Problem Solving using Search
Beyond Toy Problems
Adaptive Design

- Adaptive gameplay parameters
- Adaptive characters
- **Adaptive Visual Design**
Why Lighting?

- Dramatic Tension
- Visual Focus
- Style
- Feel of the space
- Time of Day
- Period
- visibility
Lighting in Traditional Media
Manually setting light layout (light maps)

Advantages:
- realistic
- Controllable

Disadvantages:
- Don’t adapt to variations in the environment

Images from Lightmaps (static shadowmaps) article written by Kurt Miller from: http://www.flipcode.com/articles/article_lightmaps.shtml

Image from Max Payne
Game Lighting: Static+Dynamic Lighting

- Manually scripting Lighting effects
- Dynamic Character Lighting
- Real-time Shadows
- Dynamic user control of some lights
Game Lighting: Static+Dynamic Lighting
Game Lighting: Static+Dynamic Lighting

Advantages:
- More realistic
- Adaptable to change

Disadvantages:
- Effects are scripted and rely on very careful design
- Restricted
- Based on realism
- Compositional Balancing is done at design time
Problems: No modulation

Screenshot from Mission 21
*Devil May Cry*

Saturation Graph for Screenshot
Problems: No adaptation to game Play
Problem: unpredictability

Develop a lighting plot or setup based on:

- Narrative configuration:
  - Story
  - Conflict/dramatic tension
- Timing and dramatic progression
- Physical Configuration:
  - Camera orientation and position
  - Characters positions and orientations

Requiring dynamic adaptation
A lighting system that:

- *Intelligently* adjusts lighting in real-time to accommodate context and effect.
- Based on *cinematic* & *theatric theory*.
- Allow *artist* to *control* lighting at a *high-level*.
The three subsystems: use optimization to find best solution given context, desired effects, state, and artists’ constraints.
The three subsystems: use optimization to find best solution given context, desired effects, state, and artists’ constraints
Automatic light allocation

Find best allocation given artistic constraints:

- Modeling
- Depth
- Visibility
- Visual Continuity
- Visual Focus
- Low vs. high key
Algorithm for dynamic allocation

1. Calculate visible area
2. Divide the zone into overlapping areas
3. Allocate a number of lights to areas, given that

\[ p : L \rightarrow A \]

minimize:

\[ p_{opt} = \arg \max_p (\lambda_v V(p) + \lambda_d D(p) + \lambda_m M(p) + \lambda_{vc} VC(p)) \]

visibility depth modeling Visual continuity
Layout of lights
The three subsystems: use optimization to find best solution given context, desired effects, state, and artists’ constraints.
Lighting angle Selection

Select azimuth, elevation angles, given artistic constraints:

- Visual Continuity
- Motivation of direction
- Visibility
- Modeling
- Mood

ELE finds best light angles to meet goals
Automating cinematic lighting design
Choosing key light azimuth angle

Optimizes:

\[
\text{cost}(k, s, k^-, m) = \lambda_v (1 - V(k, s)) + \lambda_- |k - k^-| + \lambda_m |k - m| + \lambda_l \min_i |k - l_i|
\]

- \(k\) is the key light angle, \(k^-\) is angle previous frame
- \(m\) is the mood angle desired, \(V\) is visibility
- \(\lambda_v\) is the cost of deviation from best visibility angle
- \(\lambda_-\) is the cost of visual continuity
- \(\lambda_m\) is the cost of deviation from realistic direction
- \(\lambda_l\) is the cost of deviation from ideal mood angle

Visibility & modeling, visual continuity, mood, motivation
Evaluation of key light angle visibility & modeling

We described these rules as:

\[ V(k, s) = \sin(k) \cos(s) \]

From: Millerson’s

*The technique of lighting for television and Film, 1991*
ELE: Angle Subsystem

Showing character’s facial expressions and gestures
ELE: Angle Subsystem

Showing character’s facial expressions and gestures, high tension
Mood Angle = side angle

Emphasizing mood and mystery
Mood Angle = side angle

Emphasizing mood

Artist's settings

Costs

Visibility  Mood  Contrast
The three subsystems: use optimization to find best solution given context, desired effects, state, and artists’ constraints.
Choosing Colors

- Color (Hue + saturation + intensity)
- Compose colors for different areas on the set
Choosing Colors

Adjust colors to accommodate desired artistic constraints:

- Depth
- Dramatic Intensity
- Dramatic focus
- low vs. high key setting
- Specific author-suggested
  - Hue, Saturation, Lightness, color Warmth for focus, non-focus, and background
  - Palette restrictions specifying style

+ maintain **visual continuity** and **style**
Choosing Colors

Optimize:

\[
cost(c^t, c^{t-1}) = \lambda_d \left( D(c^t) - d \right)^2 + \lambda_c \left( \text{contrast}_\phi(c^t) - \delta \right)^2 + p(c^t) + \sum_{i \in \{f, n, b\}} \lambda_{vc} E(c_i^t, c_i^{t-1}) + \sum_{i \in \{f, n, b\}} I(c_i^t),
\]
Choosing Colors

\[ I(c_i^t) = \lambda_{s_i} \left( S(c_i^t) - s_i \right)^2 + \lambda_{h_i} \left( H(c_i^t) - h_i \right)^2 + \lambda_{l_i} \left( L(c_i^t) - l_i \right)^2 + \lambda_{w_i} \left( W(c_i^t) - w_i \right)^2 \]

Saturation, Hue, Lightness, Warmth
Calculating Color Warmth

Based on warmth perception

\[ \text{warmth} = \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{bmatrix} 0.008 & 0.0006 & -0.0105 \end{bmatrix}^T \begin{pmatrix} R \\ G \\ B \end{pmatrix} - 0.422 \]
warm tones, low color contrast
for low tension, realistic colors
cool tones, low color contrast for night scenes, or emphasize character
ELE: Color Subsystem

warm, high intensity contrast for high tensions scenes
Demo (Dynamic Lighting)
Demo (Static Lighting)
Possible Projects

- Develop a poker game played with AI players developed with different algorithms and compare (involves chance, betting, multi-agent)
- Develop a Settlers of Catan AI (involves negotiation)
- Develop an AI that can play Starcraft based on a table approach (learning strategies from observation)

These do not just use Search but other algorithms we will talk about in class
Now for the Grand Challenge

- Play the game developed by Eric Zimmerman
- Think about replacing all the team members in this game with agents
- Develop algorithms in pseudocode for these agents will look like
- **Post your solution on Piazza. Outline:**
  - Design considerations: attributes for the agents, how environment is represented, etc.
  - Search Algorithms you will use and change (if you need to)
  - Describe any other challenges that you think search won’t handle in this case