CSCI 1106
Lecture 07
Game Project and
Project Management
Announcements

Summer School at the Department of Computer Science at Bonn-Rhein-Sieg University from May 15 - June 30, 2015.
Two courses offered in parallel:

Effective Software Testing, 6 ECTS

Game Development, 6 ECTS

No course fee
https://www.h-brs.de/de/inf/summer-school-brsu
“A note taker is required to assist a student in this class. There is an honorarium of $75/course/term, with some conditions. If you are interested, please go to the Advising and Access Services Centre, Killam G28 for more information or send an email to notetaking@dal.ca.”
Announcements

• Today’s Topics
  – The Game Design Project
  – High-level game design
    • The Unifying Theme
    • The Game Story
    • Game Mechanics
Your Mission: Write a Video Game

• Come up with an idea for a game
• Design the game
• Implement the game in Scratch
  – If it is not a first-person action game, be sure to clear it with the instructor first!
• Test and polish your game
• Create a user manual for the game
• Create a technical manual for the game
Design Considerations

• What is the theme and objective of the game?
• How will the player move?
• How will the player win and lose?
• How will the player know how well they are doing?
• How will additional levels differ?
• How will you communicate the purpose, rules, and controls of the game?
## Project Evaluation

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>No game.</td>
</tr>
<tr>
<td>D</td>
<td>Sprites are moving on the stage, some interaction between sprites but game is not playable.</td>
</tr>
<tr>
<td>C-</td>
<td>Player movement is successfully added to the game.</td>
</tr>
<tr>
<td>C</td>
<td>The game tracks collisions between the player and game objects and responds to them.</td>
</tr>
<tr>
<td>C+</td>
<td>The game has a clear objective (including the use of positive and/or negative game objects).</td>
</tr>
<tr>
<td>B-</td>
<td>The player is able to easily track their progress through the game (in the form of points or some other measure / approach).</td>
</tr>
<tr>
<td>B</td>
<td>Winning and losing the game are both possible and the game rules, purpose and how to play are clearly communicated to the player.</td>
</tr>
<tr>
<td>B+</td>
<td>The game includes multiple levels that increase in difficulty.</td>
</tr>
<tr>
<td>A-</td>
<td>The game has some polish (looks OK) and includes audio effects.</td>
</tr>
<tr>
<td>A</td>
<td>The game is polished (looks good) and has some interesting special effects.</td>
</tr>
<tr>
<td>A+</td>
<td>The game is highly polished (looks really good) and is compelling.</td>
</tr>
</tbody>
</table>

To achieve a certain grade you must also have all of the features of the lower grades.
Deliverables

• The game: .zip file
• The user manual (3 pages)
• The technical manual (7 pages)
• Presentation of your game during the presentation period

• The .zip file must be submitted before your presentation period (details to follow)
• All deliverables are due shortly after presentation day (date to follow)
The User and Technical Manuals

User Manual

• Contents:
  – Title page with screenshot
  – Game overview
  – Rules
  – How to play
• 3 pages, 11pt font
• Worth 20% of the written component

Technical Manual

• Contents
  – Title and Authors
  – Introduction
  – Description of Concept
  – Description of Sprites
  – Description of Stage
  – High-level Description
  – Description of Important Scripts
  – Description of Artwork and Sound
  – Future Work
• 7 pages, 11pt font
• 80% of the written component

• Templates are available on course website
• Rubric is in the project specification
Project Management

Good read: MIT project management tutorial
Identifying Tasks

Goal

Task A
- Time $T_A$
- Resources: Q, R, S

Task B
- Time $T_B$
- Resources: Q, S, V

Task C
- Time $T_C$
- Resources: Q, V

Task B₁
- Time $T_{B₁}$
- Resources: Q, S

Task B₂
- Time $T_{B₂}$
- Resources: Q, V

$T_B = T_{B₁} + T_{B₂}$
Scheduling Tasks & Allocating Resources

• Problem:
  – There are many tasks
  – There are many resources
  – Each task may have multiple dependencies

• Need to
  – Organize all of them in one place
  – Sort dependencies
  – Check for resource contention (i.e. capabilities)

• Idea: Use a Gantt chart
# Gantt Charts

<table>
<thead>
<tr>
<th>Resource</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource 1</td>
<td>Task 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 2</td>
<td></td>
<td></td>
<td>Task 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 3</td>
<td></td>
<td></td>
<td>Task 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Task 3</td>
<td></td>
</tr>
<tr>
<td>Resource 5</td>
<td></td>
<td></td>
<td></td>
<td>Task 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource 8</td>
<td>Task 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Gantt Chart Rules

• Time is represented horizontally (left to right)
• Resources are denoted vertically
• A task requires both time and resources
  – Represented by one or more rectangles
• If two tasks require the same resource, they cannot overlap
• If task A depends on task B, task A must follow task B
• The minimum amount of time needed to fit in all the tasks is the minimum amount needed for the project
Purpose of Gantt Chart

- Represent all tasks
- Represent resource use
- Represent dependencies
- Represent time of tasks

Question: How do we know where to place what on the chart?
Three Main Issues

- Dependency chains
- Resource contention
- Risk management
Dependency Chains

• Task A depends on B depends on C depends on D ...

• Time of longest chain is the minimum time of the project

• Place longest chain first

• Then the next longest ...
Resource Contention

• Tasks cannot use a resource at the same time
• A bottleneck occurs when many tasks need the same resource
• Solution:
  – Stagger tasks to avoid resource contention
  – Add more resources to contention
Risk Management

• Things will take longer than you think!
  – What happens to our schedule if we cannot find a solution for software bugs?

• How do we accommodate this fact of life?

• Solutions:
  – Schedule tasks as early as possible to provide time to deal with unforeseen events
  – Schedule extra time for each task
    • 10% to 15% extra time per task is not uncommon