

## Полезни функции на библиотеката GLUT

Други полезни функции от библиотеката GLUT са:

1. **int glutCreateWindow(char\*name)** – създава прозорец, но и връща неговия идентификационен номер;
2. **glutDestroyWindow(int win)** – унищожавя прозореца, определен от идентификационния номер win;
3. **int glutGetWindow(void)** – връща идентификационния номер на текущия прозорец;
4. **glutInitWindowPosition(int x, int y)** – определя началното разположение на прозореца, като приема два аргумента, които се отнасят за горния ляв ъгъл на прозореца. Имайте предвид, че при операционна система Windows най-горната лява точка от екрана на монитора има координати (0, 0). Специално функцията glutInitWindowPosition( ) трябва да бъде извикана преди функцията glutCreateWindow( ), която изгражда прозореца.;
5. **glutPositionWindow(int x, int y)** – променя позицията на създадения вече прозорец в зависимост от подадените от нас аргументи;
6. **glutReshapeWindow (int width, int height)** – променя позицията на създадения прозорец, в зависимост от подадените от нас аргументи;
7. **glutSetWindowTitle (char\*name)** – определя/променя заглавието на прозореца в зависимост от подадения аргумент.

Следващите функции трябва да бъдат извикани задължително след създаването на прозореца с glutCreateWindow и преди започване на цикъла на съобщенията, т.е. преди функцията glutMainLoop ( ):

8. **glutFullScreen(void)** – оразмерява прозореца на цял екран;
9. **glutReshapeFunc(void (\*func)(int width, int height))** – определя функцията, която се извиква, когато прозореца промени своя размер. На нашата функция се подават нови размери на прозореца.;
10. **glutSetCursor(int cursor)** – променя курсора на мишката в зависимост от подадения аргумент, който може да бъде една от константите:
  - GLUT\_CURSOR\_RIGHT\_ARROW – наклонена на дясно стрелка;
  - GLUT\_CURSOR\_LEFT\_ARROW – наклонена на ляво стрелка;
  - GLUT\_CURSOR\_INFO – сочеща ръка;
  - GLUT\_CURSOR\_DESTROY – череп с кръстосани кости;
  - GLUT\_CURSOR\_HELP – стрелка с въпросителен знак;
  - GLUT\_CURSOR\_CYCLE – кръг с пресечена черта;
  - GLUT\_CURSOR\_SPRAY – спрей;
  - GLUT\_CURSOR\_WAIT – пясъчен часовник;
  - GLUT\_CURSOR\_TEXT – знака “I”, когато добавяте текст;
  - GLUT\_CURSOR\_CROSSHAIR – кръст;
  - GLUT\_CURSOR\_UP\_DOWN – две противоположно сочещи вертикални стрелки, свързани в една цяла – появява се, когато искате да промените размера на даден прозорец;
  - GLUT\_CURSOR\_LEFT\_RIGHT – две противоположно сочещи хоризонтални стрелки, свързани в една цяла;
  - GLUT\_CURSOR\_TOP\_SIDE – стрелка, сочеща нагоре;

- GLUT\_CURSOR\_BOTTOM\_SIDE – стрелка, сочеща надолу;
- GLUT\_CURSOR\_LEFT\_SIDE – стрелка, сочеща наляво;
- GLUT\_CURSOR\_RIGHT\_SIDE – стрелка, сочеща надясно;
- GLUT\_CURSOR\_TOP\_LEFT\_CORNER – стрелка, сочеща към горния ляв ъгъл;
- GLUT\_CURSOR\_TOP\_RIGHT\_CORNER – стрелка, сочеща към горния десен ъгъл;
- GLUT\_CURSOR\_BOTTOM\_LEFT\_CORNER – стрелка, сочеща към долния ляв ъгъл;
- GLUT\_CURSOR\_BOTTOM\_RIGHT\_CORNER – стрелка, сочеща към долния десен ъгъл;
- GLUT\_CURSOR\_NONE – невидим курсор;

11. **glutKeyboardFunc(void(\*func)(unsigned char key, int x, int y))** – определя коя функция да бъде извикана при натискане на клавиш от клавиатурата. Тя трябва да е от вида void OurFunction (unsigned char key, int x, int y). На нашата функция се предава ASCII кода на натиснатия клавиш и координатите на мишката.;
12. **glutSpecialFunc(void (\*func)(unsigned char key, int x, int y))** – определя коя функция трябва да бъде извикана, ако бъде натиснат някой от специалните клавиши. Тя трябва да бъде от вида void OurFunction (int key, int x, int y). Първият аргумент може да бъде от константите: GLUT\_KEY\_F1,..., GLUT\_KEY\_F12, GLUT\_KEY\_LEFT, GLUT\_KEY\_RIGHT, GLUT\_KEY\_UP, GLUT\_KEY\_DOWN, GLUT\_KEY\_PAGE\_UP, GLUT\_KEY\_PAGE\_DOWN, GLUT\_KEY\_INSERT, GLUT\_KEY\_END. Другите два аргумента определят положението на курсора на мишката.;
13. **glutMouseFunc(void (\*func)(int button, int state, int x, int y))** – определя функцията, която се извиква при натискане/отпускане бутон на мишката. Тя трябва да е от вида void OurFunction(int button, int state, int x, int y). Първият от подадените аргументи може да е равен на една от константите: GLUT\_LEFT\_BUTTON, GLUT\_RIGHT\_BUTTON и GLUT\_MIDDLE\_BUTTON. Вторият аргумент (state) може да е равен на GLUT\_DOWN, ако бутонът е натиснат или на GLUT\_UP, ако бутонът е отпуснат. Последните два аргумента определят местоположението на курсора на мишката;
14. **glutMotionFunc(void(\*func)(int x, int y))** – определя функция, която се извиква ако мишката се премести, докато някой от бутоните ѝ е натиснат. Двата аргумента определят местоположението на мишката.;
15. **glutPassiveMotionFunc(void(\*func)(int x, int y))** – определя функция, която се извиква, ако мишката се премести докато не е натиснат никой от бутоните ѝ;
16. **glutEntryFunc(void(\*func)(int state))** – определя функция, която се извиква, когато курсора напусне прозореца или влезе отново в него. Подаденият аргумент може да бъде една от двете константи GLUT\_LEFT(ако курсорът излиза от прозореца) или GLUT\_ENTERED(ако курсорът влиза в прозореца).;
17. **glutIdleFunc (void (\*func)(void))** – определя функция, която се извиква, ако няма възникнали събития, т.е. ако никоя от горните функции не бъде извикана.;
18. **glutTimerFunc(unsigned int msec, void (\*func)(int value), int value)** – определя функция, която се извиква след определено време. Първият аргумент (**msec**) определя интервала на време в милисекунди, а последният аргумент (**value**) се подава на нашата функция, която се извиква от **glutTimerFunc ( )**. Тя трябва да бъде от вида void OurFunction (int value).

## Създаване на менюта:

Библиотеката GLUT предлага няколко функции, с които бързо и лесно можете да създадете различни контекстни менюта. Функциите, с които се създава меню са:

- `int glutCreateMenu(void(*func)(int value))` – тя връща идентификационния номер на създаденото меню, който трябва да е от тип `int` и определя функция, която трябва да бъде извикана при избиране на елемент от менюто. Функцията трябва да е от вида `void OurFunction (int value)`. Подаденият ѝ аргумент е идентификационния номер на избрания елемент на менюто.;
- `void glutSetMenu(int menu)` – определя текущото контекстно меню, в зависимост от подадения идентификационен номер.;
- `int glutGetMenu (void)` – връща идентификационния номер на текущото контекстно меню.;
- `void GlutDestroyMenu(int menu)` – унищожава контекстното меню, чиито идентификационен номер сме подали като аргумент на функцията;
- `void glutAddMenuEntry(char*name, int value)` – добавя елемент към текущото контекстно меню. Първият аргумент определя името на аргумента, което ще видим в менюто, а вторият – идентификационния номер на елемента. Обикновено първият елемент има идент. Номер 1, вторият – 2 и т.н.;
- `void glutAddSubMenu(char*name, int menu)` – добавя подменю към текущата контекстно меню. Първият аргумент определя текста, който ще се покаже в текущото меню, от който ще излиза подменюто при задържане на курсора на мишката върху него. Вторият аргумент приема идентификационния номер на вече създаденото меню, което искаме да превърнем в подменю на текущото меню.;
- `void glutRemoveMenuItem(int entry)` – премахва елемент от менюто с пореден номер `entry`. Първият елемент има пореден номер 1. Подменютата са също елементи на главното меню, така че те имат поредни номера и могат да бъдат премахвани.;
- `void glutAttachMenu(int button)` – определя с кой бутон на мишката да се пуска определено контекстно меню. Аргумента `button` може да бъде една от константите `GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON` или `GLUT_RIGHT_BUTTON`.;
- `void glutDetachMenu(int button)` – премахва възможността на дадено меню да бъде пуснато със съответния бутон, определен от аргумента `button`.

Преди да създадете меню, трябва да дефинирате променлива от тип `int`, която ще се използва за съхранение на идентификационния номер на менюто. След това създаваме менюто: `ID = glutCreateMenu(OurFunction)`, където `ID` е нашата променлива. След това трябва да определим това меню като текущо и извикваме `glutSetMenu(ID)`. Вече можем да добавяме елементи с `glutAddMenuEntry( )` или подменюта с `glutAddSubMenu( )`. Накрая извикваме функцията `glutAttachMenu(int button)`, за да определим с кой бутон на мишката да се пуска менюто. Ако искаме можем да имаме три действащи менюта, които да се пускат с трите бутона на мишката. Просто извършваме гореописаните стъпки за всяко меню поотделно.