• Mon 31 aug 15:00-17:00
• Tue 1 sep 13:00-17:00
• Mon 7 sep 15:00-17:00
• Thu 10 sep 10:00-12:00
• Mon 14 sep 15:00-17:00
• Thu 17 sep 10:00-12:00
• Tue 22 sep 10:00-12:00
• Fri 25 sep 8:00-16:00
• Mon 28 sep 15:00-17:00
• **Mon 5 oct 15:00-17:00**
• Mon 12 oct 15:00-17:00
• Fri 30 oct 9:00 – Sun 1 Nov 16:00
• Mon 2 nov 15:00-17:00
• Tue 3 nov 13:00-17:00
• Tue 10 nov 10:00-12:00
• Tue 17 nov 10:00-12:00
• Tue 24 nov 10:00-12:00
• Tue 1 dec 10:00-12:00
• Fri 4 dec 15:00-19:00

Lecture 1 – **Introduction**
Lecture 2-3: **Forming Groups and Brainstorming**
Lecture 4: **Proposals**
Lecture 5: **Discussion based on Proposals**
Lecture 6: **Hello World Demos**
Lecture 7: **Discussion based on the Hello World Demos**
Lecture 8: **Preparing ForskarFredag 2015 ForskarFredag**
Lecture 9: **Reflecting on ForskarFredag**
Lecture 10: **Agile Development 1 towards Comic Con - Gamex 2015 Comic Con Gamex**
Lecture 11: **Agile Development 2 towards Comic Con - Gamex 2015 Comic Con Gamex**
Lecture 12: **Reflecting on Comic Con Gamex**
Lecture 13-14: **Forming new groups and brainstorming project 2**
Lecture 15: **Proposals Project 2**
Lecture 16: **Hello World Demos for Project 2**
Lecture 17: **Agile Development 1 for Open House**
Lecture 18: **Agile Development 2 for Open House VIC AGI15 Open House**
Agenda

1. Hopscotch demos
2. Deliverable Oct 23
3. Results from Assignment 2
4. Logos and Posters
5. Next phase development (quickly fix interaction problems and focus on advancing the graphics)
   1. Special Effects
      1. Particle Systems
   2. Rendering quality
      1. Un-Aliasing
      2. Shadows
      3. Reflections
      4. Refractions
6. Focus on your own goals
   1. Make feature that is:
      1. Non critical
      2. Independent
      3. Focused on your learning goal
      4. Easily integrated
7. FB questions
8. Next Lecture
9. Schedule individual group meeting
Hopscotch

HOPSCOTCH
Make your own games

Download on the App Store
Is this AGI?

• Why?
  – Flexibility
  – Freedom of creation
  – Interaction can be gesture- and sound-based, but not experimental enough to have to deal with noise (like you do in your projects)

• Why not?
  – Fixed (limited to certain controls and graphics)
  – No cool shaders
  – Plugins?
  – Limited ability to mathematically model simulations of light interacting with complex objects (this is the core of modern advanced computer graphics).
Remember: Deliverable Oct 23

- Webpage with:
  - Description
    - Goal and motivation of the project
    - Explanation and Justification of the graphics and interaction technologies used and developed
    - Challenges
    - Obstacles
    - Related work
    - Lessons learned
  - Look at AGI14 web pages
  - Photos
  - "Making of" documentary (2 minutes)
  - Demo Reel (30 seconds)
  - Optional PR material (logo, trailer, flyers, posters, catalog)
  - User testimonials (what did people say)
Remember
For example... particle systems
Particle Systems

• **Original Paper**
  – Particle Systems A Technique for Modeling a Class of Fuzzy Objects
  – ACM SIGGRAPH 1983
  – William Reeves
  – Let’s read the paper
Paper’s Abstract

• Model fuzzy objects
  – Fire
  – Clouds
  – Water
• Particles = primitives of a volume
• Shape of volume is non-deterministic (stochastic)
• In time:
  – Generated
  – Move
  – Change
  – Die
Model (Algorithm)

1. Generate new particles
2. For all particles
   1. Assign individual attributes
   2. If too old, kill
   3. Else, transform = $f(\text{dynamics})$
3. Render image of living particles
Exercise

• Simple Particle System
  – Link
• Understand Code
• Modify to
  1. Simplify to one particle with simple trajectory
  2. Add physics like wind and gravity
Example of particle systems 2015

LazyFluids: Appearance Transfer for Fluid Animations

Ondřej Janůška¹  Jakub Fišer¹  Paul Asente²  Jingwan Lu²  Eli Shechtman²  Daniel Šykora¹
¹CTU in Prague, FEE  ²Adobe Research

Figure 1: LazyFluids in action—an artist first designs a target fluid animation that consists of a sequence of motion fields (a) and alpha masks (b), and then selects a video exemplar of a fluid element with desired appearance (c) and alpha mask (d). Finally, our algorithm transfers appearance of the exemplar to the target animation while respecting its motion properties and boundary effects (e). The resulting animation can then be used as a part of a more complex composition (f). All alpha masks in the paper are visualised in a way that fully opaque pixels are black and fully transparent are white. Dragon painting © Jakub Javorský.
Task

1. Next phase development (quickly fix interaction problems and focus on advancing the graphics)
   1. Special Effects
      1. Particle Systems
      2. Mathematically model simulations of light interacting with complex objects ...

2. Rendering quality
   1. Un-Aliasing
   2. Shadows
   3. Reflections
   4. Refractions

2. Focus on your own goals
   1. Make feature that is:
      1. Non-critical
      2. Independent
      3. Focused on your learning goal
      4. Easily integrated
FB questions

- How do I connect 3 screens to Unity?
- Syncing music and SFX to game tempo and mode?
- Bump Mapping
- How big space can we get at Gamex (maximum)?
- Advanced enviroment modeling in Unity
- Can we get money/material for carpentry? (e.g. wood/plywood...)
- Occlusion culling with two cameras and moving objects
- Gamex, how many days will we stand there? / Time Schedule?
- Transparency in several layers to make occluded objects visible?

+ Add an option...
What to deliver?

• Freedom to fail
• Show work
  – Research
  – Paper reading and understanding
  – Development
  – Integration
• Write a short individual report (?)
  – At most
    • 1000 words
    • 3 references
    • 3 figures
    • 1 video-figure
• If working, working code
Ahead of you

- Phase 1 – fix interaction issues and work on core graphics together
- Phase 2 – research and propose a feature that works independently from the rest of the project and implement a simple version of it.
Example work plans from AGI14

• The following are plans made between ForskarFredag and Comic Con...
### 2Pacs

<table>
<thead>
<tr>
<th>General Categories</th>
<th>Specific Categories</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Buttons</td>
<td>Use same button?</td>
</tr>
<tr>
<td></td>
<td>Motion</td>
<td>Great! No change</td>
</tr>
<tr>
<td>Haptics</td>
<td>Vibrations</td>
<td>Avoid! Change</td>
</tr>
<tr>
<td>Output</td>
<td>Level</td>
<td>Looks professional?</td>
</tr>
<tr>
<td></td>
<td>GUI</td>
<td>What is my score?</td>
</tr>
<tr>
<td></td>
<td>L10</td>
<td>I can't see the hole.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is too difficult.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Difficult to catch!</td>
</tr>
</tbody>
</table>

#### Features

- Collision Vibrations
- Ghost Proximity Vibrations
- Improved Skybox
- Transparency
- Paper Effect Animations
- At End Show Score
- Particle Effects
- Player Stimuli
- Collision Sounds
- Pellet诗* Effect
- EMF effect

#### Related Work

Gameplay

- Gameplay encapsulates tree pipeline
- search-for-player loop
- Cinematic tour of the forest while waiting

Tree Pipeline

Input
- Input separated from Interaction
- Abstract Input Interface
  - InputHelper
    - KeyboardHelper
    - KinectHelper
  - Interface:
    - HandShoulderAngle()
    - HandShoulderDist()
    - HandShoulderDist3D()
    - ElbowShoulderAngle()
    - ElbowShoulderDist()
    - ElbowHandAngle()
    - ElbowHandDist()
    - HandTorsoDist()
    - ElbowTorsoDist()
    - TorsoTiltAngle()
    - ...
  - Logging, Debugging:
    - log Actions
    - replay sessions
    - use debugging view
    - behavior statistics

Tree Generation
- Interaction is handled here
- Abstract representation of the tree
  - Tree Data
  - Branch Data:
    - id
    - parent
    - children[]
    - position, rotation, scale
    - age, depth
    - leaves, flowers, lateral buds
    - not rendering specific:
      - no vertices, faces, normals
      - no colours
      - no shaders

Simulation
- Apply physics simulations
  - Store in separate SimulationData structure.
  - Wind, Gravity ...
  - Also other systems:
    - Leaves in air
    - Particle Systems
    - Day-night cycle

Tree Construction
- Creates full tree model
  - Ready for rendering
  - Data to GPU

Rendering
- Tree ready to be rendered
- Deferred Rendering Pipeline
  - First pass: to gBuffer (positions, normals, materials)
  - Second pass: fragment shader using data in gBuffer
- NPR - Non Photorealistic Rendering
- Lots of things to add:
  - DoF
  - Motion blur
  - Anti-aliasing
  - Volumetric light scattering
  - Bloom effects
  - Godrays
  - ...
Pod Racer

- Focus on the graphics
  - Fire
  - Electricity
  - Dust
  - Pseudo-realism
  - Advanced effects
  - Avatar with arms controlling the pod
  - Visualization of engine thrust
  - Map?
  - Virtual arrows?

- Sound
  - Location?
  - Special effects
  - Sound track
    - Expert: Roberto Bresin
      - You may justify spending time on good sound by calling it “advanced interaction” but I will not force you to do it.

- Change track
- In-game tutorial
Space Survival

- Controls
  - Hard
  - Confusing
  - Vertical thrusters
- In-game tutorial
- Self-localization
  - Map?
  - Virtual Arrows?
  - On avatar’s body?!
- Space dust – speed and direction
- Look at 3D movie creation – layering
- Sound in helmet
  - Sonification
  - Early warning
  - Communication with ground control, mission command, or space craft
Next week

• Planning Comic Con GameX (45 minutes)
  – Logistics
  – Schedules
  – Space
  – Furniture
  – Posters
• Presenting updates on projects (45 minutes)
  – Group Goals
  – Individual Goals
  – Schedule
• Set up individual meeting with me
Thank you!

marior@kth.se