User manual

Sections

1. **Pacman controls**
2. **Make own games**

The menus are controlled with the mouse. Check the drivers for your operating system for use with Xbox360 Controller. On Windows these are accessible from the update center, I haven't tested on the other operating systems, but I read somewhere that xpad for Linux might work.

If by any chance the PacmanGame.jar isn't inside the folder games/PacmanGame/ you'll have to manually copy it there, from the dist folder inside PacmanGame project folder.

Pacman is controlled as follows:

Action	**Keyboard**	**Xbox360 Controller**
Up	W	Left analog stick
Down	S	Left analog stick
Left	A	Left analog stick
Right	D	Left analog stick
Toggle Debug	F1	Right Bumper
When in debug		
Victory Scene	F11	Start
Game Over Scene	F12	Back
Rotate Level Left	G	Left Trigger
Rotate Level Right	H	Right Trigger
Eat Super Snack	Space	A

Making your own game

When you want to extend the GameHub with more arcade like games by yourself, you can easily do this by following these few steps.

1. Create a new Project
2. Create the Main class at the root of the default package
3. Extend the Main class with the abstract Game class inside the Engine.jar
4. Start to make your game. Now you can use what kind of package names you like.
5. Create a runnable jar from your game, name it the same as the Main class
6. Place the runnable jar inside a folder with the same name as the runnable jar, inside the games folder next to the GameHub.jar
7. Start GameHub.jar, it will now load your game automatically. Select your game and start it by clicking the start button.
8. If you want to make your game runnable independently (without the use of the GameHub), then you'll have to include a main method that looks like this:

```
public static void main(String[] args) {
      Screen.setupNativesLWJGL();
      new Thread() {
          @Override
          public void run() {
              new YourGame().start();
          }
      }.start();
  }
```

The Screen class is found inside the Engine.jar, and the game is started inside a different thread, just to make several instances of the GlContext runnable at the same time.

If you want to make your own tests, you should call this method before starting the tests, just to enable the GlContext before trying to use OpenGL stuff.

```
@BeforeClass
public static void init() {
          Screen.setupNativesLWJGL();
          Screen.setupDisplay("PacmanTest");
          Screen.setupLWJGL();
}
```