Milestone 2 – Physics Simulation and Game Feel Gardens

This is an *INDIVIDUAL* assignment.

Due: September 23, 2016 @ 11:55 PM

Late Policy
See milestone 1 for the late policy (2^(n+1) late policy).

Description
This assignment involves the modification of your controllable character from milestone 1. You will modify your character such that it can interact with a physically simulated world in such a way that it meets some aspects of Game Feel. Additionally, you will create two Game Feel gardens for your character to interact within.

Character Controller Updates

For the first objective you will update your character controller to support physical interactions. This will be based on extending the work you completed in Milestone 1. Feel free to abandon the special grading keyboard controls and implement more conventional keyboard controls, possibly also including gamepad support.

Your character should interact with physical objects, most likely through use of a capsule collider and rigid body component. The collider should provide reasonable collisions with objects in the environment with minimal interpenetration of 3D models. For instance, your character should be able to bump into rigid body objects in a scene (e.g. push crates or spherical objects around). The specific interactions are up to you to decide upon and implement.

Footstep Sounds and Particle Effects

Your character should make footsteps based on the speed at which he/she walks/runs. This can be accomplished by creating a Mecanim animation curve that exposes a value that specifies whether a foot is touching the ground or not. Then, in script you can watch this value to decide when to play a footstep sound and create a particle effect, such as dust being kicked up.

You will also need to access the character's skeleton to get the world coordinates of the foot for placing the particle effect. Both the footstep sound and particle effect should be dependent on the type of ground surface stepped upon. For instance, sand should make a sound different than mud or a wooden floor. The particle effect can be reserved for only one of your ground material types per Game Feel garden (see below), but feel free to make more than that.

Rag Doll Physics

Your character should utilize a ragdoll mode for some aspect of interaction within your Game Feel gardens. For instance, if your character can die or become briefly stunned, then upon incapacitation your animated character should become a humanoid ragdoll (no longer controlled by the Mecanim animation system and instead simulate weighted limbs under the influence of gravity and other forces). We recommend use of the Ragdoll Wizard in Unity and perhaps a conveniently placed staircase.

Game Feel Gardens

You will be creating two Game Feel gardens (as scenes in Unity) that leverage the physics interactions of your character controller. Game Feel gardens are environments meant to showcase a game character with realtime control. The user can explore the environment without any particular goal other than to play around with the controls. Each environment provides interaction possibilities unique to that environment. For instance, an icy world may be slippery, a canyon region may have surging wind that affects the player, etc.

You can either implement a single scene with two distinct areas for your Game Feel gardens, or you can create two separate scene files selectable by the 1 and 2 number keys on the keyboard. In either case, tell the grader what to expect in your readme.

Each garden will provide a unique theme/biome in which the player can interact. Additionally, each garden must provide a variety of game objects that are part of the physics simulation and that the player can influence in some way. Furthermore, your gardens should provide some reasonable level of polish including textured surfaces, a skybox, lighting effects, and sound effects.

Consider using a level editor plugin for Unity such as ProBuilder Basic along with various prefab objects you find on the asset store.

Note that the HUD must display your full name.

Grading Criteria

Your submission should satisfy the following requirements.

Character Control

Basic Physics Interaction **(10 points)**
Footstep sounds and particle effects dependent upon ground surface **(20 points)**
Ragdoll Simulation **(20 points)**

Game Feel Gardens

Each garden represents a unique environment visually, auditorily, and interactively (via physical simulation). For example, you could make a snow world and a sand world.

Each Game Feel garden must exhibit the following environmental behaviors:

- Each garden should have at least five **unique** objects (geometry nodes with rigid bodies and collider shapes) in it that have dynamic physical properties controlled by the physics engine. In addition to rigid bodies, you can optionally implement soft bodies, cloth, etc. Examples include: crates, boulders, weighted companion cubes, happy fun ball, etc. **(10 pts)**
- Each garden should additionally have at least one unique compound object consisting of joints somewhere in it (e.g. door, suspension bridge, turnstile, chain, etc.). **(10 pts)**
- Each garden must have variable height terrain (e.g. ramps, stairs, platforms, mesh terrain, etc.) that your character can move up and down on. **(10 pts)**

- Each garden must have at least two ground materials impacting footstep sounds and particle effects of the character. **(10 pts)**
- Game Feel: Each garden must have a level of polish that improves the game feel of the character and sensation of interacting with a spatial simulation including textured surfaces, skybox, lighting effects, sound effects, etc. **(10 pts)**

Extra Credit

Do one of the options below for **UP TO** 5 points. Do two of the options below for **UP TO** 10 points. Be sure your readme clearly documents that you have completed extra credit and want the grader to assess. Actual credit awarded is determined by the grader, specifically determining if the spirit of the extra credit task is met and the interaction and aesthetic requirements are met.

Collider Animation

Your character's collider should have animated dimensions such that the volume of the collider is optimally sized in comparison to the current animation. This requires that your character have at least one animation that causes an extreme change in shape (e.g. crouching, rolling, crawling, etc.) For instance, a crouching character should possess a smaller (and perhaps wider) collider. Similarly, a jumping character should result in the collider's height and shape changing if the knees tuck in. Probably the best way to implement this is to create a Mecanim animation curve and can be used in a scale calculation of your collider. (Refer to Unity's Mechanim Animation Tutorial.) Your level must provide an environment to test the change in collider size.

Moving Platform

Create a moving platform that the player can stand or walk on. The platform must not slide out from underneath the character as it moves (e.g. the character's feet should stay planted).

Carry Rigid Body Object

Modify your character to animate picking up, carrying, and dropping a small rigid body object. (You can turn off physics for the object while carrying.)

Wall Jump

Add a wall jump animation for your character. Provide an opportunity in your Game Feel garden for the player to exercise the wall jump zigzagging up a narrow chasm.

Slick Shoes!

Modify your character control to support slippery conditions, such as ice. At minimum the character should appear to move his/her feet without getting full traction upon change of speed or direction. You may also consider adding sliding or off balance animations. This will require augmentation of root motion.

Swimming

Modify your character to support swimming in water, demonstrable in one of your Game Feel gardens.

Submission:

You should submit a 7ZIP/ZIP file of your Unity project directory via t-square. **Please clean the project directory to remove unused assets, intermediate and final build files, etc., to minimize the file size and make it easier for the TA to understand.**

The submissions should follow these guidelines:

a) Your name should appear on the HUD of your game when it is running.
b) ZIP file name: <lastName_firstInitial>_mX.zip (X is milestone #)
c) A /build/ directory should contain a build of your game. Please make sure you preserve the data directory that accompanies the EXE (if submitting a Windows build)
d) Readme file should be in the top level directory: < lastName_firstInitial >_mX_readme.txt and should contain the following
    i. Full name, email, and TSquare account name
    ii. Detail which platform your executable build targets (Windows, OSX, GlaDOS, etc.) Also let us know if you implemented game controller support, and which one you used.
    iii. Specify which requirements you have completed, which are incomplete, and which are buggy (be specific)
    iv. Detail any and all resources that were acquired outside of class and what they are being used for (e.g. "Asset Bundles downloaded from the Asset Store for double sided cutout shaders," or "this file was found on the internet has link http://example.com/test and does the orbit camera tracking"). This also includes other students that helped you or that you helped.
    v. Detail any special install instructions the grader will need to be aware of for building and running your code, including specifying whether your developed and tested on Windows or OSX
    vi. Detail exact steps grader should take to demonstrate that your game meets assignment requirements. (E.g. "First, walk towards the pile of blocks and bump into them to knock them down. This should demonstrate actor movement via physically simulated forces and interactivity with environment…")
    vii. Which scene file is the main file that should be opened first in Unity
e) Complete Unity project (any file you acquired externally should be attributed with the appropriate source information)

Submission total: (**up to 20 points deducted** by grader if submission doesn't meet submission format requirements)

Be sure to save a copy of the Unity project in the state that you submitted, in case we have any problems with grading (such as forgetting to submit a file we need).