Final Exam

Serdar ARITAN

Biomechanics Research Group,
Faculty of Sports Sciences, and
Department of Computer Graphics
Hacettepe University, Ankara, Turkey

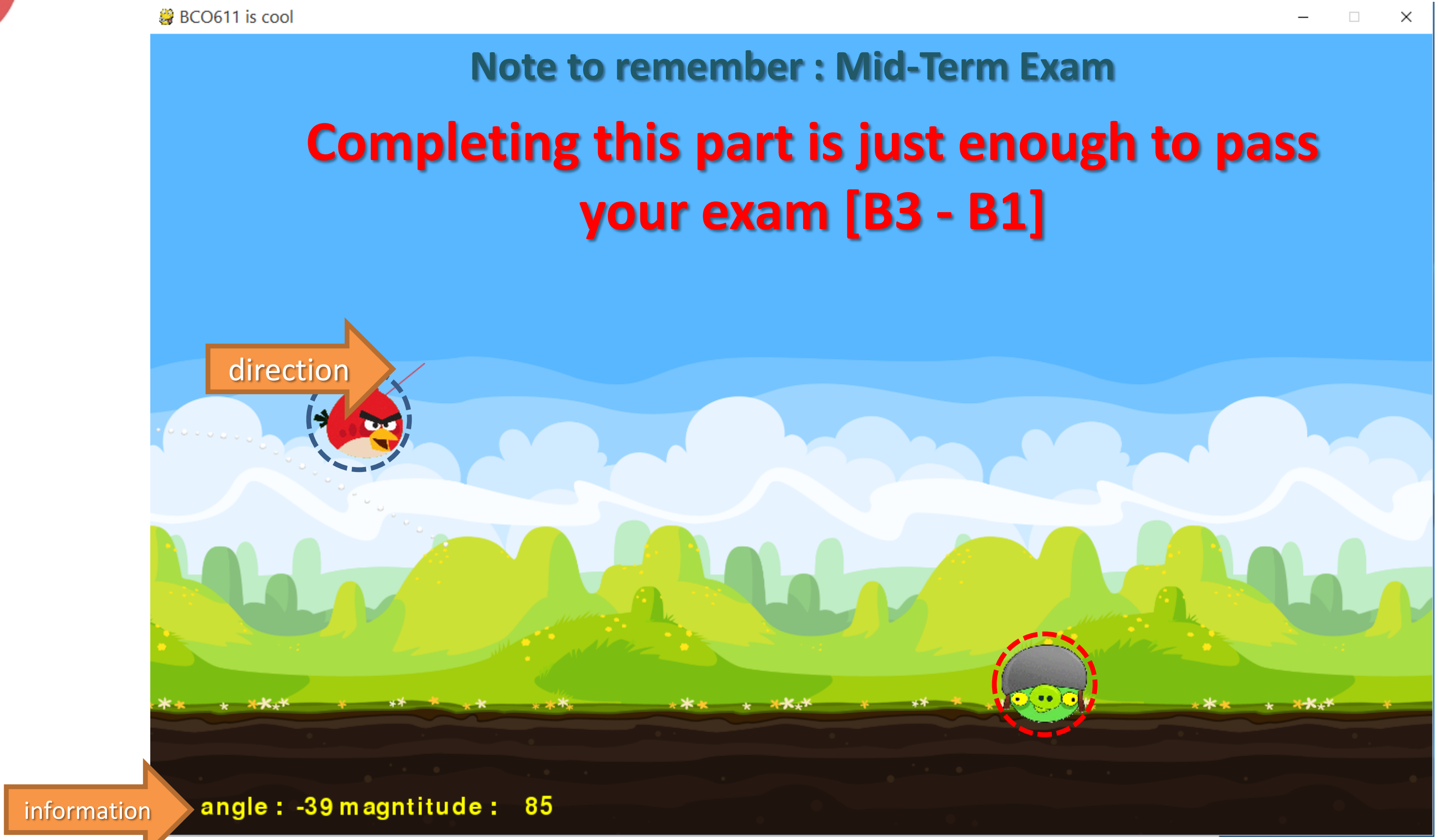
Final Exam by Using PyGame

- Apply the functions (`collision detection` and `post-collision velocity calculation`) you wrote during the mid-term exam into the `myGame.py` program.
- You can modify the `myGame.py` as much as you want.
  - See: `myGame.py`
  - There must be a collision detection between the `redStripe` and the `rock`
  - After the collision there will be a movement accordingly (`redStripe` and `rock`)
  - `redStripe`'s initial **velocity** (magnitude) and the **angle** will be **displayed**

Completing this part is enough to pass your exam [B3 – B1]

Note to remember : Mid-Term Exam

- Write a `collision detection` function in python.
  - When the collection occurs function returns **the angle and the point of collision**
- Write a `post-collision velocity calculation` function.
  - Function returns the post-collision velocities of the bodies as vector values.
  - Input arguments of the function is **the angle and the point of collision, masses of the bodies, the coefficient of restitution**

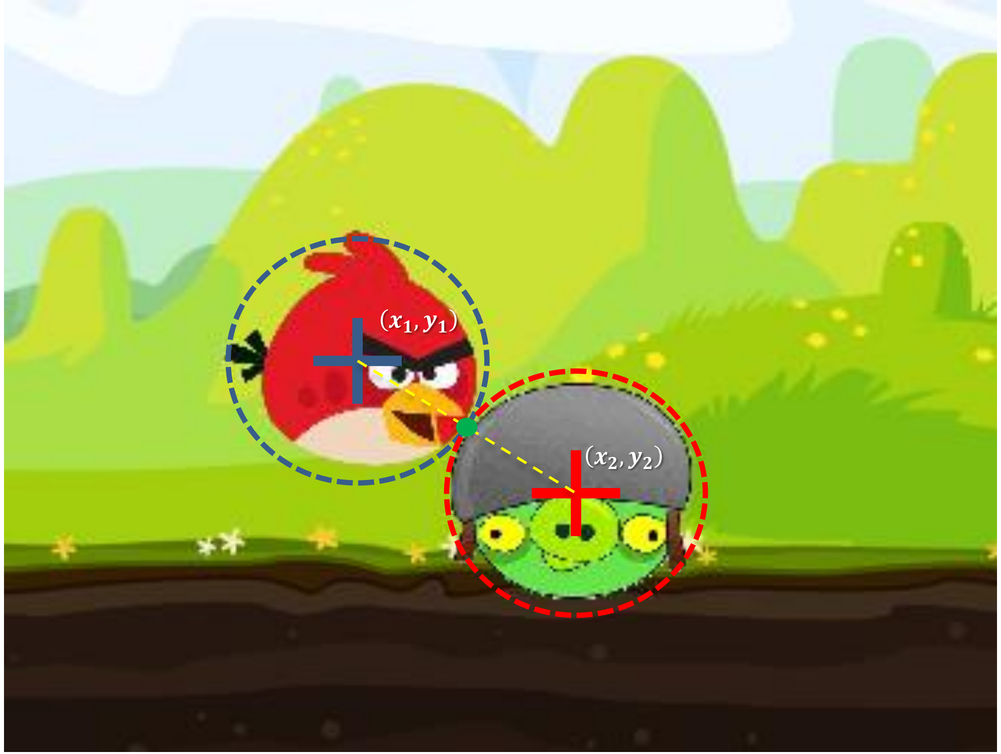


Note to remember : Mid-Term Exam

Completing this part is just enough to pass your exam [B3 - B1]

```
angle :  -39 magntitude :   85
```

Note to remember : Mid-Term Exam



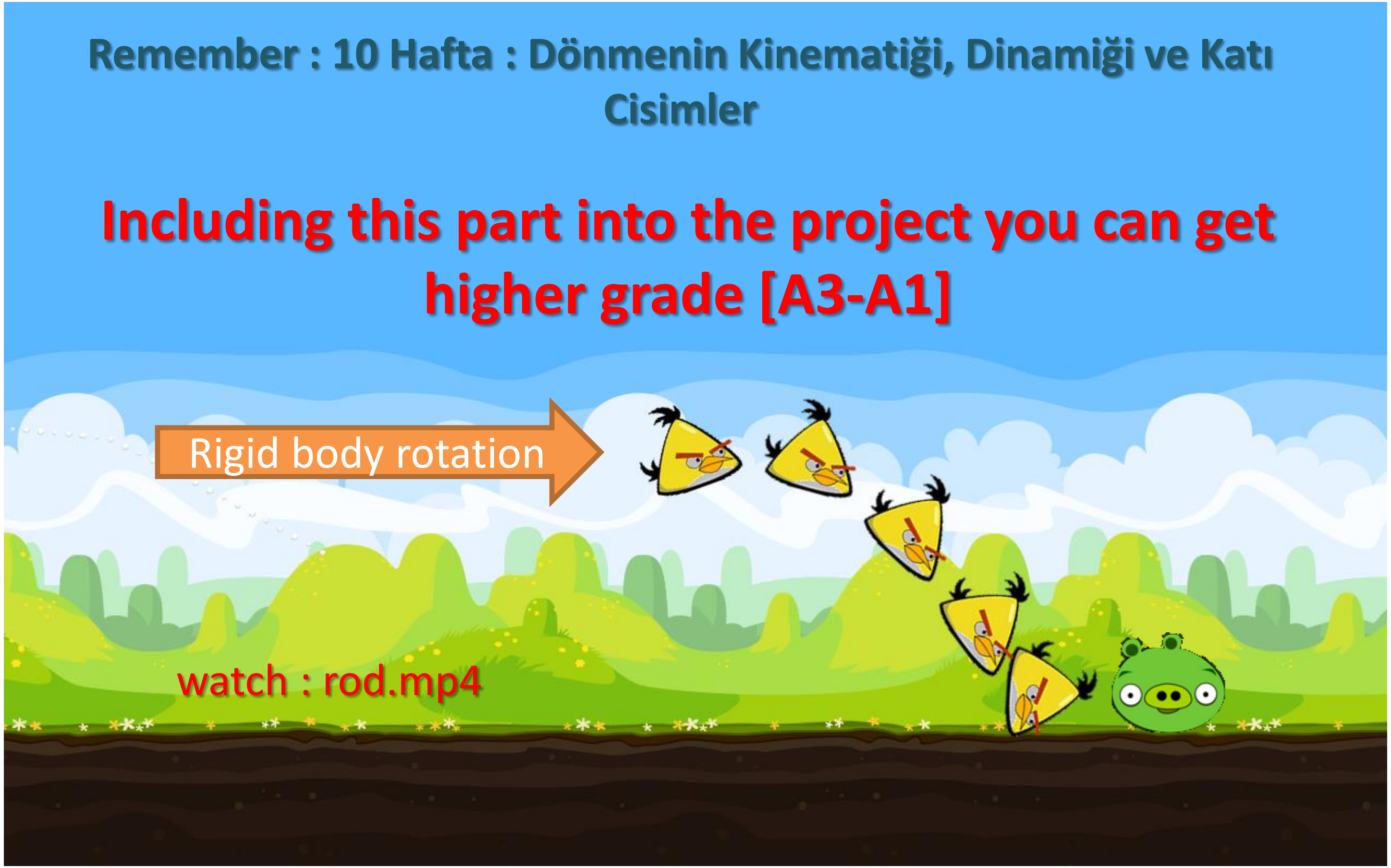
$$\theta = \operatorname{atan}\left(\frac{|y_2 - y_1|}{|x_2 - x_1|}\right)$$

$$a = r_2 \frac{|x_2 - x_1|}{r_1 + r_2} \qquad b = r_2 \frac{|y_2 - y_1|}{r_1 + r_2}$$

$$(x_c, y_c) = (x_2 + a \, , y_2 + b)$$

Final Exam by Using PyMunk

- You may also use PyMunk for the Final Exam. But your program should follow some rules.
- Considering **PyMunk** has its own integrator and collision detection.
- Your program must have multi-body objects ,which means, you must use 2 or more **bodies** that **connected** by **joints**.
- There must be a collision with another multi-body object
- Your scene must have a ground.



Remember : 10 Hafta : Dönmenin Kinematiği, Dinamiği ve Katı Cisimler

Including this part into the project you can get higher grade [A3-A1]

Rigid body rotation

watch : rod.mp4

Remember : 10 Hafta : Dönmenin Kinematiği, Dinamiği ve Katı Cisimler

Including this part into the project you can get higher grade [A3-A1]

Rigid body rotation

watch : rod.mp4

Rotational Motion Around a Moving Center of Mass

It is not simple to solve these equations analytically, but it is straight forward to implement a numerical solution. :

$$\boldsymbol{v}(t_0+\Delta t) \approx \boldsymbol{v}(t_0) + \boldsymbol{a}(t_0, \boldsymbol{r}(t_0), \boldsymbol{v}(t_0))\Delta \boldsymbol{t}$$

$$\boldsymbol{r}(t_0+\Delta t) \approx \boldsymbol{r}(t_0) + \boldsymbol{v}(t_0+\Delta t)\Delta t$$

$$\boldsymbol{\omega}(t_0+\Delta t) \approx \boldsymbol{\omega}(t_0) + \boldsymbol{\alpha}(t_0, \boldsymbol{\theta}(t_0), \boldsymbol{\omega}(t_0))\Delta \boldsymbol{t}$$

$$\boldsymbol{\theta}(t_0+\Delta t) \approx \boldsymbol{\theta}(t_0) + \boldsymbol{\omega}(t_0+\Delta t)\Delta t$$

Remember : 10 Hafta : Dönmenin Kinematiği, Dinamiği ve Katı Cisimler

```
# Calculate motion
  for i in range(n-1):
    # Find force acting on each edge
    fnet = np.array([0,0,0])
    tnet = 0.0
    u = np.array([np.cos(theta[i]), np.sin(theta[i]), 0])
    # Position of edge A
    rr = r[i] + 0.5*L*u
    # Collision with bottom wall
    dr = rr[1]
    f = -k*dr*(dr<0.0)*np.array([0,1,0])
    fnet = fnet + f
    torque = np.cross((rr-r[i]),f)
    tnet = tnet + torque
    # Position of edge B
    rr = r[i] - 0.5*L*u
    # Collision with bottom wall
    dr = rr[1]
    f = -k*dr*(dr<0.0)*np.array([0,1,0])
    fnet = fnet + f
    torque = np.cross((rr-r[i]), f)
    tnet = tnet + torque
    # Add gravity
    fnet = fnet - m*g*array([0,1,0])
    # Integration step - Newton - Euler
    a = fnet/m
    v[i+1] = v[i] + a*dt
    r[i+1] = r[i] + v[i+1]*dt
    alphaz = tnet[2]/I
    omega[i+1] = omega[i] + alphaz*dt
    theta[i+1] = theta[i] + omega[i+1]*dt
    t[i+1] = t[i] + dt
```

Remember : 10 Hafta : Dönmenin Kinematiği, Dinamiği ve Katı Cisimler

Final Exam by Using PyGame

Final Sınavı Sunumu için 7 Haziran Cuma günü saat 18:30 de Animasyon Lab da buluşaçağız.

Sınav teslim tarihi : 7 Haziran2024 Cuma saat 18:30

Teslim adresi : Animasyon Lab