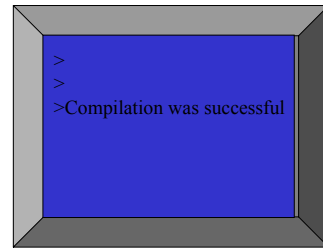
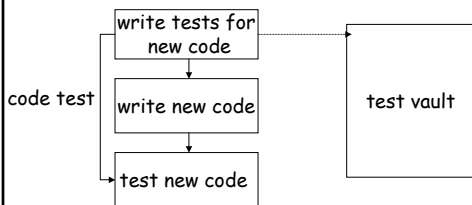


Test Plans

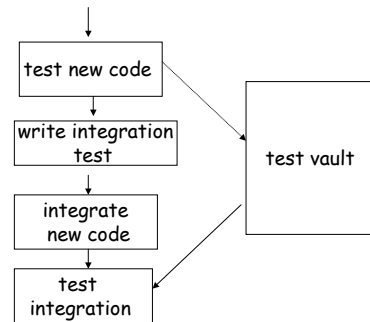


This does not mean the code is debugged!

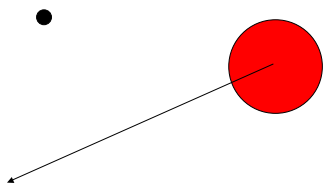
Test Driven Development I



Test Driven Development II



Unit Test for Vertex Collision



Test Cases

- You know answer ahead of time
- Tests include problem cases

Step 1

v

p_0

choose v and p_0 randomly

1. distance $< r$
2. distance $= r$
3. distance $> r$

Step 2

v

p_0

choose random vector unit w
compute path endpoints $v+(r-\delta)w$,
 $v+rw$, and $v+(r+\delta)w$

w

Step 3

v

p_0

throw out test case if dot product
is positive.

Step 4

v

p_0

$0 < \alpha < 1$

$\alpha \approx 1$

$\alpha = 1$

w

Step 5

v

p_0

$\alpha \approx 0$

$\alpha = 1$

w

test

- One jump positive tests
- Two jump positive tests
- Two jump negative tests

Questions

- Are these good test?
- Are there other kinds of tests we should do?
- How can we automate testing?
 - Test generation
 - Running the tests

Devise test plans for standardized edge and face tests.

Triangle World II

1. Extend edge and face collisions to general case
2. Integrate vertex, edge, and face collisions
3. Handle up to 3 triangles
(this is a change from overview)
4. Test plan