Második feladat: Hullámvasút

Szirmay-Kalos László

BME IIT

2025

Revíziók

Dátum	Változások	Verzió
2024. szeptember	Első változat	1.0
2025. február	Magyarázatok, tehetetlenségi nyomaték kiegészítés	1.1

Tartalom

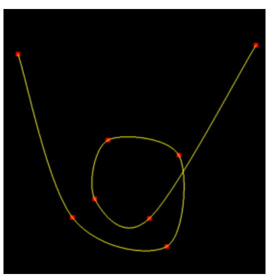
1. Feladatkiírás

Ebben a feladatban egy 2D hullámvasút szimulációt kell megvalósítani. A hullámvasút pályája Catmull-Rom spline, amelyet a kontrollpontjai definiálják. **A spline uniform paraméterezésű, azaz a csomóértékek különbsége minden két egymás utáni kontrollpontra ugyanaz. A legelső és legutolsó kontrollpontban az Hermite interpolációhoz használt sebességvektor (azaz a spline paraméter szerinti deriváltja) zérus.**

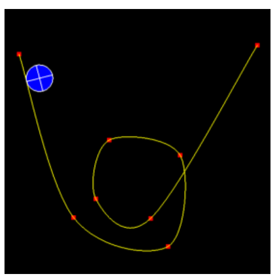
A virtuális kamera egy 20 m x 20 m-es tartományt mutat be. A világban a nehézségi gyorsulás $g = 40$ m/s^2, iránya pedig a $-y$ tengely.

A pálya kontrollpontjai az egér balgomb lenyomással hozhatók létre a kurzor helyén. A kontrollpontokat a program piros (RGB=1,0,0) négyzetként (GL_POINT primitívvel) jeleníti meg. A pontméret 10. Ha legalább két kontrollpont létezik, a kontrollpontokra spline-t illeszt, és azt is megjeleníti, mégpedig sárgával (RGB=1,1,0). A vonalvastagság 3 pixel. A piros kontrollpontok takarják a sárga spline görbét.

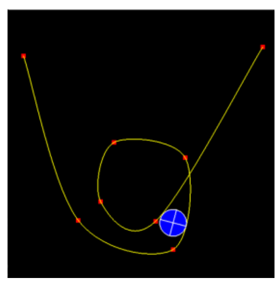
A SPACE hatására a görbe 0.01 paraméterű pontjából, álló helyzetből indítunk egy kereket. A kerék kitöltött kék (RGB=0,0,1) színű, **1 m sugarú** körlap fehér (RGB=1,1,1) körvonallal, és legalább két fehér küllővel. A kerék tömege a körvonalon egyenletesen oszlik szét, a tengely és a küllők súlytalanok. A kerék gördül a pályán. **A mozgás sebességvektorát (a görbe érintőjét) analitikusan kell számítani, a közelítő differenciálás nem elegendő pontosságú.** A sín csak nyomni tudja a vasutat, húzni nem, ha ilyen fordulna elő, akkor az azt jelenti, hogy a kocsi lerepült a sínről, és emiatt tetszés szerinti módon el kell tüntetni. Ha a kocsi elindulna visszafelé, akkor a kezdőpontba (a spline legelső kontrollpontja) kell teleportálni, és a szimulációt álló helyzetből indulva újrakezdeni.



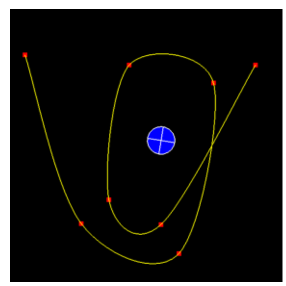
Megrajzolt pálya



Indulás után



Kicsit később



Leestem ☹

2. Dinamikai szimuláció

A feladatot valósidejű dinamikai szimulációval kell megoldani. Ez azt jelenti, hogy az onTimeElapsed(tstart, tend) eseménykezelőben az idő múlásával a virtuális világ állapotát fel kell zárkóztatni, azaz a tstart időpontban érvényes állapotból kiindulva, a fizikai törvényeket alkalmazva, a tend időpillanatban érvényes állapotot kell meghatározni. A hullámvasút feladatban a kerék mozog, tehát annak állapotát kell követni, de ez nem független a pályagörbétől.

Ha egy test merev, azaz a mozgás során nem deformálódik, akkor a mozgása leírható, mint a súlypontjának haladó mozgása, és a súlypont körüli forgás. A súlypont úgy mozog, mintha a teljes tömeg egyetlen pontban volna összesűrítve.

A $\boldsymbol{p}(t)$ súlypont pályájának idő szerinti deriváltja a $\boldsymbol{v}(t)$ sebesség, a sebesség idő szerinti deriváltja pedig az $\boldsymbol{a}(t)$ gyorsulás. A súlypont haladó mozgását a dinamika alapegyenlete, azaz Newton második axiómája határozza meg. Eszerint az $\boldsymbol{I} = m\boldsymbol{v}$ impulzus, más néven lendület, idő szerinti deriváltja a testre ható $\boldsymbol{F}$ erővel egyezik meg:

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{v}, \qquad \frac{\mathrm{d}\boldsymbol{I}}{\mathrm{d}t} = m\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = m\boldsymbol{a} = \boldsymbol{F}.$$

A forgó mozgás hasonló jellemzőkkel írható le. Az $\boldsymbol{L}$ impulzusnyomaték, más néven impulzusmomentum vagy perdület, idő szerinti deriváltja az $\boldsymbol{M}$ forgatónyomaték. A párhuzam ellenére háromdimenzióban a forgó mozgás leírása sokkal bonyolultabb, mint a haladó mozgásé. Ennek az az oka, hogy amíg az impulzust a haladómozgás sebességével összekapcsoló tömeg skalármennyiség, addig a perdület a szögsebesség vektor és a tehetetlenségi nyomaték mátrix szorzata, és a pozíció párjaként fellépő orientáció sem adható meg egy vektorként (idáig a 3D elfordulást is egy mátrixszal jellemeztük). Szerencsére a feladatunk univerzuma kétdimenziós, ahol az orientációt leíró α elfordulási szög, az ω szögsebesség, a β szöggyorsulás, a Θ tehetetlenségi nyomaték, az L impulzusnyomaték és az M forgatónyomaték egyaránt skalár, a dinamikai egyenletek pedig:

$$\frac{\mathrm{d}\alpha}{\mathrm{dt}} = \omega, \qquad \frac{\mathrm{d}L}{\mathrm{dt}} = \Theta\frac{\mathrm{d}\omega}{\mathrm{dt}} = \Theta\beta = M.$$

A dinamikai szimulációnál a rendszer t-időpontbeli állapotából a mozgás egyenletek felhasználásával kell a későbbi időpontbeli állapotra következtetni. Az állapot a $\boldsymbol{p}$ pozíciót, $\boldsymbol{v}$ sebességet, α elfordulási szöget és ω szögsebességet foglalja magában, a számításhoz pedig felhasználjuk a pillanatnyi $\boldsymbol{F}$ erőt és M forgatónyomatékot. Tegyük fel, hogy a $t + \Delta t$-időpontbeli állapotra vagyunk kíváncsiak, ahol Δt olyan kicsi, hogy a differenciálhányadosok differenciahányadosokkal közelíthetők:

$$\boldsymbol{a} = \frac{\boldsymbol{F}}{m}, \qquad \boldsymbol{v}(t+\Delta t) \approx \boldsymbol{v}(t) + \boldsymbol{a}\Delta t, \qquad \boldsymbol{p}(t+\Delta t) \approx \boldsymbol{p}(t) + \boldsymbol{v}(t)\Delta t + \frac{1}{2}\boldsymbol{a}(\Delta t)^2,$$

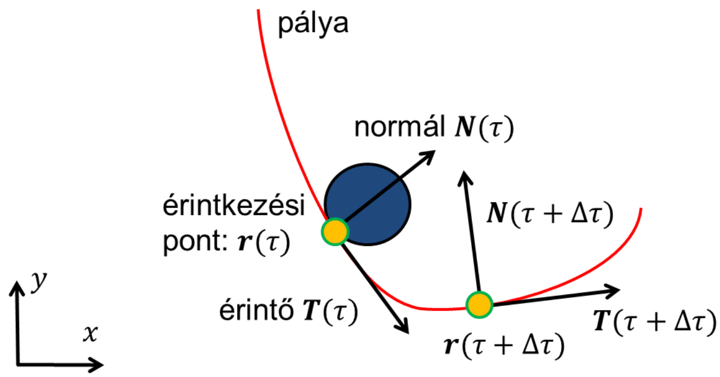
$$\beta = \frac{M}{\Theta}, \qquad \omega(t+\Delta t) \approx \omega(t) + \beta\Delta t, \qquad \alpha(t+\Delta t) \approx \alpha(t) + \omega(t)\Delta t + \frac{1}{2}\beta(\Delta t)^2.$$

Ha az állapotváltozók bármely kombinációja érvényes lehet, akkor szabad mozgásról beszélünk. A hullámvasút példában a kerék mozgása akkor szabad, ha a test már lerepült a pályáról és a nehézségi erőtérben kényszerek nélkül zuhan. A súlypont, mint követett pont választásának előnye, hogy az $\boldsymbol{F} = m\boldsymbol{g}$ nehézségi erő nem generál forgatónyomatékot a súlypontra, így a szabad mozgásra vonatkozó egyenletek:

$$\boldsymbol{a} = \boldsymbol{g}, \qquad \boldsymbol{v}(t+\Delta t) \approx \boldsymbol{v}(t) + \boldsymbol{g}\Delta t, \qquad \boldsymbol{p}(t+\Delta t) \approx \boldsymbol{p}(t) + \boldsymbol{v}(t)\Delta t + \frac{1}{2}\boldsymbol{g}(\Delta t)^2,$$

$$M = \beta = 0, \qquad \omega(t+\Delta t) \approx \omega(t), \qquad \alpha(t+\Delta t) \approx \alpha(t) + \omega(t)\Delta t.$$

A pályán gördülés viszont nem szabad mozgás, hiszen a kényszernek, hogy a kerék érintkezik a pályával és ebben a pontban nem csúszik meg, a lerepülésig érvényesülnie kell. Kényszeres mozgásoknál kényszererők lépnek fel, amelyek éppen akkorák, hogy a mozgás a kényszerfeltételeket kielégítse. A pályán mozgásnál ilyen kényszererő a pálya nyomóereje, amely mindig merőleges a pályára. A lerepülés kulcsa ugyancsak a kényszererő lesz, mert a pálya csak nyomni tudja a kereket, húzni nem. Tehát, ha a mozgásegyenletek megoldásánál a kényszererő húzásnak adódna, akkor ettől a pillanattól kezdve át kell váltani a szabadesés szimulációjára.



A pályán gördülés kényszerei.

A kényszer formalizálásához vegyünk fel egy derékszögű bázist a kerék és pálya érintkezési pontjában. A bázis egyik egységvektora $\boldsymbol{T}$ a pálya érintője, ami a pillanatnyi haladási irányba mutat. Az érintőre merőleges egységvektor pedig az $\boldsymbol{N}$ normálvektor, amely a centripetális gyorsulás irányát jelöli ki.

A pályán maradáshoz szükséges centripetális gyorsulás $v^2\kappa$ ahol v a test sebessége, κ pedig a pálya görbülete, amely a simulókör sugarának reciproka. A testre az $m\boldsymbol{g}$ nehézségi erő és az K nagyságú $\boldsymbol{N}$ normálvektor irányú nyomóerő hat. Az K kényszererő nagysága abból a kényszerből származtatható, hogy a test a pályán marad, tehát a pályára merőleges eredő erőnek éppen a szükséges centripetális erővel kell megegyeznie:

$$K - m\boldsymbol{g}\cdot\boldsymbol{N} = mv^2\kappa \quad \rightarrow \quad K = m(\boldsymbol{g}\cdot\boldsymbol{N} + v^2\kappa).$$

A $K\boldsymbol{N}$ nyomóerő kiszámításához ismernünk kell a v sebességet, a pálya $\boldsymbol{N}$ normálvektorát és κ görbületét. Legyen a pálya paraméteres egyenlete $\boldsymbol{r}(\tau)$, amely a τ függvényében meghatározza, hogy hol érintkezik a kerék a pályával. Az érintkezési pontból a kerék súlypontja, azaz a kör középpontja a normálvektor irányában R távolságra van, ahol R a kör sugara:

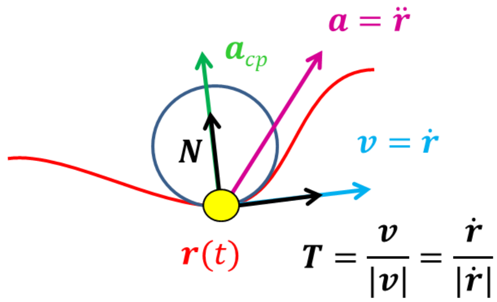
$$\boldsymbol{p} = \boldsymbol{r}(\tau) + \boldsymbol{N}R.$$

A paramétert τ-val jelöltük, hogy megkülönböztessük a szimulációs időtől, aminek a jele a t. A pálya paraméter szerinti deriváltja, $\dot{\boldsymbol{r}}(\tau)$, mindig érinti a pályát, így az egység hosszú $\boldsymbol{T}$ érintővektort a derivált normalizálásával kapjuk meg:

$$\boldsymbol{T} = \frac{\dot{\boldsymbol{r}}(\tau)}{|\dot{\boldsymbol{r}}(\tau)|}.$$

Az $\boldsymbol{N}$ normálvektor az érintővektor 90 fokos elforgatottja:

$$\boldsymbol{N} = \left(-T_y, T_x\right).$$



A pálya görbülete.

A pályát követő mozgáshoz szükséges centripetális gyorsulás $v^2\kappa$. Eszerint, ha tetszőleges választott dinamika feltételezésével meghatározzuk a centripetális gyorsulást, a görbület ebből a sebesség négyzetével való osztással megkapható. Tekintsük azt a mozgást, amikor a t idő megegyezik a τ paraméterrel. Ekkor a τ időpontban fellépő sebesség $\boldsymbol{v}(\tau) = \dot{\boldsymbol{r}}(\tau)$, a gyorsulás pedig $\boldsymbol{a}(\tau) = \ddot{\boldsymbol{r}}(\tau)$. A gyorsulás pályára merőleges, azaz normálirányú komponense a centripetális gyorsulás, így annak nagysága $\boldsymbol{a} \cdot \boldsymbol{N}$. Ezeket összerakva, a görbület kifejezhető a pálya parametrikus egyenletéből:

$$\ddot{\boldsymbol{r}}(\tau) \cdot \boldsymbol{N} = |\dot{\boldsymbol{r}}(\tau)|^2\kappa \quad \rightarrow \quad \kappa = \frac{\ddot{\boldsymbol{r}}(\tau) \cdot \boldsymbol{N}}{|\dot{\boldsymbol{r}}(\tau)|^2}.$$

A görbület ismeretében a $K = m(\boldsymbol{g} \cdot \boldsymbol{N} + v^2\kappa)$ kényszererő kiszámítható, amelynek az előjele alapján határozhatunk pályán maradásról, illetve a leesésről.

Amennyiben a kerék követi a pályát, nehézségi erőhöz hozzáadva a kiszámolt kényszererőt elvileg használhatnánk a szabad mozgás egyenleteit:

$$\boldsymbol{a} = \boldsymbol{g} + \frac{K\boldsymbol{N}}{m}, \qquad \boldsymbol{v}(t + \Delta t) \approx \boldsymbol{v}(t) + \boldsymbol{a}\Delta t, \qquad \boldsymbol{p}(t + \Delta t) \approx \boldsymbol{p}(t) + \boldsymbol{v}(t)\Delta t + \frac{1}{2}\boldsymbol{a}(\Delta t)^2.$$

Sajnos ezek a formulák közelítőek, így a test a numerikus hibák miatt minden lépésben távolodna a pályától, azaz a kényszert egyre kevésbé tartanánk tiszteletben. Vegyük észre, hogy ha az érintkezési pontban a pálya τ paraméterét ismerjük, abból az érintkezési pont helye, az érintő, azaz a sebességvektor iránya, a normálvektor, és a görbület már egyértelműen meghatározható. Tekintsük tehát a leesésig az elsődleges állapotjellemzőnek a pálya τ paraméterét, a τ paramétert léptessük az időben, és abból határozzuk meg az érintkezési pontot, sebesség irányt és a görbületet.

Hiányzik még a sebesség nagysága, amelyre két opciónk is van. Tekinthetjük a sebesség nagyságát állapotváltozónak, amit a mozgásegyenletek alapján frissítünk, tudomásul véve, hogy az egyes lépések numerikus hibái összeadódhatnak. A másik lehetőség az erők helyett az energia követése. Mivel a kényszererő merőleges a sebesség irányára, azaz a pillanatnyi elmozdulásra, az nem végez munkát a testen. Ha a csúszási súrlódástól eltekintünk, a mozgási energia és a helyzeti energia összege állandó. Vagy más szavakkal, a mozgási energia növekménye megegyezik a nehézségi erő munkájával. A mozgási energia a haladó mozgás energiájából és a forgó mozgás energiájából tevődik össze. A helyzeti energia az

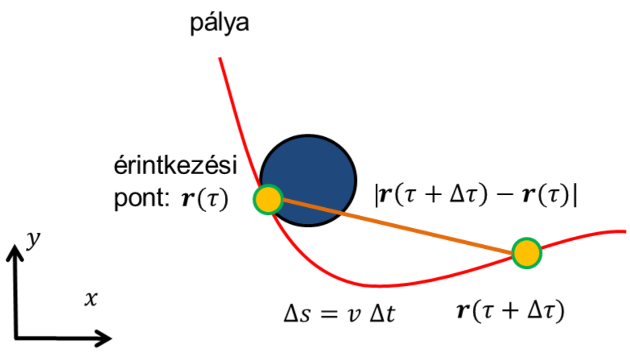
$y(\tau)$ aktuális magasság és a nehézségi gyorsulás szorzata. Ha a mozgás a $\tau = 0$ paraméterű pontból állásból indult, akkor a teljes mozgási energiának a helyzeti energia csökkenésével kell megegyeznie:

$$\frac{1}{2}mv^2 + \frac{1}{2}\Theta\omega^2 = mg(y(0) - y(\tau)).$$

A Θ tehetetlenségi nyomaték attól függ, hogy a tömeg hogyan oszlik szét a testben a forgási tengelytől mért távolság függvényében. Egy ρ távolságra lévő m tömegpont tehetetlenségi nyomatéka $m\rho^2$, ezt kell integrálással a tömegeloszlásra alkalmazni. Ha a kerék tömege a kör kerületén oszlik szét, akkor $\Theta = mR^2$. Ha viszont a tömeg a kerék tengelyében koncentrálódik, akkor $\Theta = 0$. A két szélsőérték közötti eseteket egy λ alaktényezővel vehetjük figyelembe és azt mondjuk, hogy a tehetetlenségi nyomaték általános esetben $\Theta = \lambda mR^2$. Például, ha a kerék egy homogén lemez volna, akkor $\lambda = 1/2$-re adódna.

Figyelembe véve a gördülés kényszerét, a szögsebesség $\omega = -v/R$. A tehetetlenségi nyomaték és a szögsebesség behelyettesítésével kapott egyenletet már megoldhatjuk az ismeretlen sebességre:

$$\frac{1}{2}mv^2 + \frac{1}{2}\lambda mR^2\omega^2 = \frac{1+\lambda}{2}mv^2 = mg\big(y(0) - y(\tau)\big) \;\;\rightarrow\;\; v = \sqrt{\frac{2g\big(y(0) - y(\tau)\big)}{1+\lambda}}.$$



A görbeparaméter és az idő kapcsolata.

A τ paraméter tehát a mozgásállapotot teljes mértékben meghatározza, így a szimulációnak azt kell kiszámolnia, hogy Δt kis időlépés alatt a τ paraméter milyen $\Delta\tau$-val változik. A görbeparaméter és eltelt idő összekapcsolását az időlépés alatt megtett Δs út teremti meg. A sebesség ismeretében egyrészt az út $\Delta s = v\,\Delta t$. Másrészt, az $\boldsymbol{r}(\tau)$ és $\boldsymbol{r}(\tau + \Delta\tau)$ pontok távolsága is közelítőleg ugyanez az út:

$$\Delta s \approx |\boldsymbol{r}(\tau + \Delta\tau) - \boldsymbol{r}(\tau)| \approx |\dot{\boldsymbol{r}}(\tau)|\Delta\tau \;\;\rightarrow\;\; \Delta\tau = \frac{\Delta s}{|\dot{\boldsymbol{r}}(\tau)|} = \frac{v\,\Delta t}{|\dot{\boldsymbol{r}}(\tau)|}.$$

3. Előfeltételek

Előfeltétel a „Geometriai modellezés” és „Szabadformájú görbék” előadások feldolgozása, valamint a „2d képszintézis” objektum-vezérelt csővezeték ismerete. Javasolt még görbület fogalmának felidézése is.

4. Alapok

1. A kameraablakot világkoordináta rendszerben a közepével és a vízszintes/függőleges méretével adjuk meg. Ezekből számítsa ki a kamera View és Projection transzformációs mátrixokat és inverzüket (a mátrixokra szükség lesz a csúcspontárnyaló transzformációs mátrixának beállításához, az inverzre pedig a bemeneti csővezeték implementálása során).
2. Adja meg Catmull-Rom spline $\boldsymbol{r}(\tau)$ paraméteres egyenletét és annak $\dot{\boldsymbol{r}}(\tau)$ deriváltját a τ paraméter szerint. A τ paraméter most nem az idő, és nem is a pályán megtett távolság.
3. Határozza meg a csomóértékek értékét abból a feltételből, hogy a spline uniform.
4. A $\boldsymbol{r}(\tau)$ Catmull-Rom spline τ szerinti deriváltjának felhasználásával adja meg a $\boldsymbol{T}$ egység hosszú érintővektort és $\boldsymbol{N}$ egység hosszú normálvektort a τ függvényében. A két vektor és az aktuális pozíció felhasználásával fejezze ki modellezési transzformációs mátrixot.
5. Fejezze ki a vasút v pályamenti sebességének nagyságát az energiamegmaradás felhasználásával a τ paraméter függvényében.
6. A v pályamenti sebesség ismeretében adja meg, hogy Δt idő alatt a vasúti kocsi mennyire tolódott el, mennyi paraméter $\Delta\tau$ megváltozása ezalatt, és a kocsi mennyit fordult a középpontja körül (azaz a $\boldsymbol{T}$ mennyit változott). Célszerű a következő közelítés használata a Δt idő alatt megtett útra, ami alatt a paraméter $\Delta\tau$-t változik (Ezt kell használni arra, hogy a szimuláció során kapott Δt időlépésből a gondola helyét és állapotát meghatározó τ paraméter új értékét megkapjuk a paraméter $\Delta\tau$-vel növelésével):
$$|\dot{\boldsymbol{r}}(\tau)|\Delta\tau \approx |\boldsymbol{r}(\tau+\Delta\tau)-\boldsymbol{r}(\tau)| \approx v\Delta t$$
7. A v pálya menti sebesség valamint a szögsebesség (ami az elfordulási szög és a Δt hányadosa) felhasználásával fejezze ki a mozgáshoz szükséges centripetális gyorsulást, és döntse el, hogy a nehézségi erő normálirányú komponense és a pálya normálirányú nyomóereje kiadhatja-e ezt a centripetális erőt. Ha nem, mert a nyomóerőnek ehhez húznia kellene, akkor a kocsi lerepül a pályáról.
8. Alternatív lehetőségként határozza meg a görbületet az adott pontban, majd a görbület és a sebesség négyzetének szorzatával számítsa ki a centripetális gyorsulást.

4. Programozás

4.1. Camera osztály

Implementáljon egy Camera osztályt, amely a kamerát világ-koordinátarendszerben a kameraablak középpontjával és vízszintes/függőleges méretével tartja nyilván. A kamera műveletei a View és Projection transzformációk és inverzeik kiszámítása, valamint a bemeneti csővezeték transzformációjának megvalósítása, amely az operációs rendszer pixelkoordinátáiból (a kurzor kapott helye) világkoordinátákba visz át.

4.2. Csúcspont árnyaló

A csúcspont árnyaló a kamera által kiszámított Modell-View-Projection mátrixszal transzformálja a csúcspontokat.

4.3. Pixelpont árnyaló

A pixelárnyaló az uniform változóként kapott színnel színezi a pixelt.

4.4. Spline osztály

A hullámvasút pályájához implementáljon egy Spline osztályt, amely a kontrollpontokat tárolja. A csomóértékeket automatikusan számítja. A kontrollpontok alapján a görbe $\boldsymbol{r}(\tau)$ paraméteres egyenletét kiértékeli, és a paramétertartományt 100 kis részre felosztva, a vektorizált görbét dinamikusan a GPU egy VAO/VBO-jába tölti. A Draw hívás hatására a görbét maximális intenzitású sárga színnel és GL_LINESTRIP primitívtípussal a GPU csővezetéken végigzavarja, majd a kontrollpontokat rajzolja fel maximális intenzitású pirossal és 10-es pontmérettel.

4.5. Gondola osztály

A kocsit ábrázoló körhöz vegyen fel egy új osztályt, ami a kékkel kitöltött és fehér kontúrral rendelkező kört és a küllőket tartalmazza, valamint a fizikai animációhoz szükséges változókat, az állapotot (várakozik, indított, leesett), és a pálya referenciáját tartalmazza. A konstruktor a tartalmazott geometriai primitíveket létrehozza. A Start elindítja a kocsit a pályán zérus kezdősebességgel. Az Animate *dt* idővel lépteti a kocsi állapotát elvégezve a fizikai animációt. A Draw a pozíciónak és elfordulási szögnek megfelelően beállítja a vertex árnyaló transzformációs mátrixát és felrajzolja a felépítő geometriai primitíveket.

4.6. onKeyboard eseménykezelő

Az onKeyboard függvényben SPACE lenyomására indítsa útjára a kocsit.

4.7. onMouse eseménykezelő

Az onMouse függvényt implementálja úgy, hogy az operációs rendszerbeli pixelkoordinátákban kapott kurzor pozíciót alakítsa át világkoordinátákká, és a bal egérgomb lenyomásának hatására vegyen fel az aktuális görbének egy újabb kontrollpontot ezen a helyen, végül pedig érvénytelenítse az ablakot (glutPostRedisplay).

4.8. onIdle eseménykezelő: diszkrét idő szimuláció

Az onIdle érkezési gyakoriságára nincs garancia, miközben mi a differenciálhányadost a görbület számításánál differenciahányadossal közelítettük, ami kis lépéseknél jó csupán. Ezért, ha az onIdle túl sok idővel kívánja felzárkóztatni a hullámvasút állapotát, akkor az időintervallumot kis képésekre kell bontani és azokban egyenkét végrehajtani az animációt. Például így:

```
void onIdle() { // Idle event indicating that some time elapsed: do animation here
	static float tend = 0;
	const float dt = 0.01; // dt is ”infinitesimal”
	float tstart = tend;
	tend = glutGet(GLUT_ELAPSED_TIME) / 1000.0f;
	for (float t = tstart; t < tend; t += dt) {
		float Dt = fmin(dt, tend - t);
		gondola->Animate(Dt);
	}
	glutPostRedisplay();	// redraw the scene
}
```