GDP Game Jam 2018: Wednesday, April 25th, 2018

|  |  |
| --- | --- |
| Student name(s): | |
| Jorge Amengol | |
|  | |
|  | |
| Game name: | From the Graveyard to the Forest |

Description:

|  |
| --- |
| Inspired by one of the most iconic arcade games from the 80’s, this Tech Demo depicts the first stage of Ghosts ‘n Goblins: From the Graveyard to the Forest. Take Arthur’s sword and fight against the guardian beasts. Try to take down some statues too! ☺ |

Feedback, ranking, and recommendations (0-5: think “0 stars”, “5 stars”, etc.):

Keep in mind that these developers are:

* Here for programming, not art
* Are \*not\* using a “game engine” – everything here is implemented “from scratch”
* The only exception would be things like the Havok Physics engine, but even that is a huge struggle to integrate
* Think of these as early “Early Alphas”, “Early ‘Green Light’”, or “Proof of concept” sort of designs, not complete, polished products ready to hit store shelves...

|  |  |
| --- | --- |
| Fun? | |
| Close to what you expected (from name+desctiption)? | |
| Game play (nightmarish, awkward, good, etc.)? | |
| Appealing? | |
| Like to see more of it when it’s “done”? | |
| Anything else (Balance, etc.)? | |
| Evaluated by: |  |

|  |
| --- |
| GDP Game Jam 2018: Wednesday, April 25th, 2018  Student name(s): |
| Jorge Amengol |
|  |
|  |

|  |  |
| --- | --- |
| Game name: | From the Graveyard to the Forest |

Description:

|  |
| --- |
| Inspired by one of the most iconic arcade games from the 80’s, this Tech Demo depicts the first stage of Ghosts ‘n Goblins: From the Graveyard to the Forest. Take Arthur’s sword and fight against the guardian beasts. Try to take down some statues too! ☺ |

Technical “things” implemented (see next page for a suggested list):

|  |  |
| --- | --- |
| 1. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| 2. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| 3. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| 4. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| 5. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| 6. | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |
| (7.) | |
| Easy (75%), Hard (100%), or Super Hard (115%)? | Mark (could be zero): |

|  |  |
| --- | --- |
| Evaluated by: |  |

**Relevant Technical Aspects**

**Day/Night Cycle and Environment control**

Implemented a fully Day/Night cycle with:

* Time of the day control
* Multiple sky maps seamless blending the scene
* Control for the Sun and Moon Light at the right position of the "Dummy Sun/Moon Texture"
* Moon and Sun lights change colours during the cycle to match the colour of the sky
* Shadows that also follows the Sun/Moon position (see next item)

**Shadows**

Implemented Shadows with transparency (foliage):

* Using Ortho projection, render to a Colour Frame Buffer using the Alpha textures (foliage) of the scene
* Using Ortho projection, render to a Depth Frame Buffer. Discard fragments of the Depth that are in a the alpha threshold
* Implement PCF (percentage-closer filtering) to soft the shadows
* Change the Shadow's camera position for rendering according to the environment control

**Multi-layer texture**

Implement a multi-layer texture for the ground

* The ground is a blended from 3 textures (with normal maps too)
* A forth texture with different texture coordinates commands the blend between them according its RGB channel

**Scene Control**

Implement a scene control to handle Menus, Background music and transitions

**NPC mini system**

Implemented a NPC mini system with "Guardians" behaviors

* The NPCs turn to the character correctly
* The NPCs use different attacks depending on the distance from the character

**Deferred render - Old-film effect**

Implement and "old-film" effect using a texture displacement combined with a Sepia tone base for the main menu (that is originally coloured)

**Physics**

Physics is being driven by Bullet indirectly by a Physics interface

* Implement a debug drawing using a different shader
* Debug drawing is being called from own Bullet's DebugDrawWorld()
* Scene with various objects as convex hulls
* Various objects reacts to the character
* Characters implemented as a regular Bullet capsule

**FOG Effect**

Implement a fog effect that fades during the day and increases at night