

Termanator

Final Project Presentation

By

Rodrigo Ipince, Donald Eng, Kevin Luu

November 15, 2007

6.111 – Introductory Digital Systems Laboratory
Massachusetts Institute of Technology

Introduction

- Game: “Pseudo 3D” shooting game
- Objective: obtain highest timed score
- Controllers:
 - Shaker: generates the power for the gun
 - Pointer: directs the projectile of the shot



<http://ps2.ign.com/articles/721/721470p1.html>

Equipments Used

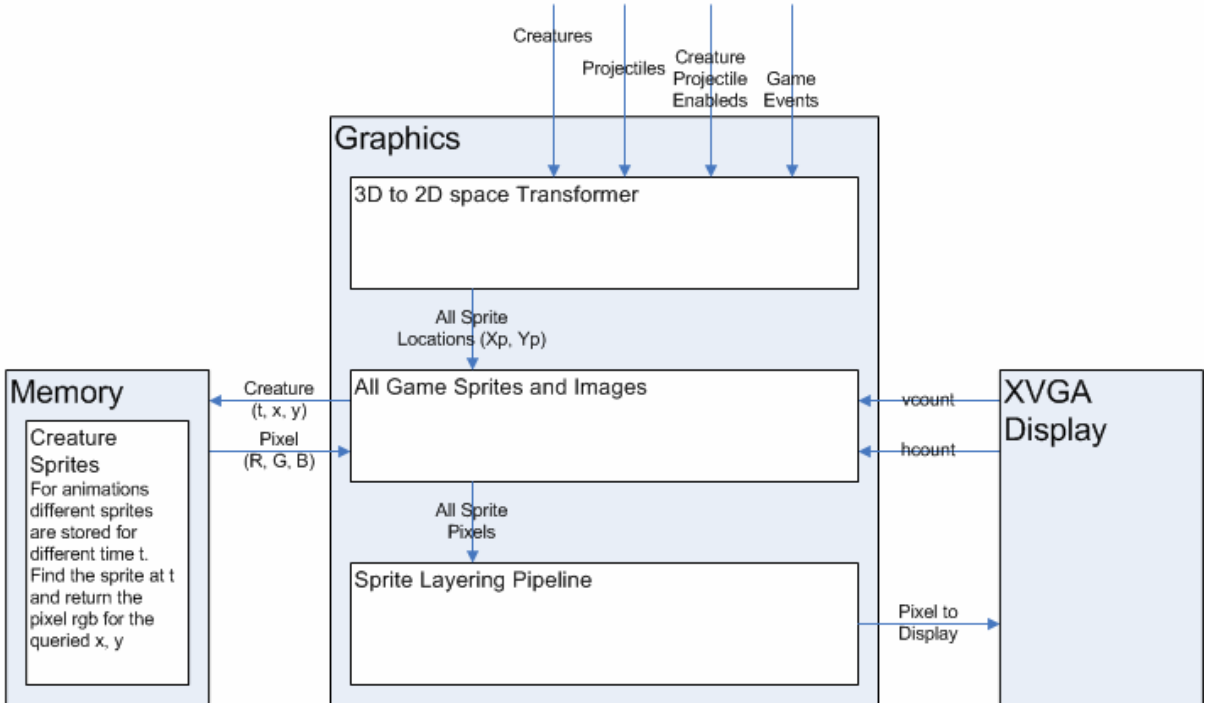
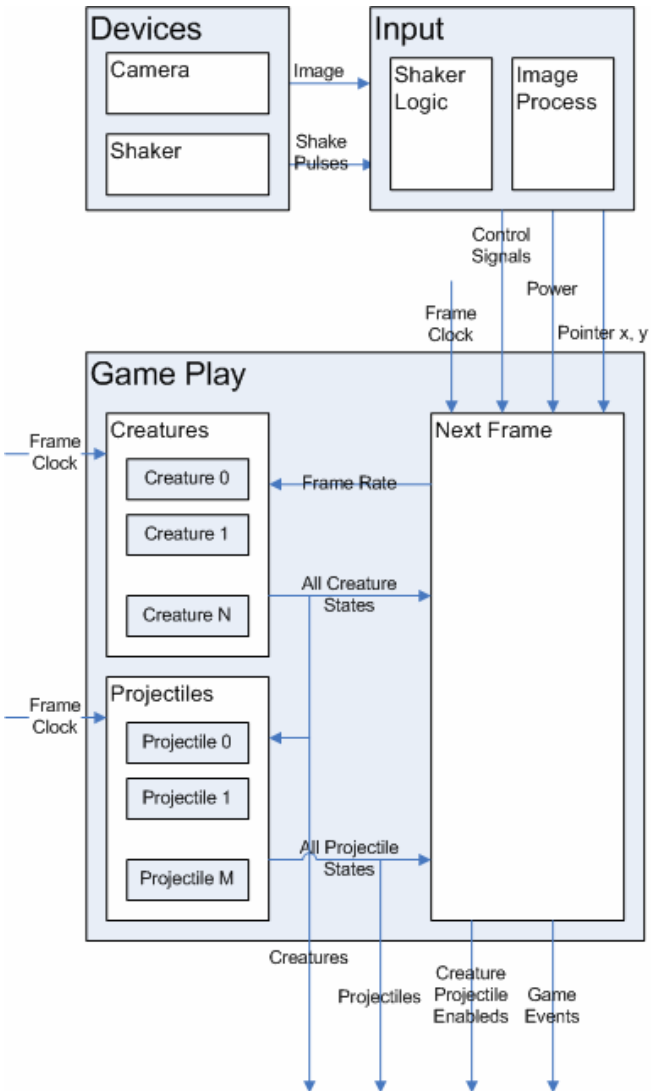
- X VGA Display:
 - 1024 X 768, clocked at 65Mhz
- NTSC Video Camera
 - provided by 6.111 staff
- Shaker
- Pointer



http://web.mit.edu/6.111/www/f2006/projects/jburnham_Project_Final_Report.pdf

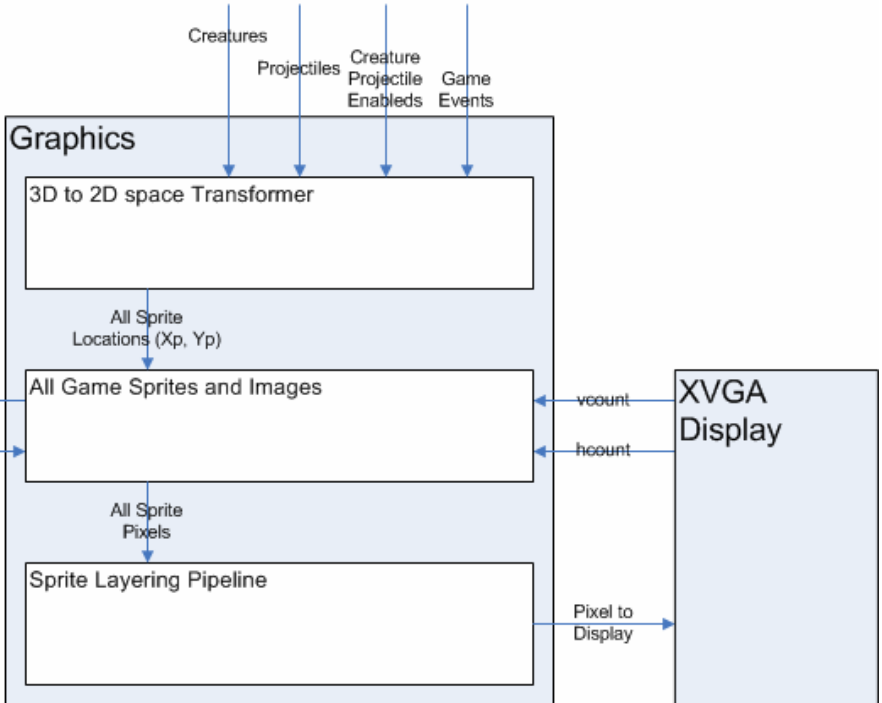
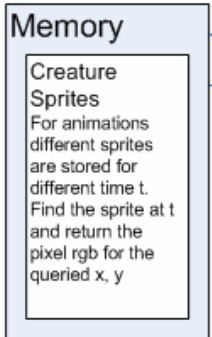
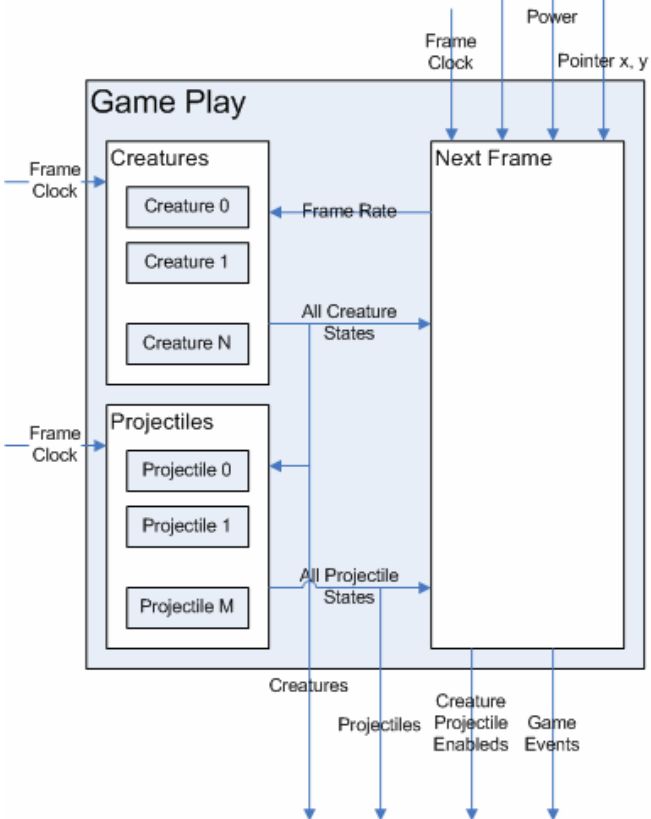
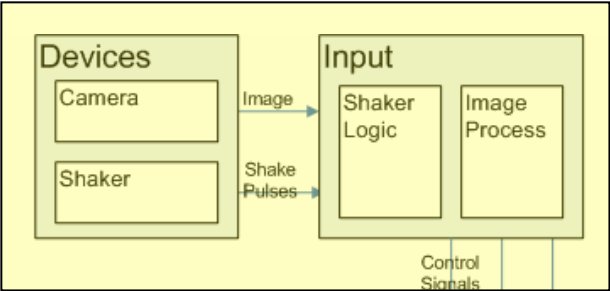


High-Level Block Diagram



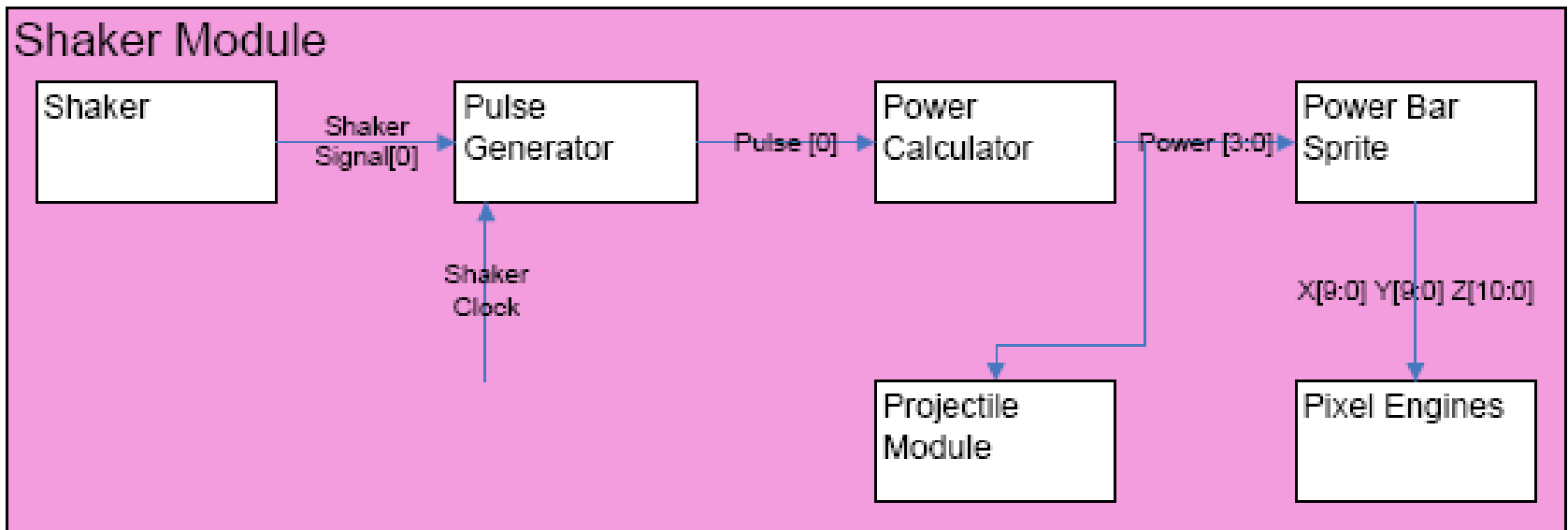
High-Level Block Diagram

Devices/Input

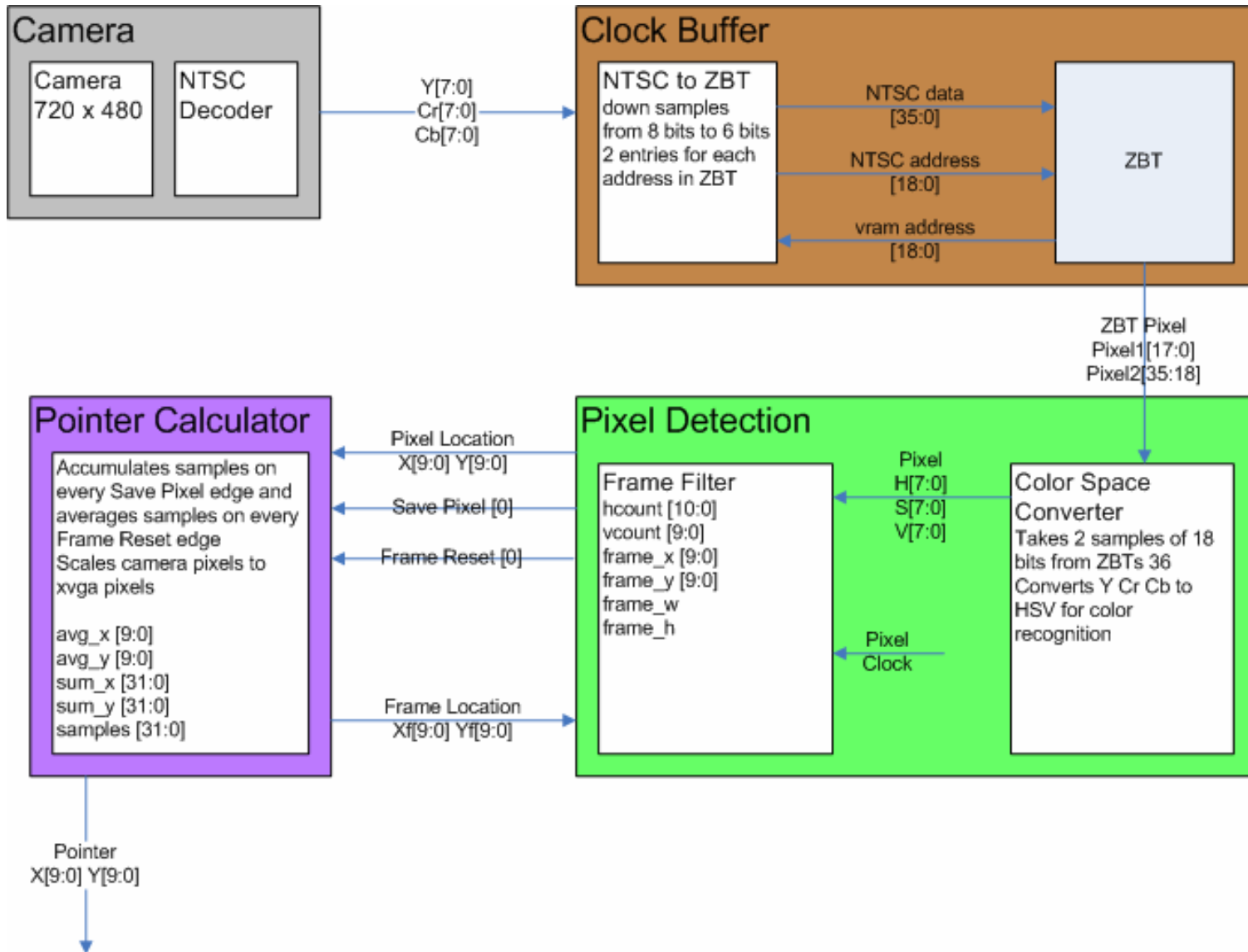


Shaker

- Time difference between pulses = Value for power level sent to Game Engine Module
- Sprite bar displayed on XVGA to indicate the current power level
- Moving-Average Filter (incorporated into Power Calculator)

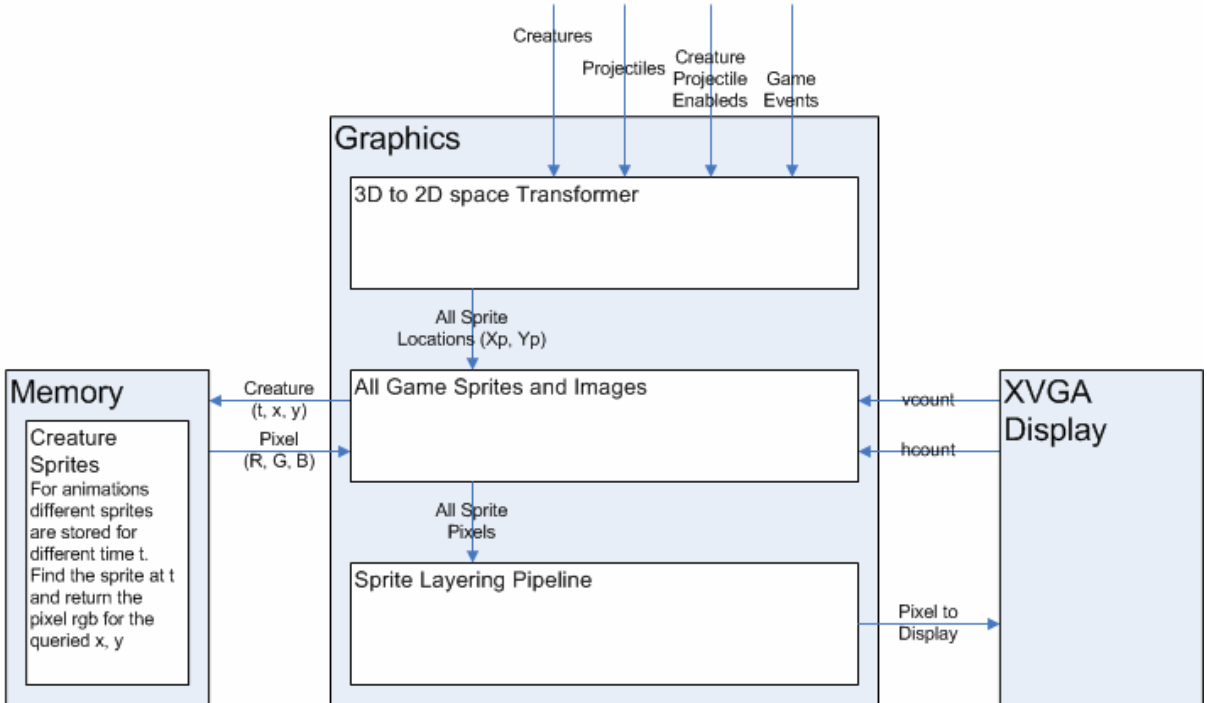
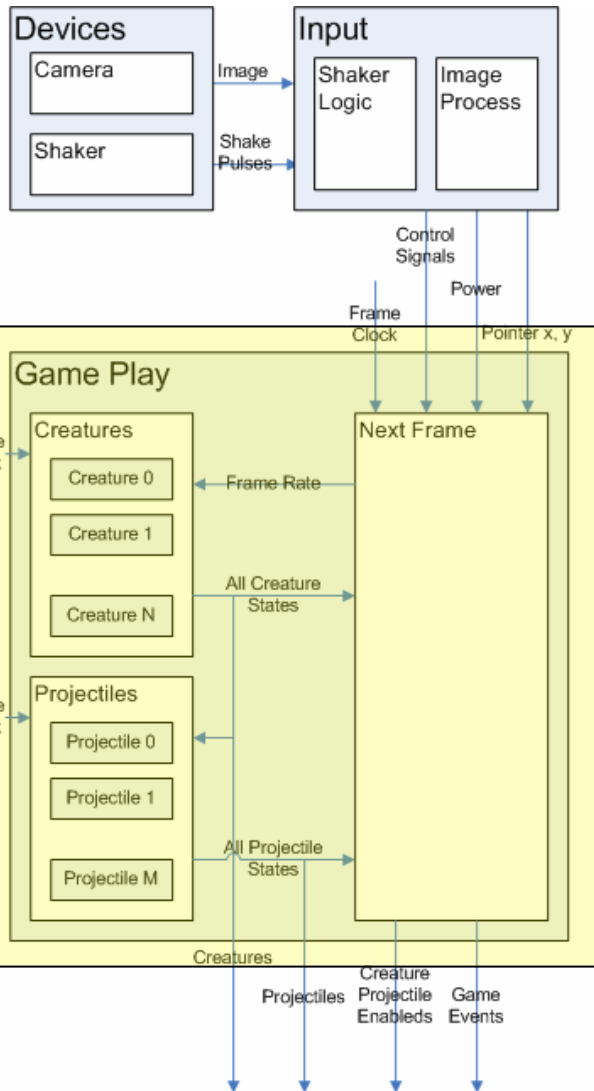


Video Processing



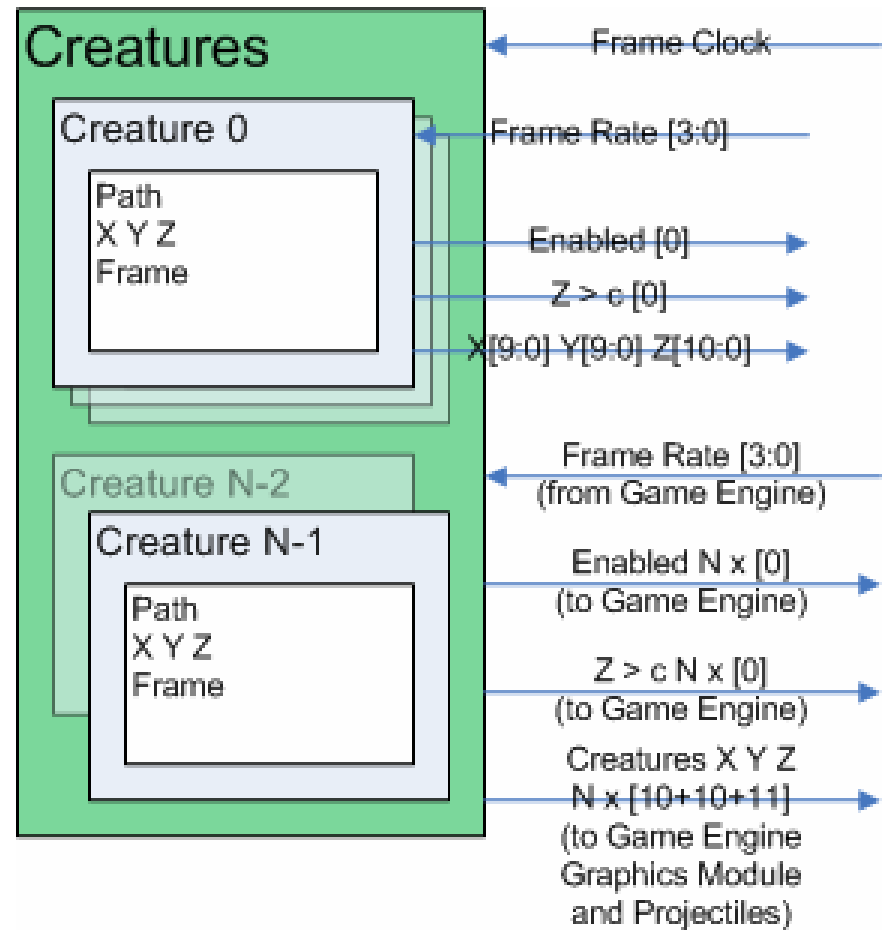
High-Level Block Diagram

Game Play



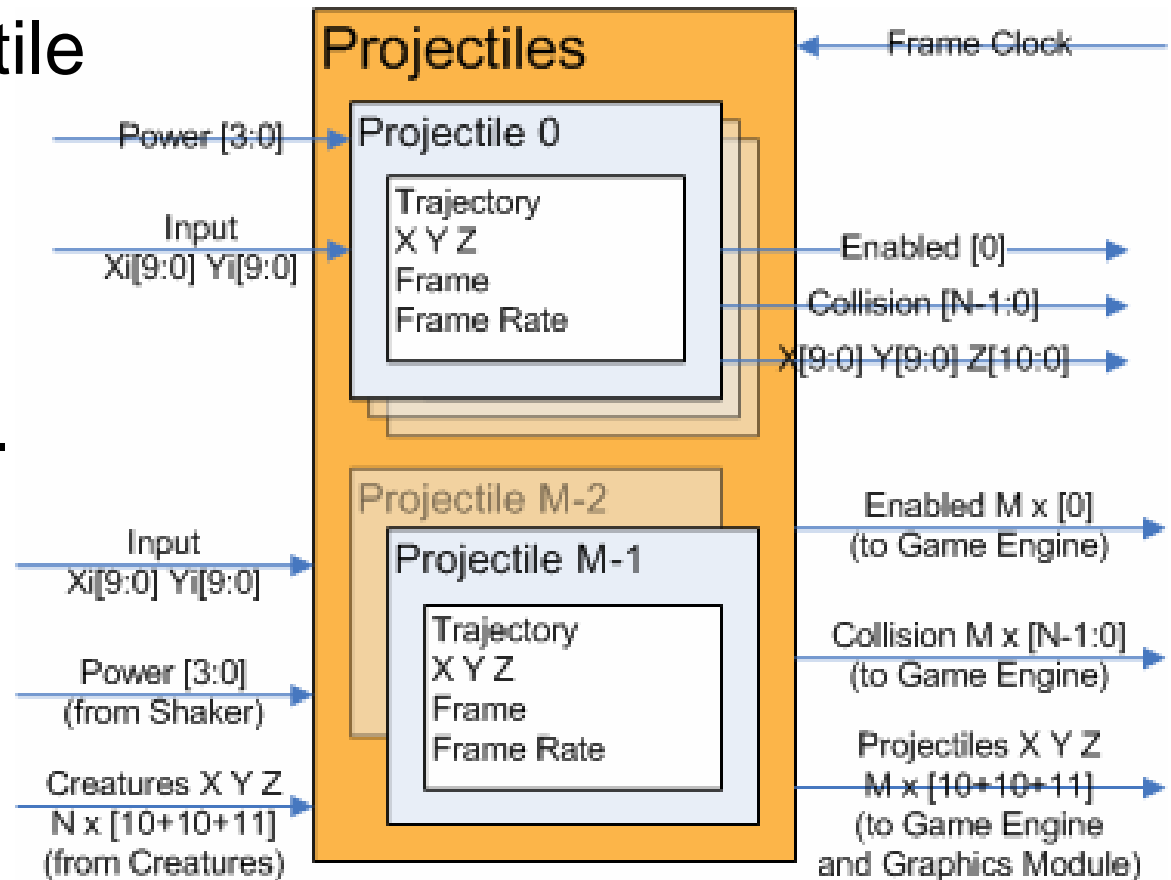
Creatures

- Maintains path, position (x,y,z), and frame for each creature.
- Creatures advance according to a frame rate.
- Creatures getting too close.
- Randomly generates new paths for creatures.
- Later: parameter to determine how many active creatures to have at any point in time.



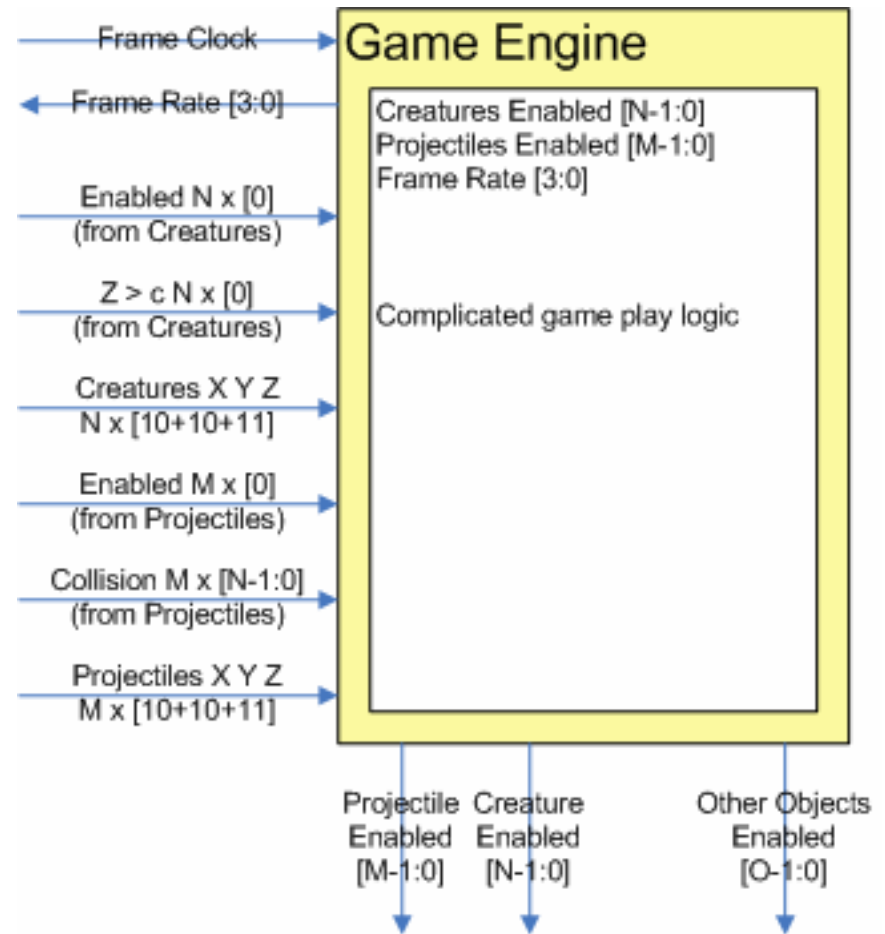
Projectiles

- Maintain projectile information.
- Trajectories determined by pointer location.
- Frame rate determined by power.
- Collision check.



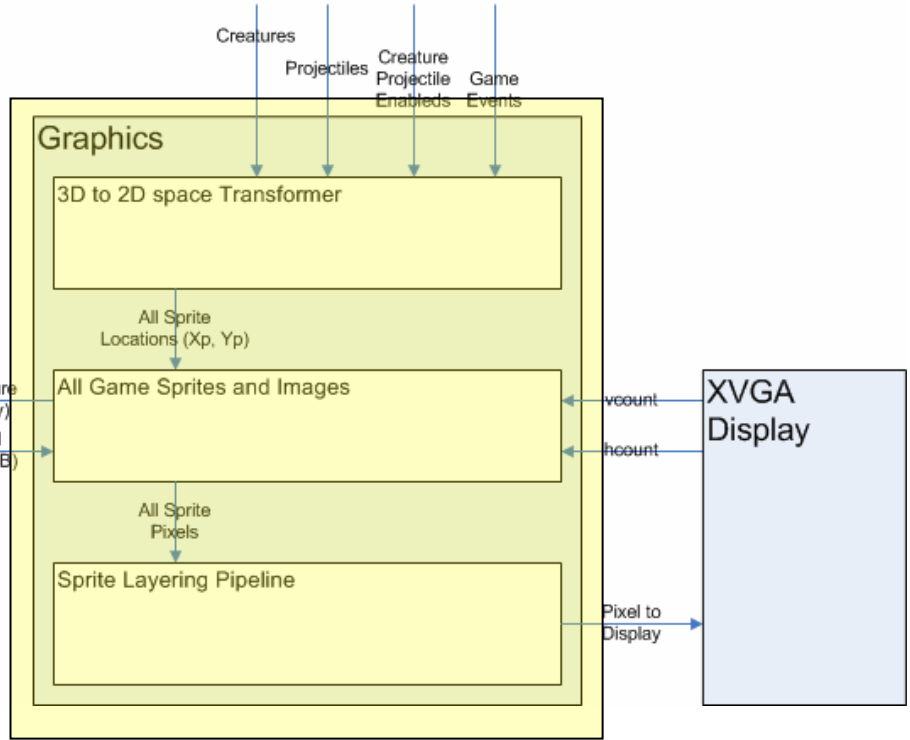
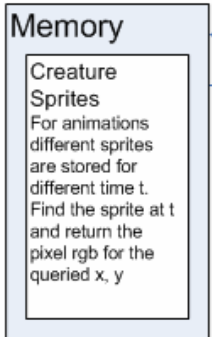
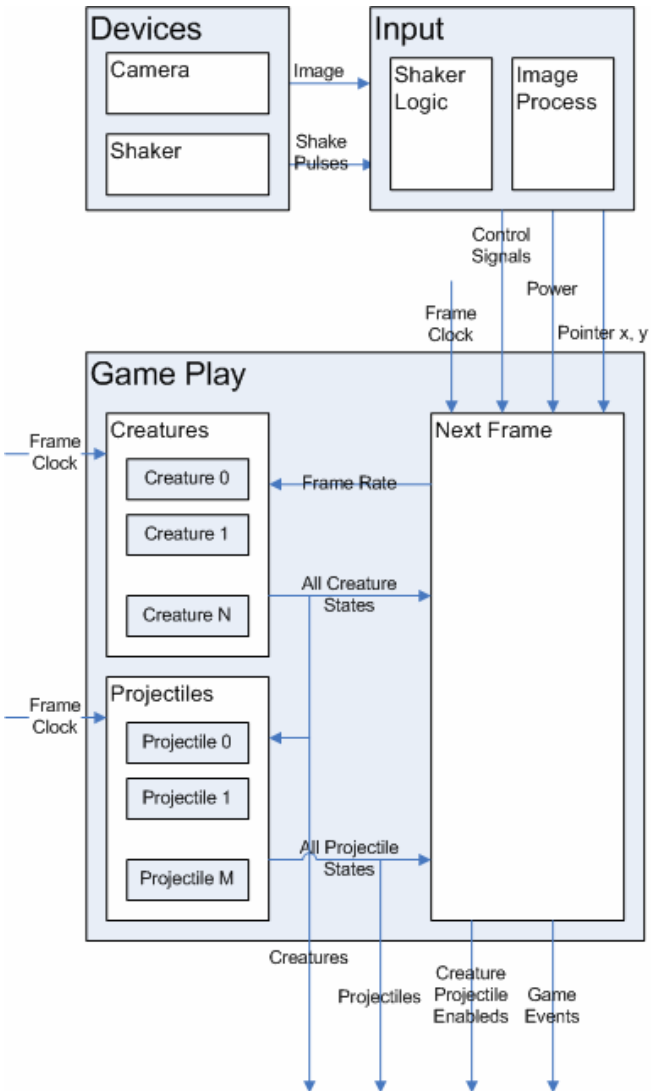
Game Play

- Game start/end
- Control enabled status for each object.
- Increase frame rate as time progresses.
- Check if collisions were amongst active objects.

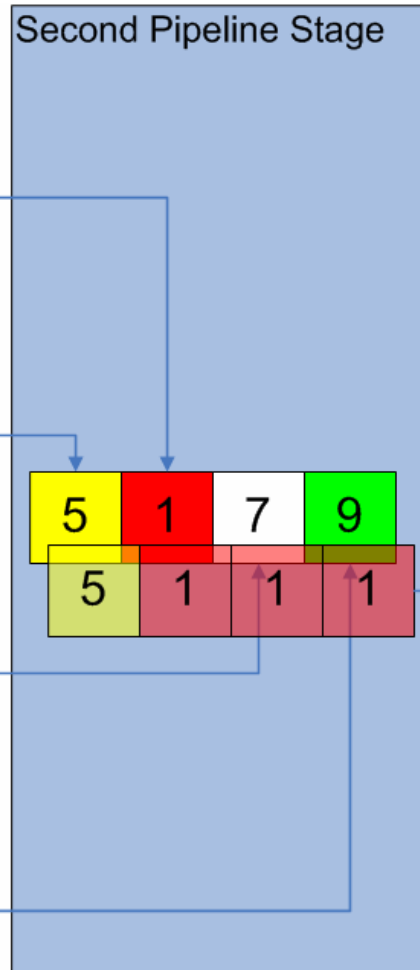
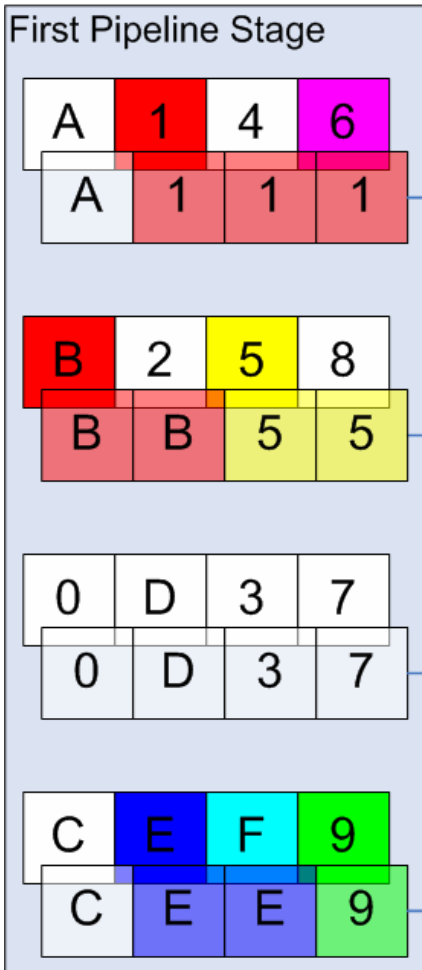
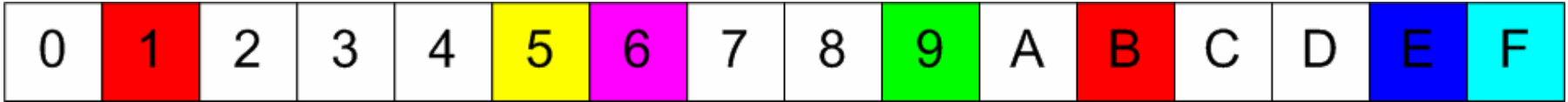


High-Level Block Diagram

Graphics

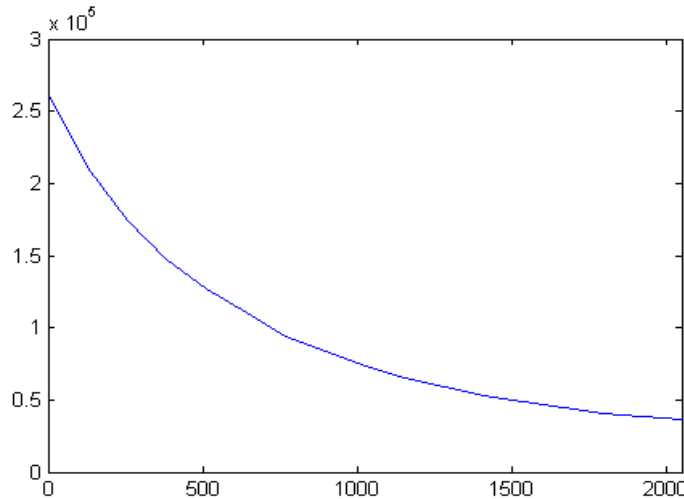
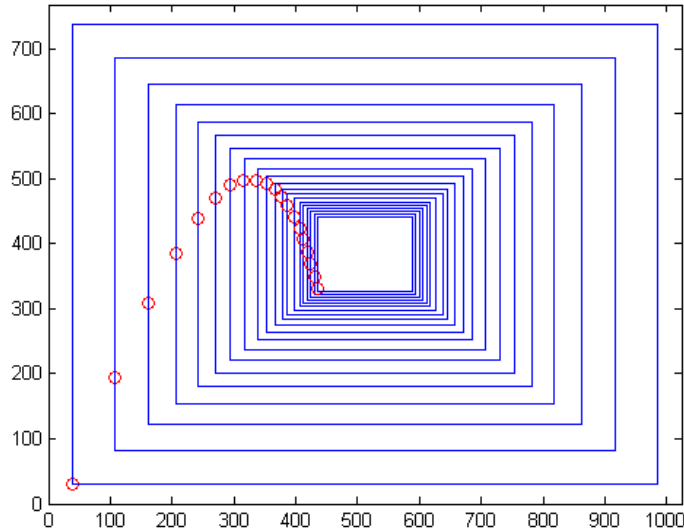


Graphics Pipeline



- Sprites: Pixels and Depths
- Display closest opaque pixel
- Pipelining
 - Divide and conquer
 - Screen shift by pipeline stages

Graphics Perspective



- 3D to 2D transform
 - Game play in 3D space
 - Linear Interpolation of perspective data
- Accessing Sprites from RAM/ROM
 - Parse *.bmp into *.coe
 - Animation sprites based on frame



<http://www.thinkgeek.com/tshirts/japanese/9980/>

Timeline

	11/18/2007	11/25/2007	11/30/2007	12/2/2007	12/5/2007	12/7/2007
Inputs	Complete Shaker Module	Working prototype of Video Processing	All modules tested and working properly			
Game Engine	Working prototype of Creature Module	Working prototype of Projectile Module				
Graphics	Complete Portable Graphics Module for Debugging	Sprites read from *.bmp to ram to XVGA	Miscellaneous Sprites (i.e. Splash Screen)			
Integration			Combine Game Engine with Graphics			
Debugging	Testing Game Engine on Graphics			Getting all Modules to work together		Working System

Questions?