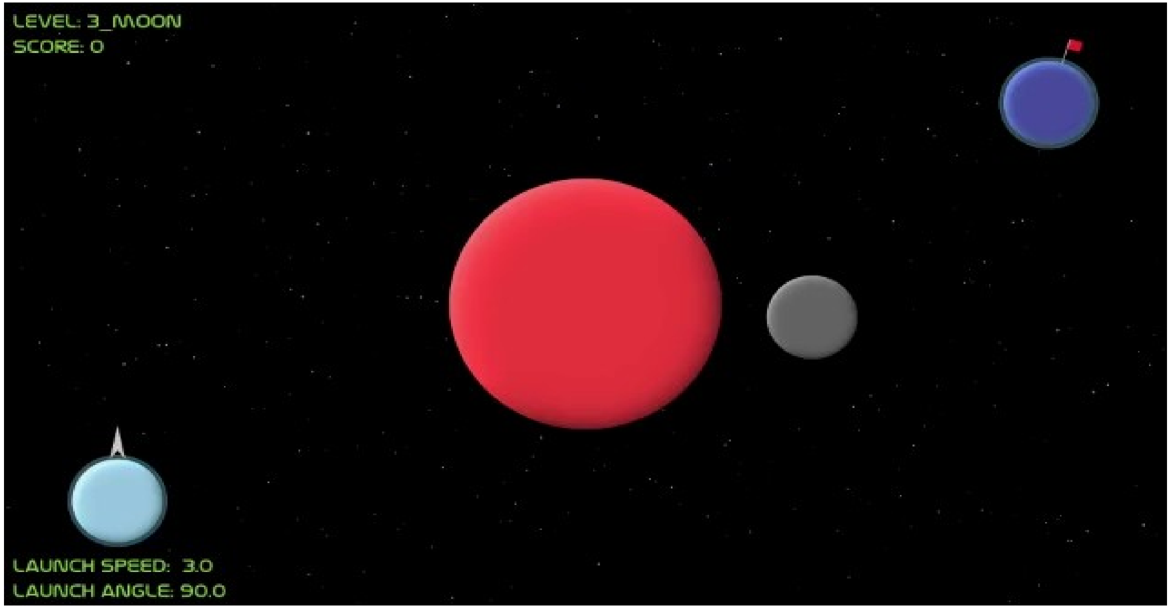


Celestial mechanics

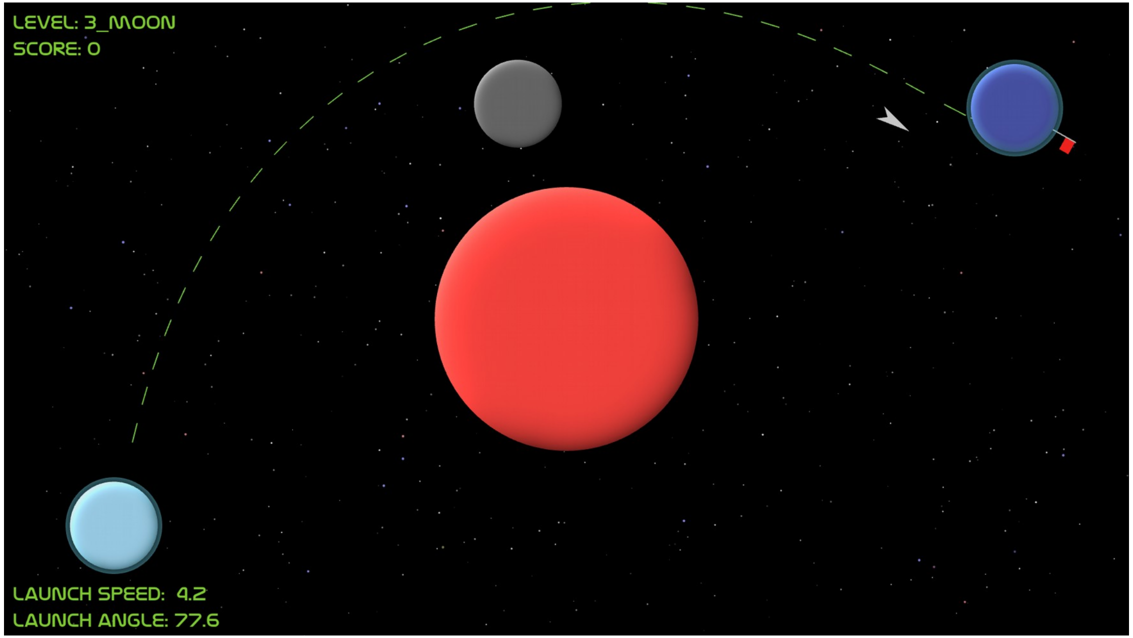
$$|F_{Centrifugal}| = |F_{Gravity}|$$

$$m\frac{v^2}{r} = G\frac{Mm}{r^2}$$

$$v = \sqrt{\frac{GM}{r}}$$

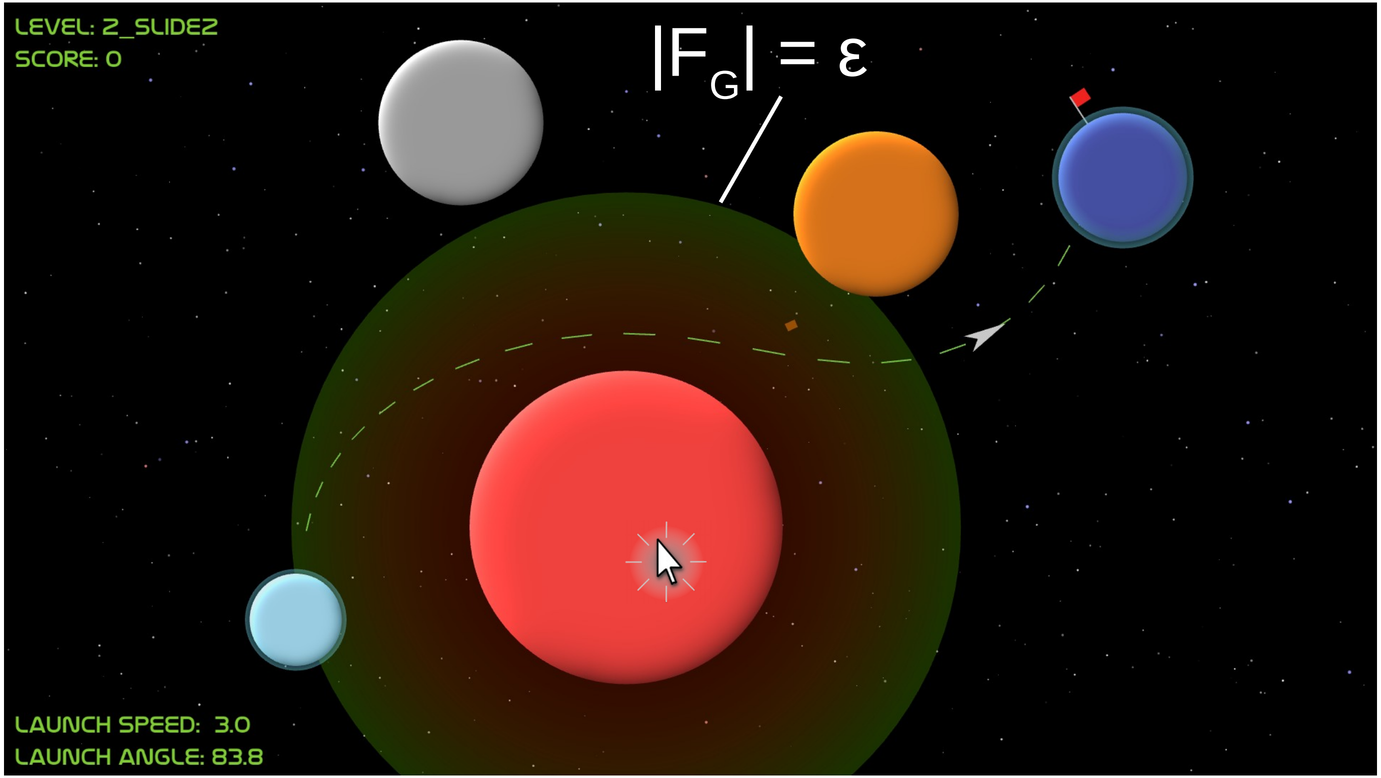


Non-autonomous systems

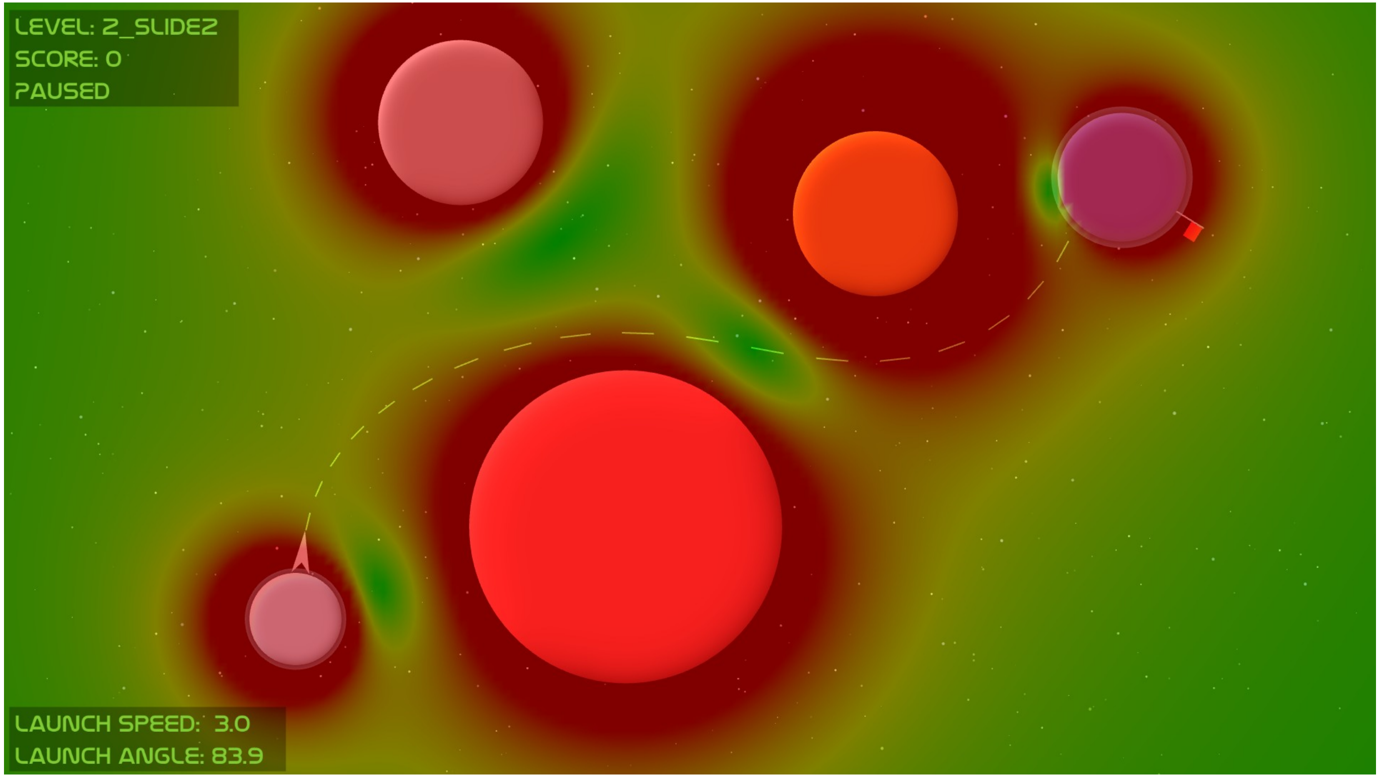


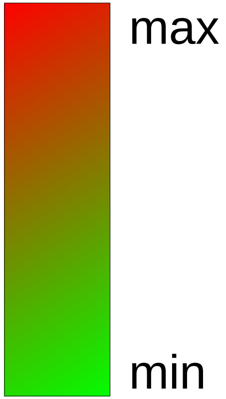
→ Less predictability!

$r_1 = r_2 \not\Rightarrow m_1 = m_2$



Better visualization of $|F_G|$





Outlook

Possible further implementations:

- Bonus goals
- Level editor
- Procedual level generation
- N-body problem
- Creating a “pseudo-universe”

blackhole.lvl

0

2

130 70 100 100 200 1000 200 0.2 0.5

100 60 200 100 0 150 300 0.01 -0.1

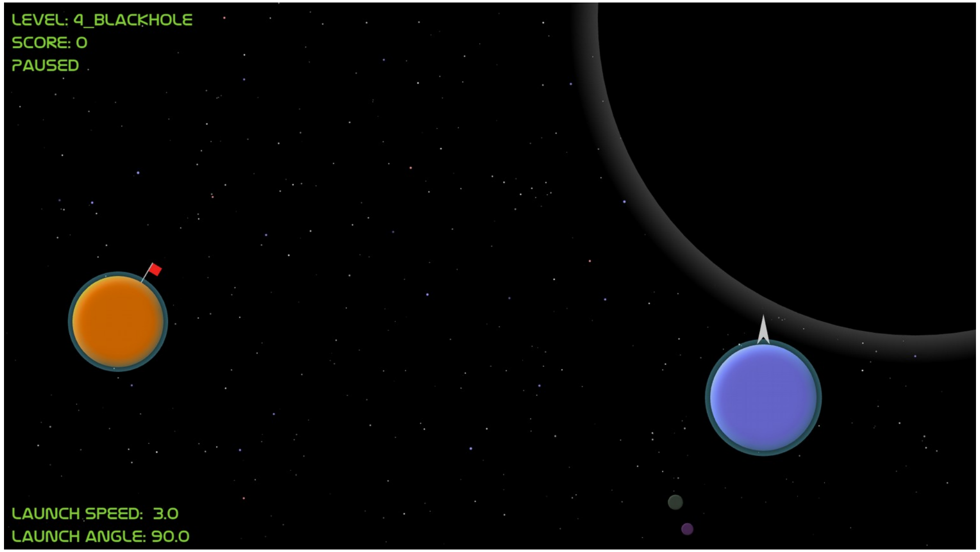
2

5 10 50 60 50 0 180 230 1

3 8 70 40 80 0 200 240 1

1

1700 1200 700 0 0



Schwarzschild radius for non-rotating black holes:

$$r_s = \frac{2GM}{c^2}$$

Outlook

Possible further implementations:

- Bonus goals
- Level editor
- Procedual level generation
- N-body problem
- Creating a “pseudo-universe”

Pseudo universe

- Three elements R, G, B with different masses
- Planets are composed of these elements
- Mass of planet = Area * density
- Is the planet terraformable?

  → Composition / Color

Any questions?