Lecture 6:
Architecture and Planning
Pong / Asset Creation Resubmission

- Due Today, September 17\textsuperscript{th} at 11:59pm
Game design document

- Due Tomorrow, September 18th at 11:59pm
Throwaway Prototype Testing

- Pick *some piece* of your game and build it
- In class on **Tuesday, September 22nd**
Final Presentations

- December 12th
- 2-4:30pm
- Room TBA
Newgrounds Postmortems

- Tuesday, November 24th in class
- This is the Tuesday of Thanksgiving week
- This is a major presentation
Unnecessary Tactics
Project management

- Deliver software
  - on time
  - on budget
  - with required functionality
  - without exhaustion
  - without infighting
Activity Chart
Earliest Time
Latest Time
Earliest and Latest Time
Earliest and Latest Time
Critical Path

Level design

Integrate levels

Learnability testing
Critical Path

0/0 → Engine

20/20 → Integrate engine

25/25 → Usability testing

28/28 → Learnability testing

31/31
Uncertainty
Uncertainty
Plan for the *worst case*

- Prioritize big unknowns
- Major early unknown: will the game idea work?
Alpha Prototype

- Art assets
- Engine
- Event triggers
- Level design

Integrate art assets
Integrate UI
Integrate engine
Integrate event triggers
Integrate levels

Usability testing

good?
bad?
Alpha Prototype

• Implement the minimum to accept/reject the idea
• Answer key questions:
  • Is it fun?
  • Is it usable?
  • Can it scale in complexity?
• Build these components:
  • three functioning levels: easy, medium, hard
  • core game mechanics for these levels
  • minimal art/UX integration
  • a list of input keys
NOT needed for Alpha Prototype

- More than three levels
- Game mechanics not needed for these levels
- Tutorials
- Music
- Sound effects
- Full art integration
- Full user interface
- Logging
Alpha Prototype testing: 10/1

- I will invite testers from outside the class!
Alpha Prototype Examples
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Traditional Way to Break Up a Game

• Rules and Mechanics
• User Interface
• Game Engine
• Content
Game Engine

- Component that powers the
  - graphics and sound
  - physics
  - artificial intelligence
  - game mechanics
  - interactions

- Game environment is
  - simulated by the engine
  - populated by the content
History of Engines
Doom (1994)
Unreal (1998)
Box2D – in Flash!
Flixel
Flixel
Flixel
public class MyGame extends FlxGame
{
    super(width, height, MyState);
}
public class MyState extends FlxState
{
    public override function create():void
    {
        // create stuff
    }
    public override function update():void
    {
        // update stuff
    }
    public override function destroy():void
    {
        // destroy stuff
    }
}
public override function create():void {
    player = new FlxSprite(FlxG.width/2 - 5);
    player.makeGraphic(10,12,0xffaa1111);
    player.maxVelocity.x = 80;
    player.maxVelocity.y = 200;
    player.acceleration.y = 200;
    player.drag.x = player.maxVelocity.x*4;
    add(player);
}
public override function update():void
{
    player.acceleration.x = 0;

    if(FlxG.keys.LEFT)
        player.acceleration.x = -player.maxVelocity.x*4;

    if(FlxG.keys.RIGHT)
        player.acceleration.x = player.maxVelocity.x*4;
}
FlxG

// size of game
FlxG.width
FlxG.height

// useful methods!
FlxG.overlap(coins, player, getCoin);
FlxG.overlap(exit, player, win);
FlxG.collide(level, player);
Content

• Everything else
  • Levels
  • Art assets
  • Story messages
  • Sound effects
  • Music
  • Tutorial messages
Level Editor
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Division of Labor

“nine women can’t make a baby in one month”

- Fred Brooks
Model-View-Controller

User
- uses
- shows

Controller
- manipulates
- updates

View

Model
Architecture of Thermo
Gradual disorganization
Appoint a cleaner

source: pets4homes.co.uk
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Version control

- SVN
- Git
- Mercurial
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
Alpha Prototype

1. Plan components
2. Divide up labor
3. Merge
## Architecture activity

<table>
<thead>
<tr>
<th>AI Engine</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability</td>
<td>Interactions</td>
</tr>
<tr>
<td>Pathfinding</td>
<td>Game state</td>
</tr>
<tr>
<td>Minimax solver</td>
<td>Game state</td>
</tr>
</tbody>
</table>