The ESCAPE* Game

Group members: Vijay Prema, Mihailo Palevich, Greg Gilbert

Supervisor: Dr. Burkhard Wuensche

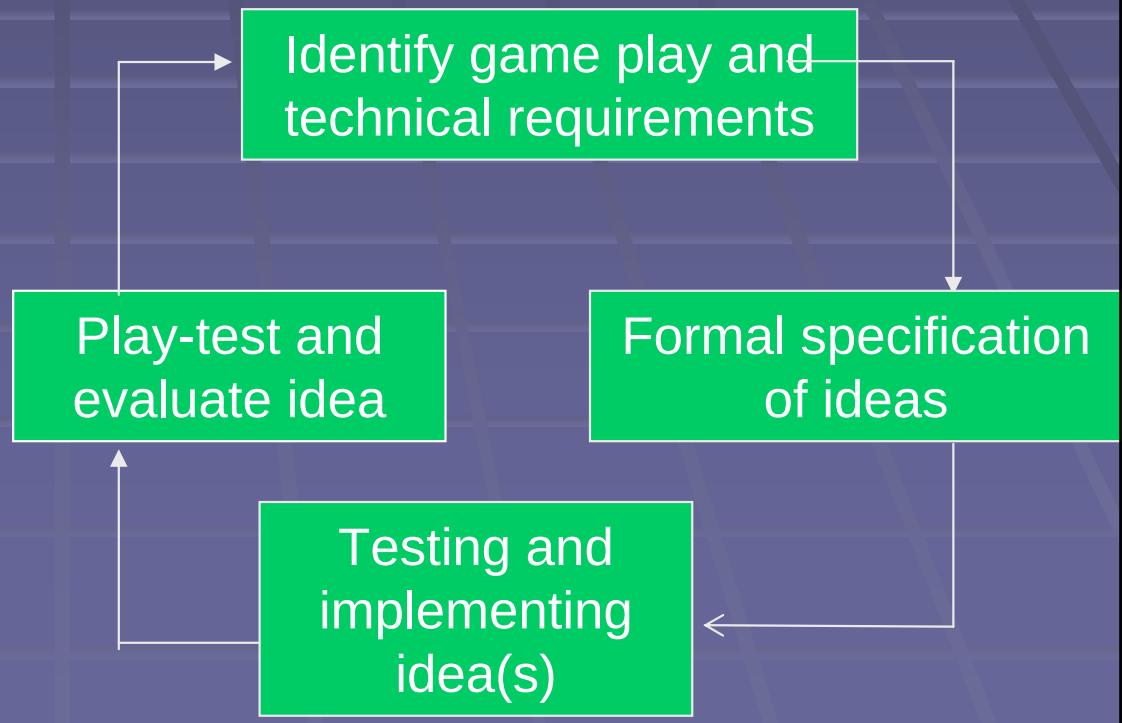
* Working title

Overview

- First person “Escape” game?
  - You must escape from a building/location.
  - Enemies will be trying to capture you.
  - NO GUNS, must use objects in the environment (“physics puzzles”) to overcome obstacles and/or slow down enemies.
  - Enemies will also use the environment and teamwork.
  - Aim for realism and immersion, while being playable and fun.

Development Plan

- Development process: Iterative
  - Iterative/cyclical design pattern, balanced with deadlines and milestones.
  - Ideal for systems that require frequent modifications and/or changes in requirements.
  - Our concept is relatively original and experimental.



Possible Game Theme

- “Zombie Escape”
- Player starts in a region, can move to other regions, but may need to use objects to gain access to them. (e.g. pile up boxes).
- Zombies (intelligent, slow moving and tough) pursue the player.
- Player must make it to the exit.

Gameplay Theory

Digital Games Research Association
http://www.digra.org/

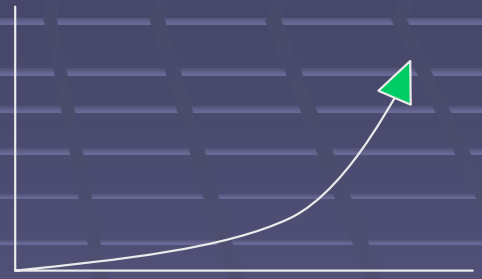
Game Studies
http://www.gamestudies.org/

What Makes Things Fun

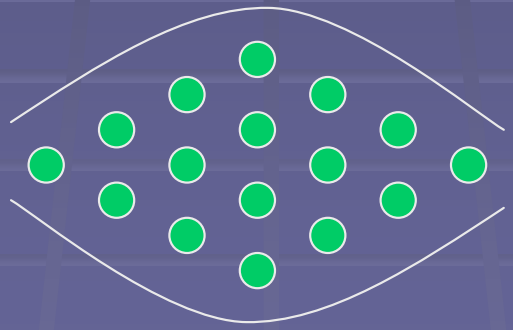
- Physical Conditioning
- Mental Patterning
- Social Interaction

Structure and Flow

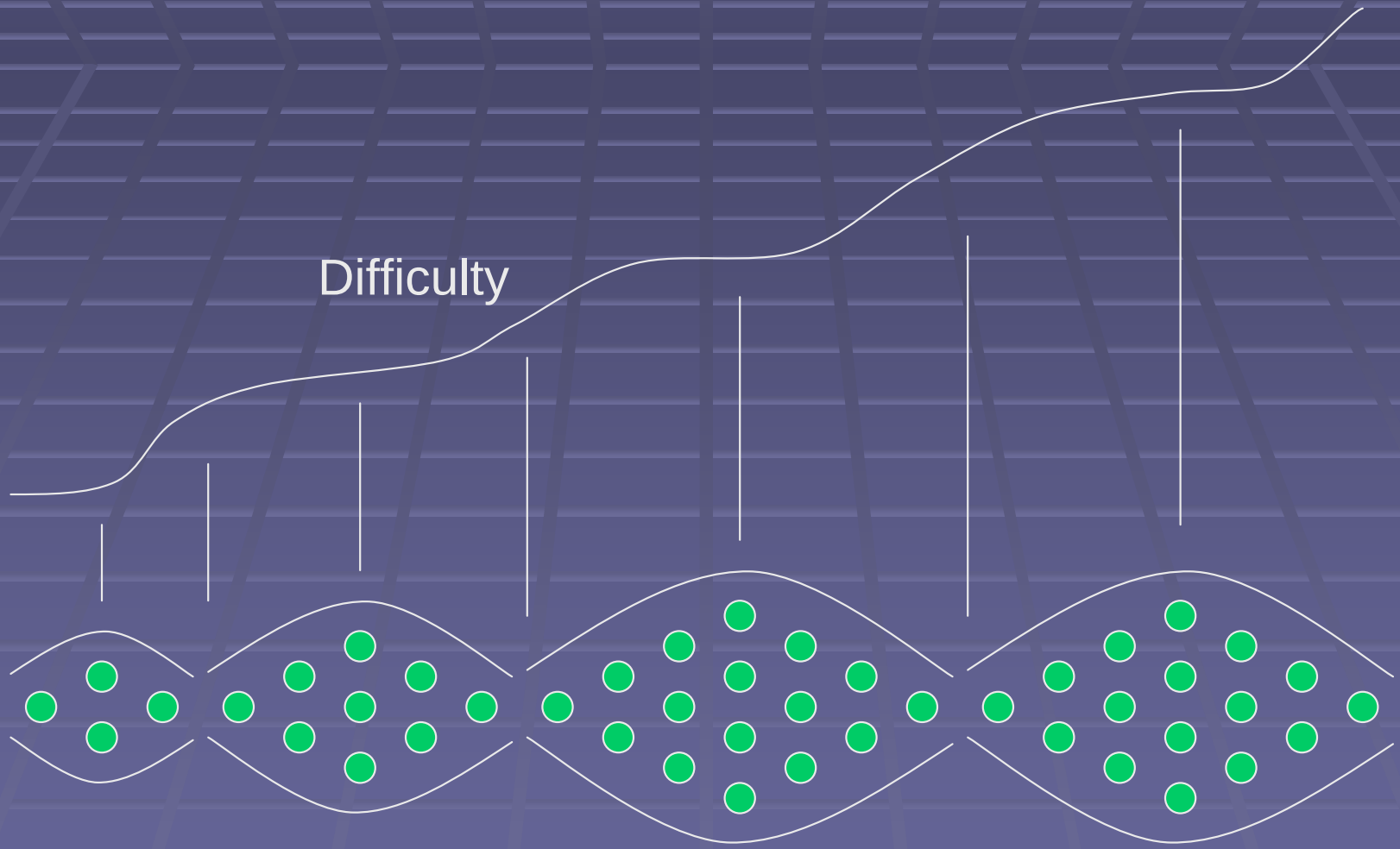
- Difficulty Progression



- Goal Awareness
- Convexity

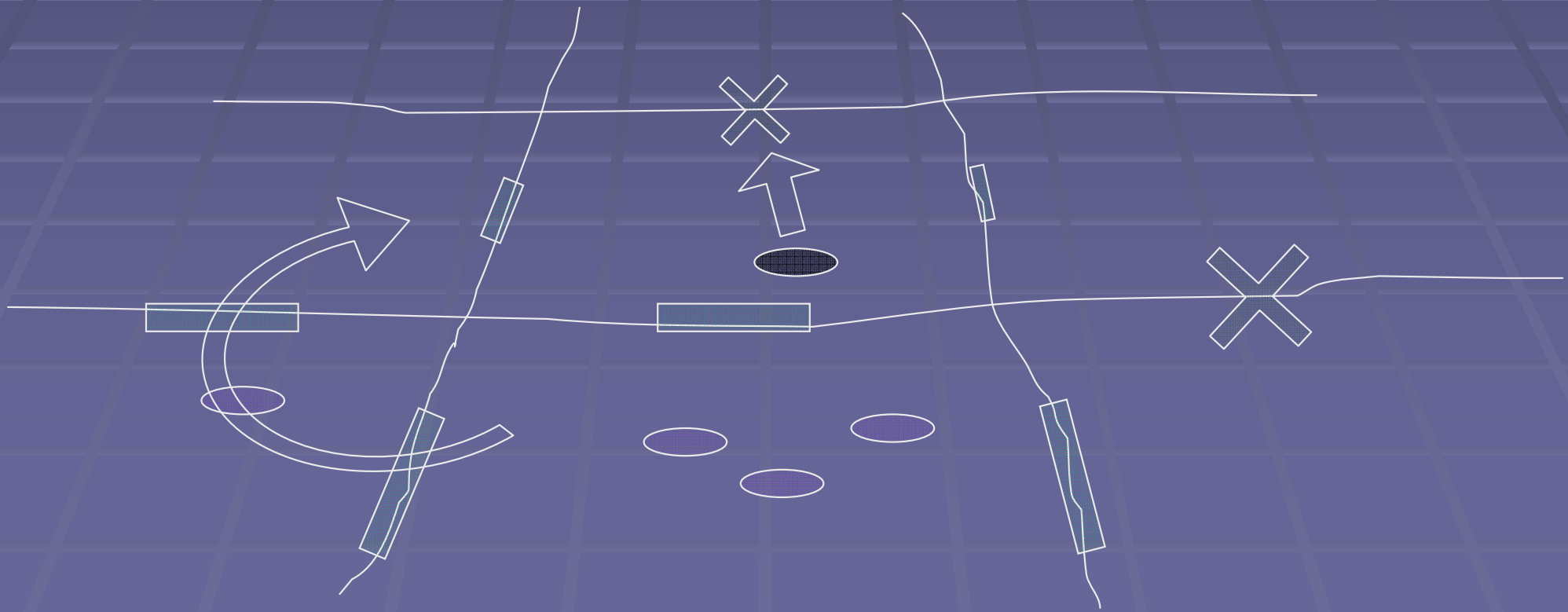


Combined Example



Our Game Play

- Patterning in puzzles
- Survival based
- Incorporation of flow



Technical Aspects

Physics

- Realistic, real world physics modeling
  - Objects have **position**, **velocity**, **orientation**, **angular velocity.**
  - For realistic physics simulation: **mass**, **inertia** (distribution of mass), **COM**.
  - **Forces** act on objects to make them change velocity.
  - Objects must collide and interact realistically.
  - We will use ODE (Open Dynamics Engine)

Player Interaction

- How will the player interact with world and objects?
- Player actions: pick-up/drop/push/pull/throw objects, run, jump, crouch etc.
- Models for object interaction:
  - “Gravity gun” – simple but not realistic.
  - Virtual hands – use mouse cursor to drag and drop objects around world.
- Some objects are heavy, can only push and pull these.

Half life 2 - gravity gun

Penumbra – virtual hand

Graphics

- Aim for a realistic look-and-feel.
- Reasonably detailed models and (photo realistic) textures.
- Basic special effects
  - Particles
  - Smoke
  - dynamic lighting
  - Water(?)

Enemy AI

- AI agents attempt to capture the player.
- May have simple weapons and tools (batons, tear gas).
- Use customized A* algorithm for path finding.
- Analyze possible decisions and chose the best one.
- React to player and use teamwork.
- Intelligently utilize objects and environment in a basic way.

Development Tools

Development Tools

- Visual Studio 2005
  - C/C++ implementation language
    - Fast
    - Good control over resources
    - Group has better familiarity than with C#
    - Optimizing compiler
- Use WinMerge for source control.

Game and Physics Engine

- Irrlicht
  - Cross-platform
  - Built in graphics and I/O libraries.
  - Comprehensive documentation
- ODE (Open Dynamics Engine)
  - Require mesh-to-mesh collision detection.
    - Simulates articulate rigid body physics
  - Cross platform (C++) and compatible with other game engines/simulators.

Models and Textures

- Blender (mesh modeling)
  - 3D mesh modeling tool
  - Able to create key frame animations
  - Mesh texturing
- GIMP (textures)
  - Seamless texture feature
  - Powerful open source image editor

Seamless and non-seamless textures

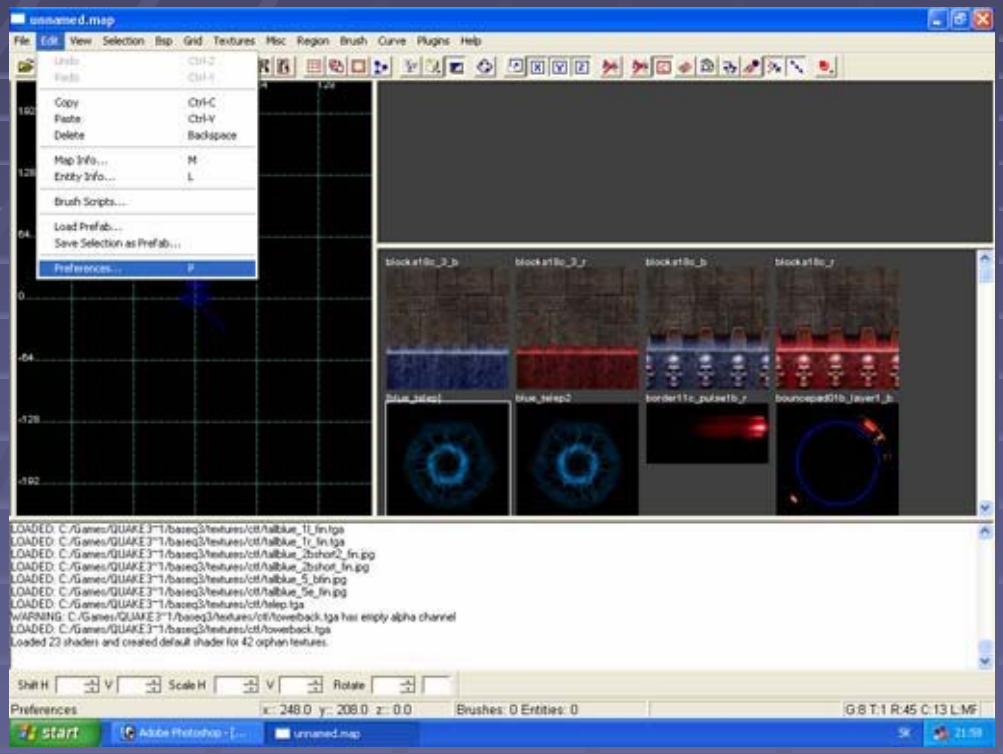
Non-Seamless

Seamless

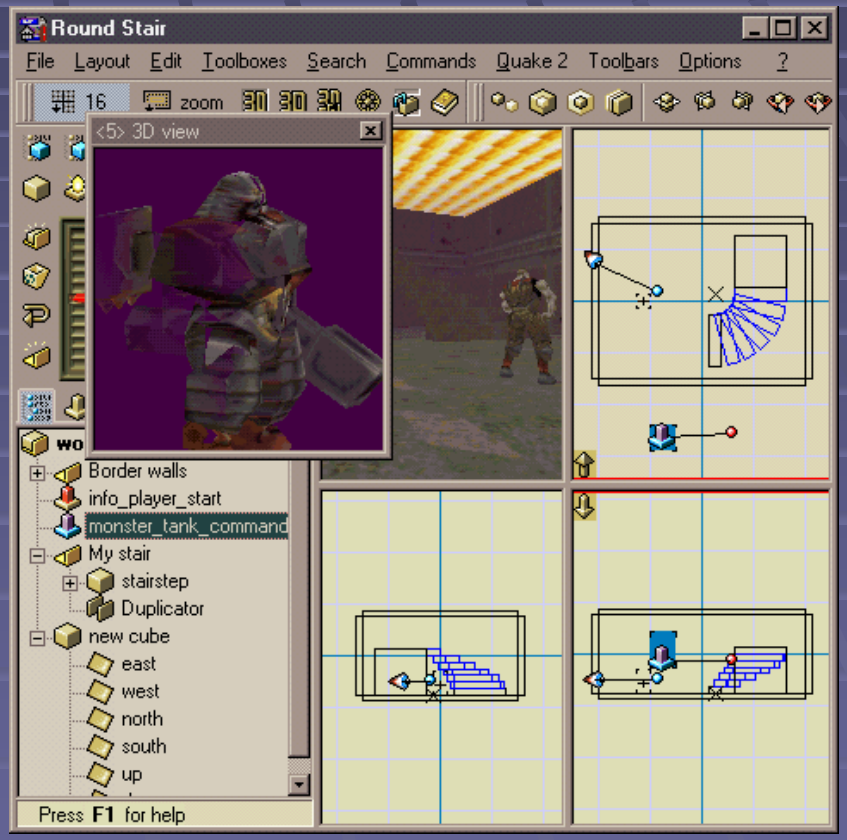
Map Construction

- GTK Radiant
  - Originally used for Quake maps
  - Compiles BSP maps
  - Powerful but primitive
- QuArK (Quake Army Knife)
  - High level
  - Doesn’t compile BSP
  - Used for map editing in many games

Radaint vs. QuArK



GTK Radiant



QuArK

Project Plan

- Week 1: Research, initial concept.
- Week 2: Research technology, integrate engine components.
- Week 3-5: Develop AI, interaction model, apply physics engine.
- Week 6: Prepare interim presentation and tech demo (test maps)
- Week 7-11: Further refinement/development and creation of final game assets.
- Week 11-12: Prepare final presentations and demos

Learning Objectives

- Integration and utilization of game/physics engine.
- Development of AI which can respond dynamically to a complex environment and use teamwork.
- Exploring new and innovative gameplay concepts with iterative development.

TECH DEMO