```
AbstractGame g;
if(type.equals("asteroid") {
  g = new AsteriodGame();
}
else if(type.equals("tux")) {
  g = new TuxRunner();
}
g.run();
```

AracadeGameFramework
___
+ void run(String type)
...

```
frame = new JFrame("Title");
displayPanel = new JPanel();
statusPanel = new JPanel();

JPanel cPane =(JPanel)frame.getContentPane();
cPane.add(displayPanel, BorderLayout.CENTER);
cPane.add(statusPanel, BorderLayout.SOUTH);

setupDisplay(displayPanel);
setupStatus(statusPanel);
```

«abstract»
AbstractGame
___
- JFrame frame
- JPanel displayPanel
- JPanel statusPanel
...
___
- void setup(JFrame f)
void setupAdditional(JFrame f)
void setupDisplay(JPanel p)
void setupStatus(JPanel p)
int readKeyPress()
void handle(int key)
- boolean gameOver()
+ void run()
...

```
while(!gameOver()) {
  int key = readKeyPress();
  handle(key);
}
```

setupAdditional(frame);

AsteroidGame
___
- JPanel controlPanel
...
___
+ AsteroidGame()
# void setupDisplay(JPanel p)
# void setupStatus(JPanel p)
- void setupControl(JPanel p)
int readKeyPress()
void handle(int key)
void setupAdditional(JFrame f)
...

TuxRunner
___
...
___
+ TuxRunner()
# void setupDisplay(JPanel p)
# void setupStatus(JPanel p)
# int readKeyPress()
# void handle(int key)
...

```
setup(frame);
frame.setTitle("Awesome Tux Runner");
frame.setVisible
```

```
setup(frame);
frame.setTitle("Amazing Asteroid");
controlPanel = new JPanel();

JPanel cPane = (JPanel)frame.getContentPane();
cPane.add(controlPanel, BorderLayout.NORTH);

setupControl(controlPanel);

frame.setVisible(true);
```

The ArcadeGameFramework has an AbstractGame object and instantiates it at runtime to run the specified game. All of the duplicate setup code has been implemented in the AbstractGame class. If additional JPanels need to be setup, then each class can implement the setupAdditional

method. For instance, the AsteroidGame had a control panel that needs to be setup so it implements setupAdditional to do this. Any methods that could have custom implementations such as setupDisplay, setupStatus, readKeyPress, and handle are all implemented in the subclasses. The AbstractGame delegates the logic for these methods to its subclasses. The framework does need to be modified though if additional games are added…