**TEMASEK POLYTECHNIC**

**SCHOOL OF INFORMATICS & IT**

**DIPLOMA IN GAME DESIGN & DEVELOPMENT**

**AY2023/2024 OCTOBER SEMESTER (LEVEL 2) TERM A**

**GAME MATH AND PHYSICS (CGE2C15) TERM A**

**Project Guide**

*This document was partly made with the help of ChatGPT. You may use ChatGPT to help understand your group’s topic, and to structure the content in your presentation slides, but you CANNOT use ChatGPT to generate the actual text, code, or other contents for any component(s) submitted for your project. If there is a strong indication that you have done so, you will score 0 marks for the component(s).*

This document will provide guidance for your GMAPS project.

You must create a group presentation about your chosen topic. Each student must present either:

* Theory
* Unity implementation
* Case Study

An example of what content might be included for each section is given below. The example is for rope physics (swinging & grappling), which is *not* included as one of the project topics.

The content for each section must be presented as a set of PowerPoint slides. For each section, this content has been structured into 10 slides for each section.

Note: the Content section for each slide below lists only the main topics for that slide, not the contents. You need to make sure that your slides are informative and include all necessary information, with diagrams and other illustrations where appropriate.

You may use this document as a guide, but you may also decide to structure your presentations in a different way.

* Part (a) of your presentation should be no more than 10 minutes per student.
* Part (b) of your presentation should be no more than 10 minutes per student.

**Presentation Part (a)**

**SECTION 1: THEORY (Content)**

**Overview**: Swinging and grappling mechanics are a popular gameplay feature, allowing players to traverse environments dynamically, often seen in titles like "Spider-Man" or "Sekiro: Shadows Die Twice". These mechanics rely on simulated physics to achieve fluid, realistic motion.

**Main Problem**: The challenge lies in creating a natural arc of motion while grappling, considering factors like the player's momentum, length of the grappling line, and point of attachment. The system must react to player input while anchored in realistic physics.

**Math and Physics Involved**:

1. **Pendulum Motion**: Swinging, at its core, mimics the motion of a pendulum. The length of the grappling line acts as the pendulum's rod, and the player is the pendulum's bob. This motion is governed by the formula for pendular period: T = 2π√(L/g) Where:

* T = period of the pendulum
* L = length of the grappling line
* g = acceleration due to gravity

1. **Centripetal Force**: To keep the player moving in a circular arc when swinging, there's a need for a centripetal force directed towards the centre of the circle. This force is provided by the tension in the grappling line.
2. **Momentum Conservation**: As the player swings, momentum is conserved. If the player releases the grappling line at any point, they will move tangentially to their last position in the arc, based on their built-up momentum.

**THEORY (10 slides)**

**Slide 1**:

* **Title**: Overview
* **Content**:
  + Introduction to the mechanics of swinging and grappling.
  + Importance in creating realistic and dynamic gameplay.

**Slide 2**:

* **Title**: Pendulum Basics
* **Content**:
  + Swinging acts like a pendulum.
  + The period depends on length and gravity.

**Slide 3**:

* **Title**: Pendulum Period
* **Content**:
  + Formula: Period (T) = 2 times pi times the square root of (length divided by gravity).
  + Explanation of variables involved.

**Slide 4**:

* **Title**: Tension in Swinging
* **Content**:
  + Tension changes throughout the swing.
  + Highest at the lowest point of the swing.

**Slide 5**:

* **Title**: Tension & Centripetal Force
* **Content**:
  + Formula: Tension (T) = weight + (mass times velocity squared divided by radius).
  + Explanation of centripetal force.

**Slide 6**:

* **Title**: Rope Elasticity
* **Content**:
  + Not all ropes are rigid; some have elasticity.
  + This elasticity can affect the swing.

**Slide 7**:

* **Title**: Hooke's Law
* **Content**:
  + Formula: Force (F) = spring constant times change in length.
  + How elasticity affects tension in the rope.

**Slide 8**:

* **Title**: Player Control
* **Content**:
  + Players can influence swing dynamics.
  + Adjusting rope length, jumping off, and timing affects motion.

**Slide 9**:

* **Title**: External Factors
* **Content**:
  + Wind, objects in the path, and player's movements can influence the swing.
  + These must be factored into the physics calculations.

**Slide 10**:

* **Title**: Summary
* **Content**:
  + Swinging/grappling mechanics are a combination of pendulum physics, tension, and elasticity.
  + These principles guide game implementation.

**SECTION 2:** **UNITY IMPLEMENTATION (Content)**

**Overview:** Unity's joint system provides an intuitive approach to simulate physics-based connections, making it a good solution for implementing swinging and grappling mechanics. These joints can create and maintain connections between objects, allowing for realistic motion dynamics.

**Core Components and Steps:**

1. **Character Setup:** Begin with the player character having a Rigidbody component, enabling it to interact with Unity's physics system.
2. **Line Renderer:** Utilize the Line Renderer component to visually represent the grappling line. Its start and endpoints can dynamically adjust based on the player's position and the grappling target.
3. **Raycasting for Grappling Point:** When the player initiates a grapple, use raycasting to determine the target point of attachment. This detection will determine where the joint will be anchored.
4. **Spring Joint or Hinge Joint:**
   * **Spring Joint:** This joint will try to maintain its distance between connected objects but allows for stretching, which can simulate the dynamic tension and flexibility of a grappling line. Adjust properties like spring, damper, and max distance to fine-tune the swinging mechanics.
   * **Hinge Joint:** This joint rotates around a single axis and can be useful if you want a more rigid swinging motion without the stretching characteristic of a spring joint.
5. **Scripted Force Application:** When the player is connected via the joint, monitor user input to influence the swinging motion. Players might want to gain momentum, change direction, or adjust height. Apply subtle forces to the Rigidbody to simulate these actions.
6. **Dynamic Joint Creation & Destruction:** Once the grapple point is detected and the player chooses to swing or grapple, instantiate the joint and connect it to the player. When the player releases or reaches their destination, destroy the joint to free the character.
7. **Collisions and Obstacles:** With the joint-based system, much of the collision response will be handled naturally by Unity's physics. Still, it's essential to fine-tune the player's reactions to hitting obstacles or the ground while swinging.

**UNITY IMPLEMENTATION (10 slides)**

**Slide 1**:

* **Title**: Overview
* **Content**:
  + Unity's capabilities in simulating swinging and grappling.
  + Using physics components and scripts for desired effects.

**Slide 2**:

* **Title**: Rigidbody & Collider
* **Content**:
  + Importance of Rigidbody for physics interactions.
  + Colliders for detecting grappling points.

**Slide 3**:

* **Title**: Visualizing the Rope
* **Content**:
  + Using the Line Renderer.
  + Adjusting dynamically for rope length and trajectory.

**Slide 4**:

* **Title**: Joints in Unity
* **Content**:
  + Introduction to Distance Joint 2D and Spring Joint.
  + Their roles in simulating tension and elasticity.

**Slide 5**:

* **Title**: Player Control Scripts
* **Content**:
  + Detecting valid grappling points.
  + Real-time calculation of rope dynamics.

**Slide 6**:

* **Title**: Adjusting the Swing
* **Content**:
  + Allowing player input to influence the swing.
  + Changing rope length and controlling momentum.

**Slide 7**:

* **Title**: Particle Systems
* **Content**:
  + Enhancing visual feedback.
  + Effects for grappling, swinging, and detaching.

**Slide 8**:

* **Title**: Audio Integration
* **Content**:
  + Immersing players with sound.
  + Rope, wind, and latch sounds for realism.

**Slide 9**:

* **Title**: Challenges & Solutions
* **Content**:
  + Common challenges in implementing swinging.
  + Tips and solutions for smoother gameplay.

**Slide 10**:

* **Title**: Summary
* **Content**:
  + Unity offers a robust set of tools for swinging/grappling mechanics.
  + With careful tuning, developers can achieve realistic and dynamic gameplay.

**SECTION 3: CASE STUDY (Content)**

1. **Three example games**
2. **Spider-Man by Insomniac Games**

"Spider-Man" emphasizes fluidity in its web-swinging mechanic. Players traverse New York City, attaching webs to real buildings, allowing for momentum-based movement. The traversal is dynamic, with contextual actions like skimming above the ground. Swinging is also integrated into aerial combat, enhancing versatility.

1. **Sekiro: Shadows Die Twice by FromSoftware**

In "Sekiro," the grappling hook introduces vertical gameplay using predetermined points, ensuring controlled movement. The mechanic is essential for stealth approaches and dynamic combat, allowing players to scale obstacles, reposition rapidly, and engage foes from different vantage points.

1. **Just Cause Series by Avalanche Studios**

The "Just Cause" series offers a multi-purpose grappling hook. Protagonist Rico Rodriguez can tether objects, pull enemies, and slingshot across the map. The mechanic encourages player creativity, serving both as a traversal tool and a means to create chaotic, sandbox scenarios.

**2. In-depth analysis of "Spider-Man" Series by Insomniac Games**

**PlayStation 4 and PlayStation 5** Spider-Man" series, particularly their titles for the PlayStation 4 and PlayStation 5, are heralded for their intuitive and fluid web-swinging mechanics. As a core gameplay feature, swinging through New York City as Spider-Man required a deep understanding of physics and player expectation.

**The Challenges**:

1. **Realism vs. Playability**: Striking a balance between a realistic swinging mechanic and one that feels good and responsive for the player is challenging. Too much realism can lead to cumbersome controls, while too much freedom can break immersion.
2. **Urban Environment**: New York City's skyline is irregular. Developers had to ensure Spider-Man's webs attached to actual buildings and structures, not the sky.
3. **Pace and Flow**: Players needed to maintain momentum, quickly pivot directions, and have a series of moves at their disposal to navigate the environment smoothly.

**Insights & Solutions**:

1. **Dynamic Attach Points with Raycasting**: Insomniac used raycasting to determine where Spider-Man's webs would attach. If a building or structure was in range, the web would attach there. This approach grounded the web-swinging in the game world's physicality.
2. **Pendulum Physics with Unity-like Joints**: While not explicitly stated that Insomniac used Unity's joint systems, the principles they employed mirror those systems. Spider-Man acts as the bob of a pendulum, with the web determining the arc and length of the swing. This approach, combined with the ability to adjust swing height and length on-the-fly, offers players control and responsiveness.
3. **Player-Centric Tweaks**: Insomniac made several player-centric decisions to enhance playability. For instance, if players swing close to the ground, Spider-Man automatically performs a move to prevent a crash, maintaining momentum.
4. **Animation Blending**: Swinging animations blend seamlessly with other actions, like wall-running or diving. This fluidity allows players to chain moves together, creating a dynamic traversal system.
5. **Contextual Actions**: Depending on the environment and player input, Spider-Man will perform different swinging moves. Near a building corner, he might whip around it; close to the ground, he might pull off a low swing.

**Conclusion**:

Insomniac Games' "Spider-Man" offers a master class in grappling and swinging mechanics, showing that a balance of realistic physics, player-centric design, and fluid animations can create a system that's both immersive and exhilarating.

**CASE STUDY (10 slides)**

**Slide 1:**

* **Title**: Overview
* **Content**:
  + Introduction to the three games with grappling/swinging mechanics: Spider-Man, Sekiro, and Just Cause.
  + The role of physics and player expectations in creating a compelling swinging mechanic.

**Slide 2:**

* **Title**: Spider-Man's Unique Web-Swinging
* **Content**:
  + Emphasis on fluid traversal across New York City.
  + Dynamic integration of swinging into aerial combat.
  + Contextual actions enhancing the movement experience.

**Slide 3:**

* **Title:** Sekiro’s Vertical Movement
* **Content:**
  + Role of the grappling hook in introducing vertical gameplay.
  + The importance of controlled movement using predetermined points.
  + Engaging foes and repositioning from different vantage points.

**Slide 4:**

**• Title:** Just Cause’s Creative Chaos

**• Content:**

* + - Rico Rodriguez's multi-purpose grappling hook capabilities.
    - Encouraging player creativity with traversal and sandbox scenarios.
    - Dual functions: movement and chaos creation.

**Slide 5:**

• **Title**: Challenges in Designing Spider-Man's Mechanics

• **Content**:

* + Balancing realism and playability.
  + Handling New York City's irregular skyline.
  + Ensuring a smooth pace and flow in movement.

**Slide 6:**

• **Title**: Raycasting & Dynamic Attach Points

• **Content**:

* + Role of raycasting in determining web attachment points.
  + Ensuring webs attach to actual physical structures.

**Slide 7:**

• **Title**: The Physics Behind Spider-Man’s Swing

• **Content**:

* + The pendulum physics principles.
  + Spider-Man as the bob, with webs influencing swing arc and length.
  + Adjustability in swing height and length for user control.

**Slide 8:**

• **Title**: Enhancing Player Experience

• **Content**:

* Player-centric tweaks like avoiding ground crashes.
* Seamless animation blending, from swinging to wall-running and diving.

**Slide 9:**

• **Title**: Contextual Swinging Actions

• **Content**:

* + Adaptive swinging moves based on player's environment.
  + Examples like whipping around building corners or executing a low swing.

**Slide 10:**

• **Title**: Insomniac’s Mastery & Conclusion

• **Content**:

* + Achieving a balance of realism, player-centric design, and fluid animations.
  + Lessons for developers in prioritizing player experience while retaining realism.

**Presentation Part (b)**

For Part (b), which should come after Part (a), you must each present an individual section that illustrates and discusses your Unity implementation of your chosen topic.

You must discuss:

* The components used;
* Any code written;
* The main *technical* problems you faced, and how you overcame them (or not; and if not, why not).

This section should be about 5-10 minutes each.