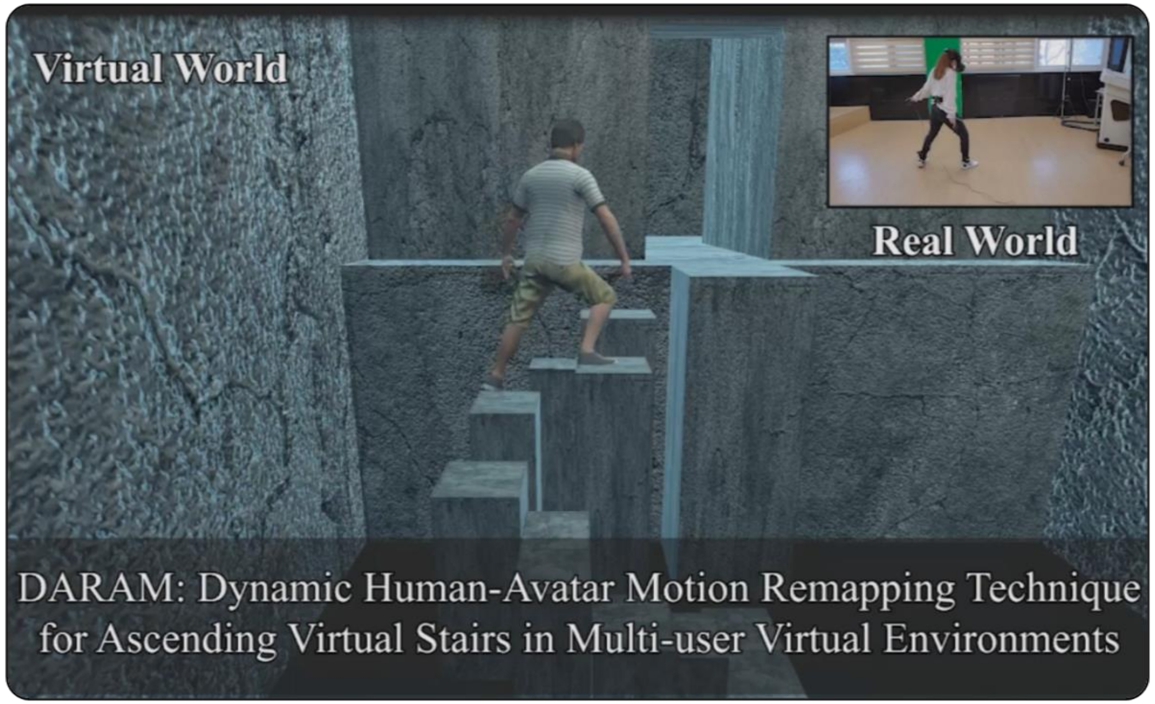
0. Introduction

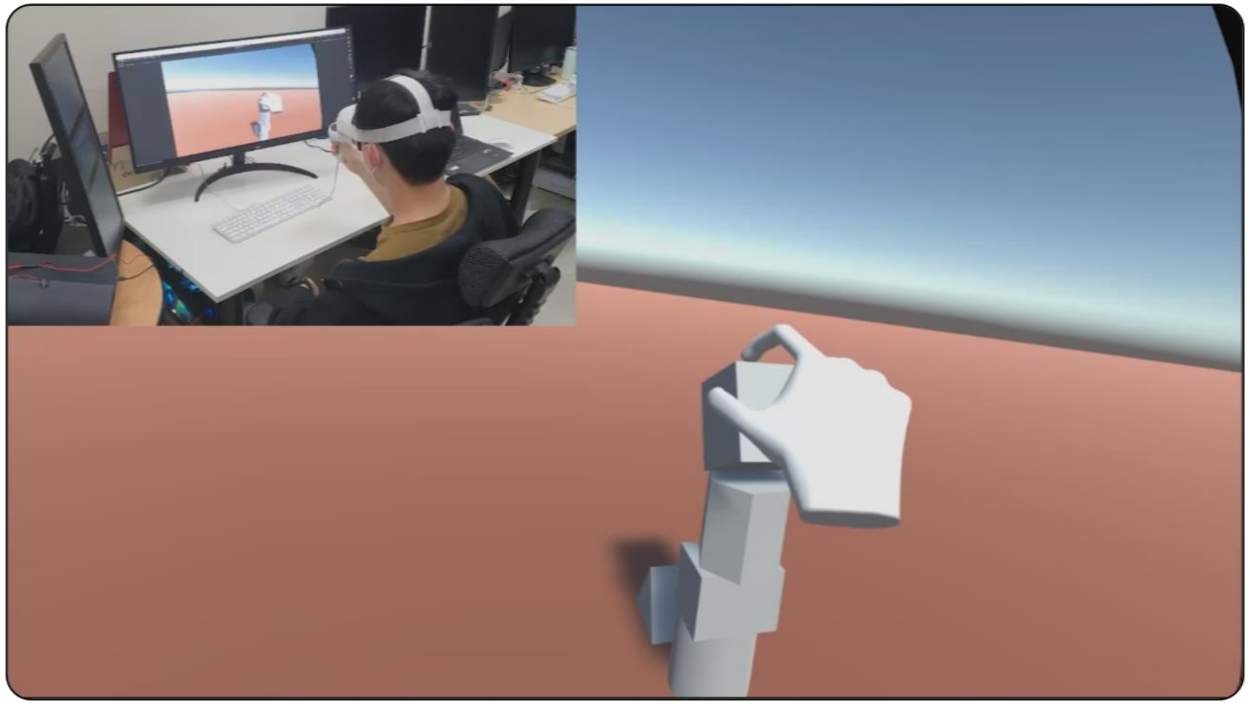
Game Engineering
Prof. HyeongYeop Kang
siamiz@khu.ac.kr
IIIXR LAB

H. Kang?

HyeongYeop Kang

- ***Research Field***: Extended Reality (virtual reality, mixed reality, augmented reality, etc.), Computer Graphics, Human-computer Interaction, Reinforcement Learning
- ***Courses***: Game Graphics, Game Engineering, Game PX Design, etc.
- ***Office***: 전자정보대학관 314
- ***Email***: siamiz@khu.ac.kr
- ***Homepage***: https://siamiz88.github.io/, http://iiixr.khu.ac.kr/





Class Overview

Ultimate goal

- (Theory) Understand the advanced game engine architecture.
- (Practice) Experience the development of game engine and advanced algorithms
- (Term Project) Propose an advanced algorithm for game.

Evaluation

- Team project - interim (15%)
- Team project - final (25%) – must submit a showcase movie.
- Final exam (40%)
- Homework (10%)
- Additional points – MVP, leader, etc. (10%)

Class Overview

Each week (or several weeks) will cover separate topic.

- We learned overall architecture of the game engine in the Game Engine Basics class.
- This class focused on more in-depth topics.

Topic example:

- Optimization
- Inverse kinematics
- Statistics
- Reinforcement learning
- ...

Term Project Overview

Team building

- 4~6 students per team.
- One team leader (+2 additional points)
- MVP (optional, +3 additional points, if recommended by at least 2 team members)
- Black list (optional, -5 deduction, if reported by at least 2 team members)

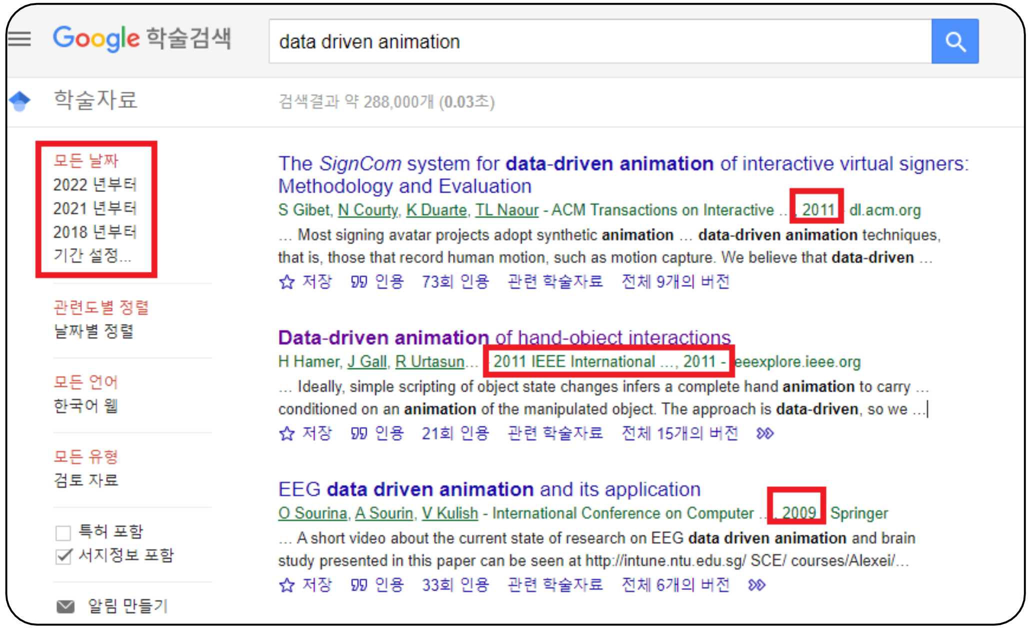
Team building process

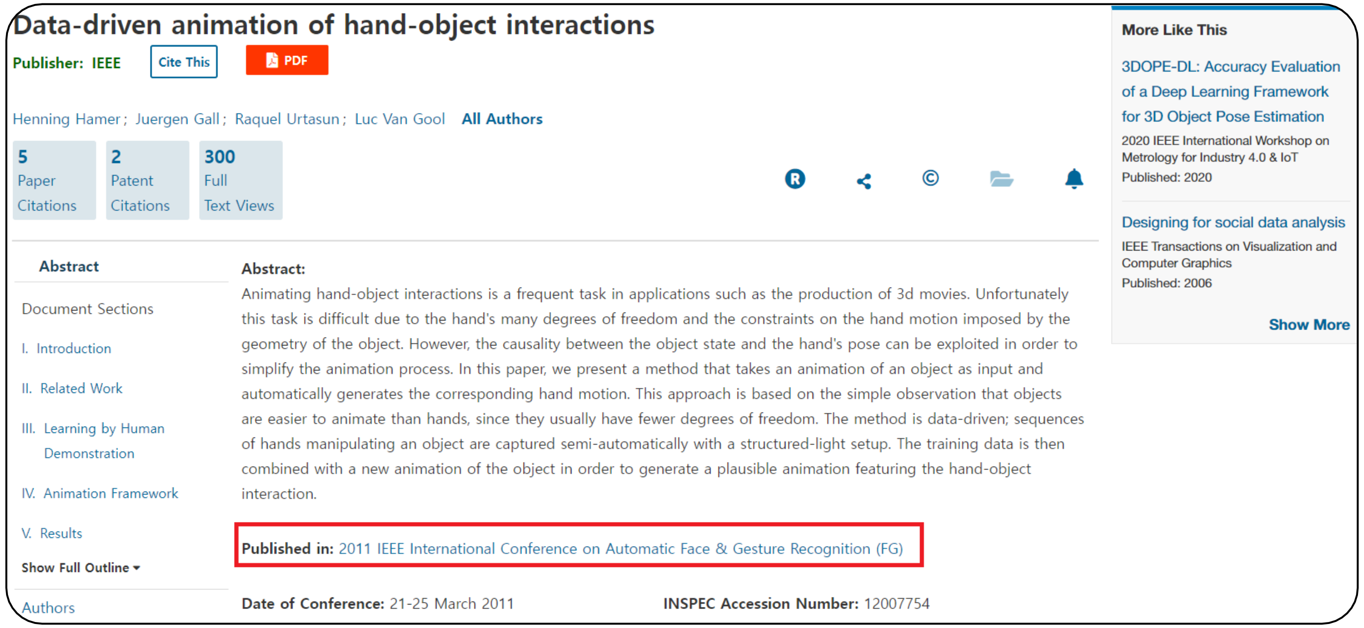
- You can choose up to one friend you want to be with (important!)
- The team will be constructed automatically. (manually by Prof..)

Term Project Overview

Project goal

- Based on the previous 'research paper', propose a novel idea/algorithm that seem necessary for future games.
  - Research papers published after 2010 should be selected.
  - Articles written in Korean are not allowed.
- Develop and visualize your algorithm in Unreal Engine.

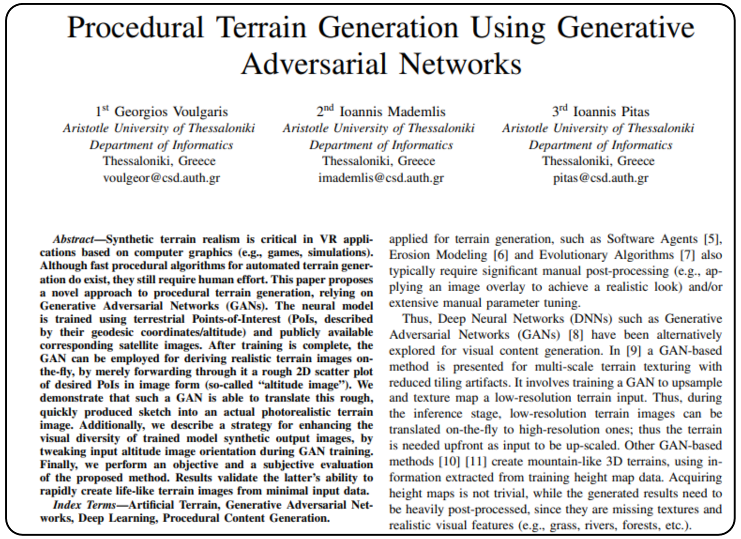




Term Project Overview

Project topic examples

- Procedural terrain rendering



Procedural Terrain Generation Using Generative Adversarial Networks

1st Georgios Voulgaris
Aristotle University of Thessaloniki
Department of Informatics
Thessaloniki, Greece
voulgeor@csd.auth.gr

2nd Ioannis Mademlis
Aristotle University of Thessaloniki
Department of Informatics
Thessaloniki, Greece
imademlis@csd.auth.gr

3rd Ioannis Pitas
Aristotle University of Thessaloniki
Department of Informatics
Thessaloniki, Greece
pitas@csd.auth.gr

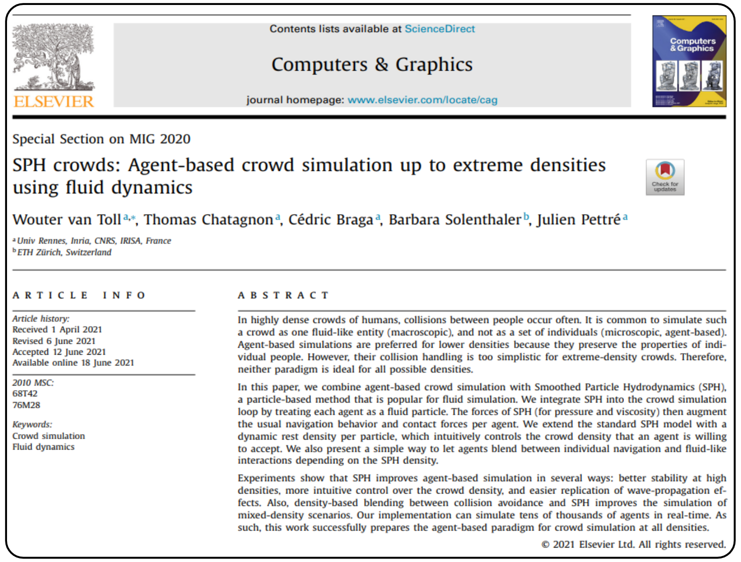
***Abstract*—Synthetic terrain realism is critical in VR applications based on computer graphics (e.g., games, simulations). Although fast procedural algorithms for automated terrain generation do exist, they still require human effort. This paper proposes a novel approach to procedural terrain generation, relying on Generative Adversarial Networks (GANs). The neural model is trained using terrestrial Points-of-Interest (PoIs, described by their geodesic coordinates/altitude) and publicly available corresponding satellite images. After training is complete, the GAN can be employed for deriving realistic terrain images on-the-fly, by merely forwarding through it a rough 2D scatter plot of desired PoIs in image form (so-called "altitude image"). We demonstrate that such a GAN is able to translate this rough, quickly produced sketch into an actual photorealistic terrain image. Additionally, we describe a strategy for enhancing the visual diversity of trained model synthetic output images, by tweaking input altitude image orientation during GAN training. Finally, we perform an objective and a subjective evaluation of the proposed method. Results validate the latter's ability to rapidly create life-like terrain images from minimal input data.**

***Index Terms*—Artificial Terrain, Generative Adversarial Networks, Deep Learning, Procedural Content Generation.**

applied for terrain generation, such as Software Agents [5], Erosion Modeling [6] and Evolutionary Algorithms [7] also typically require significant manual post-processing (e.g., applying an image overlay to achieve a realistic look) and/or extensive manual parameter tuning.

Thus, Deep Neural Networks (DNNs) such as Generative Adversarial Networks (GANs) [8] have been alternatively explored for visual content generation. In [9] a GAN-based method is presented for multi-scale terrain texturing with reduced tiling artifacts. It involves training a GAN to upsample and texture map a low-resolution terrain input. Thus, during the inference stage, low-resolution terrain images can be translated on-the-fly to high-resolution ones; thus the terrain is needed upfront as input to be up-scaled. Other GAN-based methods [10] [11] create mountain-like 3D terrains, using information extracted from training height map data. Acquiring height maps is not trivial, while the generated results need to be heavily post-processed, since they are missing textures and realistic visual features (e.g., grass, rivers, forests, etc.).

- Crowd simulation



Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

ELSEVIER

Special Section on MIG 2020

SPH crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics

Wouter van Toll[a,*], Thomas Chatagnon[a], Cédric Braga[a], Barbara Solenthaler[b], Julien Pettré[a]

[a] Univ Rennes, Inria, CNRS, IRISA, France
[b] ETH Zürich, Switzerland

ARTICLE INFO

Article history:
Received 1 April 2021
Revised 6 June 2021
Accepted 12 June 2021
Available online 18 June 2021

2010 MSC:
68T42
76M28

Keywords:
Crowd simulation
Fluid dynamics

ABSTRACT

In highly dense crowds of humans, collisions between people occur often. It is common to simulate such a crowd as one fluid-like entity (macroscopic), and not as a set of individuals (microscopic, agent-based). Agent-based simulations are preferred for lower densities because they preserve the properties of individual people. However, their collision handling is too simplistic for extreme-density crowds. Therefore, neither paradigm is ideal for all possible densities.

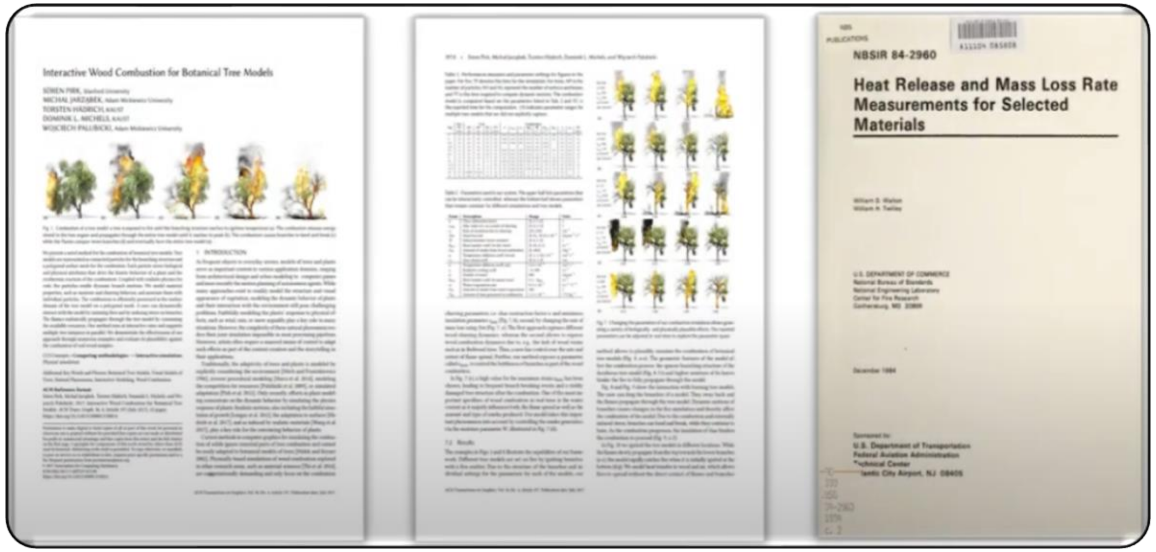
In this paper, we combine agent-based crowd simulation with Smoothed Particle Hydrodynamics (SPH), a particle-based method that is popular for fluid simulation. We integrate SPH into the crowd simulation loop by treating each agent as a fluid particle. The forces of SPH (for pressure and viscosity) then augment the usual navigation behavior and contact forces per agent. We extend the standard SPH model with a dynamic rest density per particle, which intuitively controls the crowd density that an agent is willing to accept. We also present a simple way to let agents blend between individual navigation and fluid-like interactions depending on the SPH density.

Experiments show that SPH improves agent-based simulation in several ways: better stability at high densities, more intuitive control over the crowd density, and easier replication of wave-propagation effects. Also, density-based blending between collision avoidance and SPH improves the simulation of mixed-density scenarios. Our implementation can simulate tens of thousands of agents in real-time. As such, this work successfully prepares the agent-based paradigm for crowd simulation at all densities.

Term Project Overview

Project topic examples

- Smoothed-particle Hydrodynamics with Unreal Engine

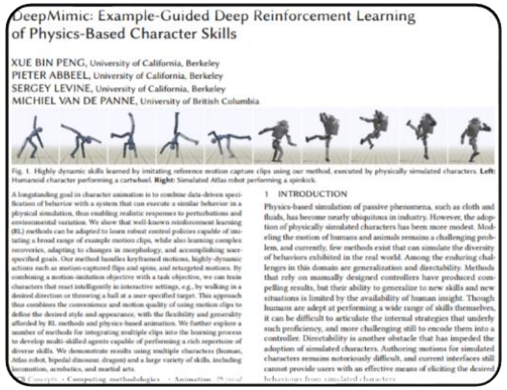


Interactive Wood Combustion for Botanical Tree Models

NBSIR 84-2960

Heat Release and Mass Loss Rate Measurements for Selected Materials

- Reinforcement Learning for Character Animation



DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills

XUE BIN PENG, University of California, Berkeley
PIETER ABBEEL, University of California, Berkeley
SERGEY LEVINE, University of California, Berkeley
MICHIEL VAN DE PANNE, University of British Columbia

Fig. 1. Highly dynamic skills learned by imitating reference motion capture clips using our method, executed by physically simulated characters. **Left:** Humanoid character performing a cartwheel. **Right:** Simulated Atlas robot performing a spinkick.

A longstanding goal in character animation is to combine data-driven specification of behavior with a system that can execute a similar behavior in a physical simulation, thus enabling realistic responses to perturbations and environmental variation. We show that well-known reinforcement learning (RL) methods can be adapted to learn robust control policies capable of imitating a broad range of example motion clips, while also learning complex recoveries, adapting to changes in morphology, and accomplishing user-specified goals. Our method handles keyframed motions, highly-dynamic actions such as motion-captured flips and spins, and retargeted motions. By combining a motion-imitation objective with a task objective, we can train characters that react intelligently in interactive settings, e.g., by walking in a desired direction or throwing a ball at a user-specified target. This approach thus combines the convenience and motion quality of using motion clips to define the desired style and appearance, with the flexibility and generality afforded by RL methods and physics-based animation. We further explore a number of methods for integrating multiple clips into the learning process to develop multi-skilled agents capable of performing a rich repertoire of diverse skills. We demonstrate results using multiple characters (human, Atlas robot, bipedal dinosaur, dragon) and a large variety of skills, including locomotion, acrobatics, and martial arts.

CCS Concepts: • Computing methodologies → Animation; Physical

1 INTRODUCTION

Physics-based simulation of passive phenomena, such as cloth and fluids, has become nearly ubiquitous in industry. However, the adoption of physically simulated characters has been more modest. Modeling the motion of humans and animals remains a challenging problem, and currently, few methods exist that can simulate the diversity of behaviors exhibited in the real world. Among the enduring challenges in this domain are generalization and directability. Methods that rely on manually designed controllers have produced compelling results, but their ability to generalize to new skills and new situations is limited by the availability of human insight. Though humans are adept at performing a wide range of skills themselves, it can be difficult to articulate the internal strategies that underly such proficiency, and more challenging still to encode them into a controller. Directability is another obstacle that has impeded the adoption of simulated characters. Authoring motions for simulated characters remains notoriously difficult, and current interfaces still cannot provide users with an effective means of eliciting the desired behaviours from simulated characters

Term Project Overview

How to select a topic?

- Key-word based topic selection
  - Go to ‘scholar.google.co.kr’ and enter keywords
- Contents based topic selection
  - Visit popular conference, or journal page such as SIGGRAPH, CVPR, CHI, IEEEVR, EG, PG, CGI, etc. (dblp also provides all papers)
  - Visit popular youtube page such as 2minute paper
  - Visit IIIXR youtube.