

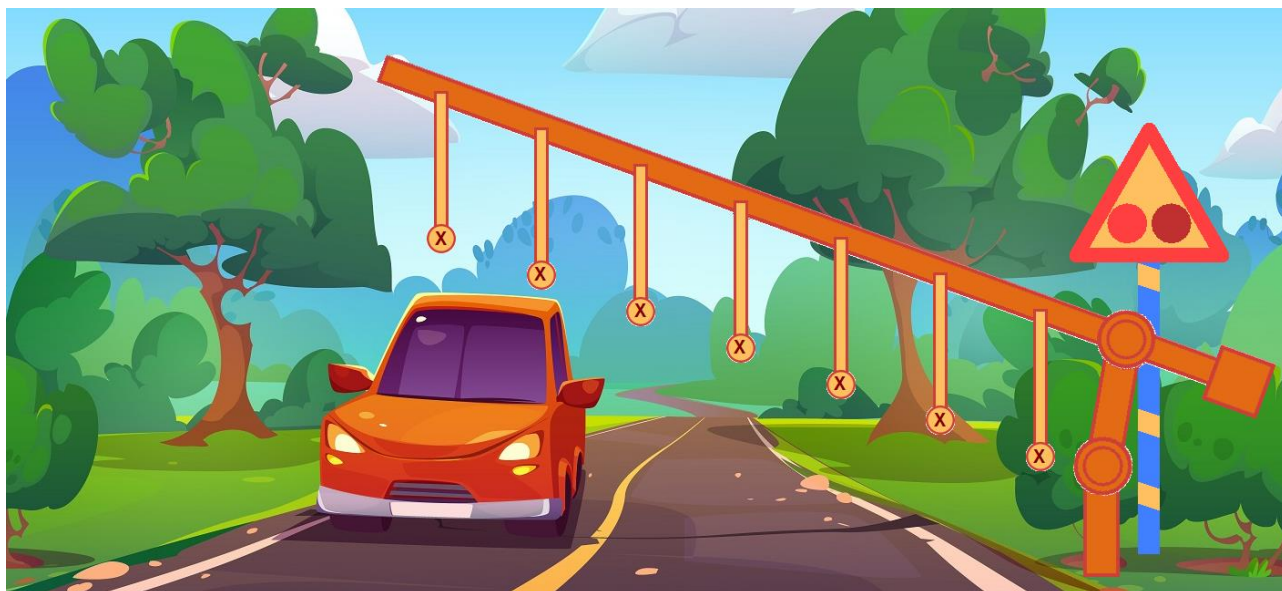
ПРЕДМЕТ:

РАЧУНАРСКА ГРАФИКА

31.1.2026.

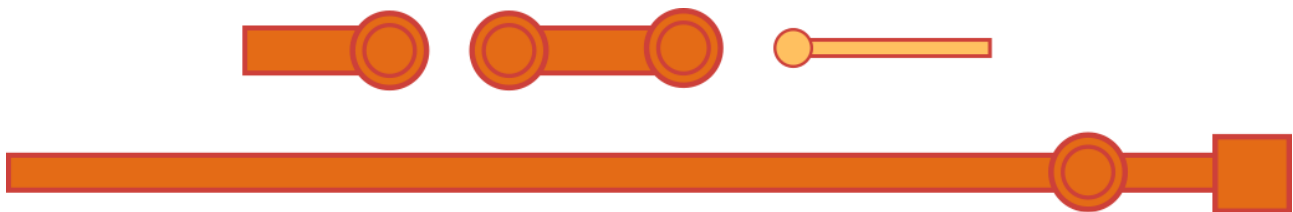
## I КОЛОКВИЈУМ (GDI) – Б ГРУПА

**Задатак 1.** Коришћењем GDI-а имплементирати следеће:



1. У конструктору класе погледа, коришћењем класе `DImage`, учитати слике **Background.jpg**, **Pole1.bmp**, **Pole2.bmp**, **BarrierB.bmp** и **StripB.bmp**. Ослобађање меморије заузете сликама обезбедити кроз деструктор. [4 бода]
2. Обезбедити да се приказ исцртава без треперења (*flicker*) коришћењем меморијског DC-a. [8 бодова]
3. Написати функцију `void DrawBackground(CDC* pDC)`, која исцртава слику **Background.jpg** у оригиналној величини тако да њена доња ивица буде поравната са доњом ивицом приказа и тако да средиште слике буде за 100 пиксела улево од средишта приказа по хоризонталној оси без обзира на величину прозора. [5 бода]
4. Написати функцију `void DrawImageTransparent(CDC* pDC, DImage* pImage)`, која исцртава слику `pImage` у оригиналној величини са транспарентном позадином, при чему се боја позадине поклапа са бојом првог пиксела у слици. За транспарентно исцртавање слике `pImage` искористити постојећу функцију `DrawTransparent` класе `DImage`, која има прва два параметра иста као функција `Draw`, а последњи параметар је боја која треба да буде транспарентна. [5 бода]
5. Написати функције којима се мењају светске трансформације множењем текуће трансформационе матрице са стране одређене последњим параметром [8 бода]:
  - a. `void Translate(CDC* pDC, float dX, float dY, bool rightMultiply)`
  - b. `void Rotate(CDC* pDC, float angle, bool rightMultiply)`
  - c. `void Scale(CDC* pDC, float sX, float sY, bool rightMultiply)`

6. Написати функцију `void DrawTrafficSign(CDC* pDC, int a, int h, int n, bool light[])` којом се црта саобраћајни знак за пружни прелаз. Знак је нацртан тако да се средиште доње основице стуба налази у координатном почетку. Стуб је ширине  $0.15 \cdot a$  и висине  $h$ . Основна боја стуба је светло плава (64, 128, 255) и на њему се налази  $n$  штрафи жуте боје (255, 192, 96). Висина једне штрафте је  $0.2 \cdot h / n$ , док је свака штрафта укошена за половину своје висине. Стуб нема оквир. На врху стуба налази се знак у облику једнакостраничног троугла стране  $a$ , Оквирна линија троугла је дебљине  $0.15 \cdot a$  црвене боје (255, 64, 64), док је унутрашњост жуте боје (255, 192, 96). На знаку се налазе два светла облика круга полупречника  $r = 0.14 \cdot a$  постављена симетрично тако да им је центар удаљен  $1.1 \cdot r$  од осе симетрије. Кругови су померени за  $1.6 \cdot r$  од доње основе троугла. Кругови треба буду испуњени једном од две нијансе црвене боје, тамнијом (255, 64, 64) или светлијом (192, 48, 48). Којом бојом ће бити обојени кругови дефинише низ **light** који има два елемента, за свако од светала (кругова). Нетачна вредност одговара тамнијој нијанси црвене боје, а тачна светлијој. Избор боја ће се мењати помоћу одговарајућих тастера. [20 бодова]



7. Написати функцију `void DrawStrip(CDC* pDC, CString s)` којом се црта висећа трака са тегићем облика круга. Трака треба да буде нацртана усправно наниже. За цртање траке користити слику **StripB.bmp** у оригиналној величини. Трака је нацртана тако да се средиште горње основице траке налази у координатном почетку. На средини тегића (на растојању 16 пиксела од врха траке) исписати текст  $s$ . Текст исписати подебљаним фонтом Arial, висине 24 пиксела тамно црвене боје (128, 0, 0). Текст исписати тако да буде усправан. [16 бодова]
8. Написати функцију `void DrawBarrierBoom(CDC* pDC, CString s, int nStrip, double angle)` којом се црта пречка рампе са висећим тракама. Пречку рампе нацртати коришћењем слике **BarrierB.bmp** у оригиналној величини, тако да центар кругова са слике око кога се рампа окреће буде у координатном почетку. Центар кругова је удаљен 870 пиксела. На растојању од 100 до 800 пиксела у односу на координатни почетак на подједнаком растојању је распоређено  $nStrip$  висећих трака. За њихово цртање користити функцију из претходне ставке. Висеће траке треба да буду окренуте за угао **angle** у односу на правац дуж осе рампе. Број висећих трака се може мењати одговарајућим тастерима. [14 бодова]
9. Написати функцију `void DrawRailwayBarrier(CDC* pDC, CString s, int nStrip, double alpha, double beta)` којом се црта целокупна рампа. Рампа је нацртана тако да се средиште доње основе налази у координатном почетку. Рампа се састоји од основе коју треба насртати коришћењем слике **Pole1.bmp** у оригиналној величини. Око центра кругова са слике основе који се налази на растојању 122 пиксела може се окретати носач. Носач окренути за угао **alpha**. Носач нацртати коришћењем слике **Pole2.bmp** у оригиналној величини тако да се ротира око центра својих левих кругова који се налази на растојању 32 пиксела. Око центра десних кругова са слике основе који се налази на растојању 140 пиксела може се окретати пречка рампе. Пречку рампе исцртати функцијом из претходне ставке тако да буде окренута за угао **beta**. Обезбедити да се при цртању пречке рампе висеће траке увек исцртавају усправно. Углови се могу мењати одговарајућим тастерима. [14 бодова]

10. Нацртати целу сцену исцртавањем позадине као што је наведено у ставки 3. Саобраћајни знак нацртати тако да је центар његовог стуба на растојању 550 пиксела по хоризонталној оси од центра прозора и 50 пиксела по вертикалној оси од доње ивице прозора. Висину стуба знака **h** поставити на 320 пиксела, старицу троугла **a** на 160 пиксела и број штрафти на 4. Рампу нацртати тако да је центар њене основе на растојању 500 пиксела по хоризонталној оси од центра прозора и 25 пиксела по вертикалној оси од доње ивице прозора. На тегићима исписати слово X. Обезбедити могућност промене угла *alpha* (носача у односу на основу рампе) на тастере A и S, *beta* (пречке у односу на носач рампе) на тастере D и F, броја *nStrip* висећих трака на рампи на тастере Q и W, укључивање знака за долазак воза на тастер E и промену које ће светло на знаку сијати на тастер R. [6 бода]

**Напомена:** Пројекти који се не компајлирају или не приказују ништа у оквиру прозора неће бити оцењивани. По завршетку, читав пројекат запаковати у архиву са називом који садржи број индекса, име, презиме и број задатка (нпр. **12345\_Pera\_Peric\_Z2.zip**), и снимити на флеш меморију дежурног асистента. Пре креирања архиве, из пројекта обрисати **sdf** датотеку, као и све **Debug**, **Release**, **.vs** (скривен је) и **ipch** директоријуме (ако постоје у пројекту).