

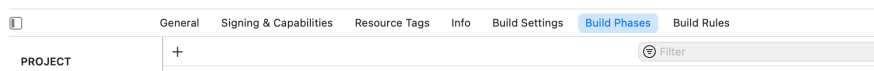
24780 Group Project
Final Documentation
Team Name: Water Army
12/6/2023

List of external toolkits:




- Cocoa framework
- openGL framework
- GLUT framework
- AVFoundation framework

Compile instruction using X-Code (Recommended):















- Create an X-Code project and remove AppDelegate.h, AppDelegate.c, main.c, and Sand Box.
- Copy source files into the project.
 - If X-Code asks “Would you like to configure an Objective-C bridging header?” Select No.
- Go to Build Phase of project settings.



- Replace items in Link Binary With Libraries with these 3 items.

Link Binary With Libraries (3 items)		
Name	Status	
 OpenGL.framework	Required ↕	
 Cocoa.framework	Required ↕	
 GLUT.framework	Required ↕	
+ —		Drag to reorder linked binaries

- Make sure Compile Sources includes these 14 items.

Compile Sources (14 items)		
Name	Compiler Flags	
 yssimplesound_macosx_objc.m ...in final		
 breakbricks.cpp ...in final		
 gomoku.cpp ...in final		
 yspng.cpp ...in final		
 fishing.cpp ...in final		
 yssimplesound.cpp ...in final		
 fssimplewindowobjc.m ...in final		
 Menu.cpp ...in final		
 starwar.cpp ...in final		
 textstring.cpp ...in final		
 fssimplewindowcpp.cpp ...in final		
 GoldMiner.cpp ...in final		
 textinput.cpp ...in final		
 ysglfontdata.c ...in final		
+ —		

- Enjoy our games!

Compile instruction using Terminal (Not Recommended):

- PS: This method does not guarantee successful keyboard interaction.
- Running the compilation command requires the use of the external libraries listed above. If any of the libraries has not been installed, then the compilation process will not proceed.
- G++ Command for compiling C++ source files (with C++ 11), which also specifies the use of external frameworks that are use for graphic and user interface:
g++ -std=c++11 -c breakbricks.cpp fishing.cpp fssimplewindowcpp.cpp GoldMiner.cpp gomoku.cpp Menu.cpp starwar.cpp textstring.cpp ysglfontdata.c yspng.cpp yssimplesound.cpp -framework Cocoa -framework OpenGL -framework GLUT
- Clang Command for compiling Objective-C source Files:
clang -framework Cocoa -framework OpenGL -framework GLUT -c fssimplewindowobjc.m yssimplesound_macosx_objc.m
- G++ command for linking all the object files together and creating a executable file named “final_project_exec”:
g++ -o final_project_exec -framework Cocoa -framework OpenGL -framework GLUT -framework AVFoundation breakbricks.o fishing.o fssimplewindowcpp.o GoldMiner.o gomoku.o Menu.o starwar.o textstring.o ysglfontdata.o yspng.o yssimplesound.o fssimplewindowobjc.o yssimplesound_macosx_objc.o
- Execute the created file: “final_project_exec”: `./final_project_exec`
- Enjoy our games!