2D Simulation of Rigid Bodies

Alan Hazelden

Motivation

- Almost all games involve physics
  - Pong, Mario
- Almost all game objects are rigid bodies
  - Detailed representation not required

Project Goals

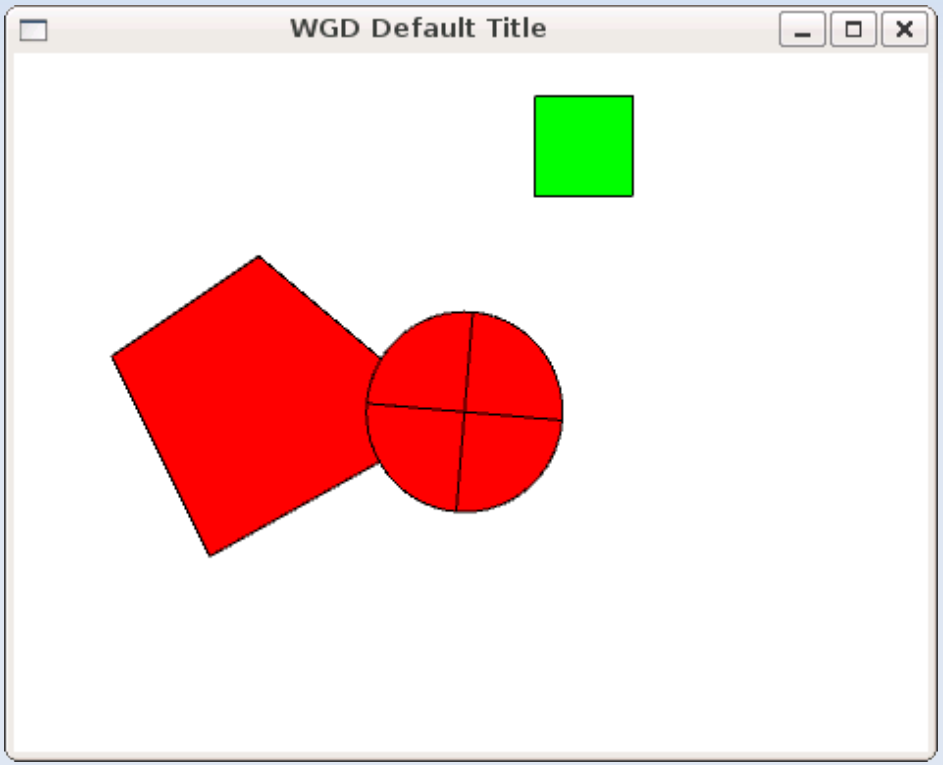
- Realistic looking behaviour
- Primitive objects:
  - circles
  - convex polygons
- Compound bodies
  - union of two or more primitive shapes
- Friction

Choice of Language

- Written in C++
- Using the Warwick Game Design C++ library
  - Sets up the display window and graphics
  - Internal database – DOSTE
    - Easy saving/loading of objects and world state
    - Event handlers
    - In-game console

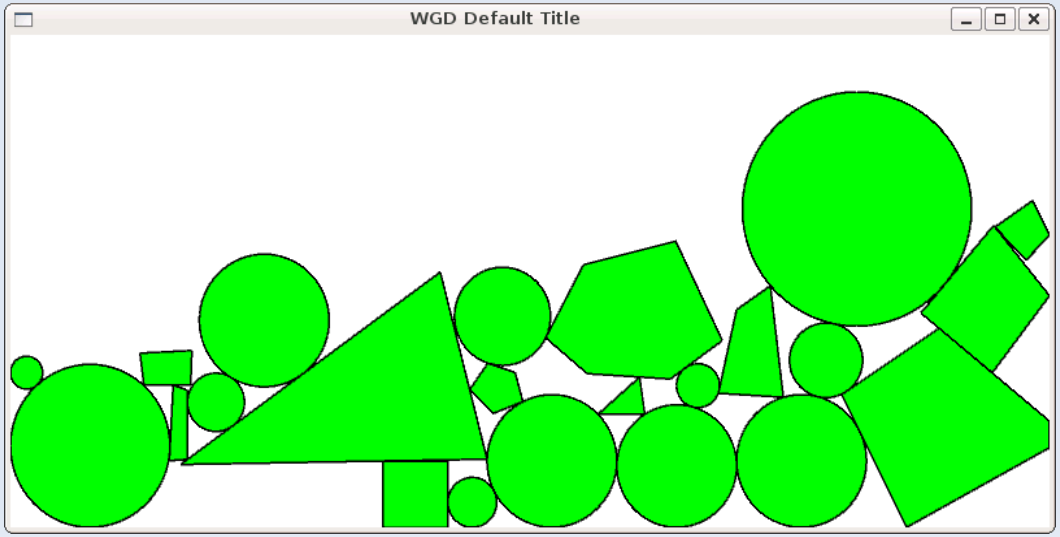
Implementation

- Creating and drawing objects
- Collision detection



Implementation

- Collision resolution
  - Conservation of momentum
  - Coefficient of restitution
  - No angular movement
- Resting contacts

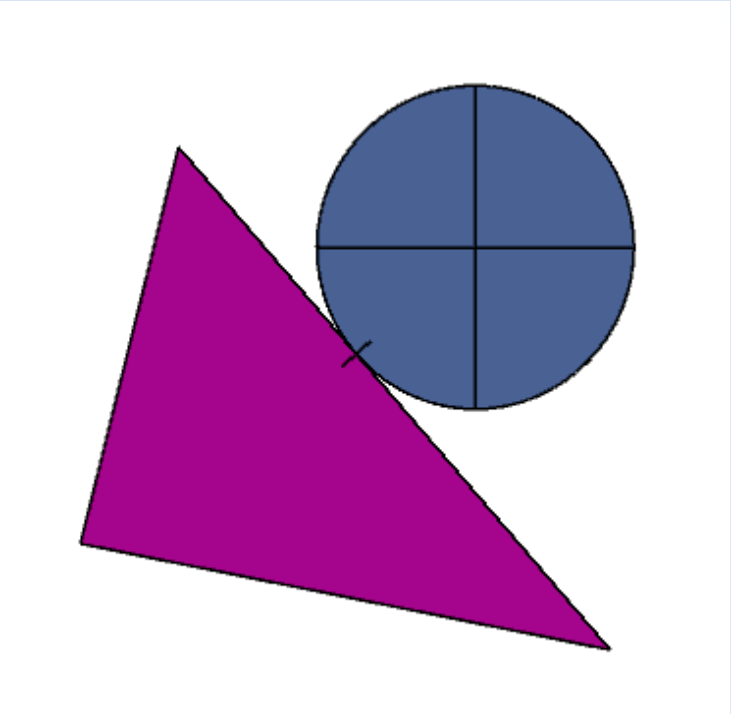


Implementation

- Rotation
  - Must ensure that no energy is added to the system
- Resting contacts
  - No longer stable
  - Problems appear with many stacked objects
- Solution: more contact points
  - More time spent processing

Implementation

- Friction
  - Forces applied at right angles to the contact normal



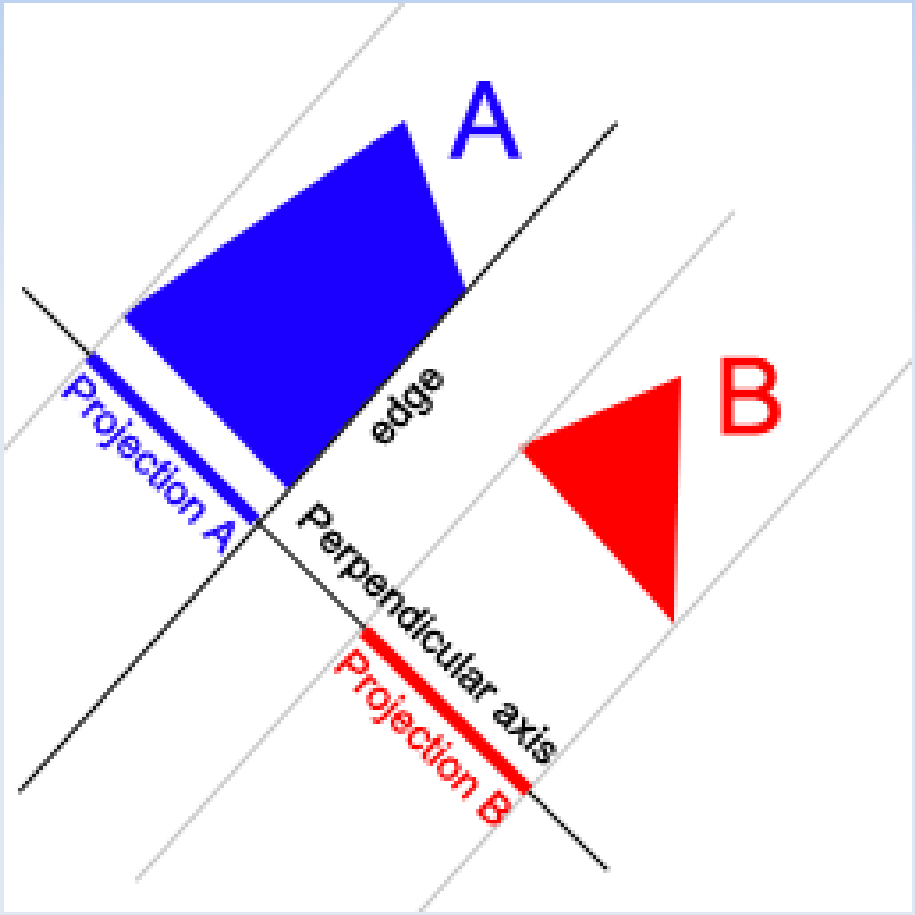
The final result

- Every frame:
  - All bodies are moved simultaneously
  - The collision detection system is run
    - Every pair of colliding bodies is detected and stored
  - The collision resolver is run
    - Velocities are updated
    - Penetration is removed
  - All bodies are drawn at their new positions

Collision Detection

- Information needed:
  - Contact normal
  - Contact point
  - Penetration distance
- How to get this information?
  - Separating Axis Theorem

Separating Axis Theorem



Conclusions

- What works well
  - Collision detection
  - Collision response
- What doesn't
  - Not easily reusable
  - Unstable with large numbers of contacts
- Changes from initial plan
  - Resting contacts more complex than expected
  - Compound bodies not implemented

Possible Extensions

- Compound bodies
- Joints/Springs/Rods
- Simultaneous calculation of collision points
- Smarter algorithms
- 3D

Demo & Questions