

Brief explanation of the game

The game is an endless runner in which the terrain is randomly generated, either to the top or left of the current tile. In addition to the randomly generated tiles, a powerup may appear at random. There are five power-ups, increase speed, decrease speed, as well as +5, +10 and +50 points. The more valuable a powerup is, the less likely it is to appear. Users can pick from four different ball types in the options and they must avoid falling into the water at all costs! Finally, the speed increases as the user gains points, making the game harder as the user progresses.

How to use it (especially the user interface aspects)

A user can only navigate through the user interface using their keyboard by pressing the appropriate letters stated. In order to move the ball, a user can either use the arrow keys or the W, A, S, D keys. All of the keys perform their expected function, with the exception of the S or down arrow key. This key stops movement, which may be useful in the earlier stages of the game, but may work against you as the speed increases. Power-up models and their effects are displayed in the instructions tab, along with basic instructions. Additionally, a user can pick between 4 different particle effect types for the ball in the options menu. These options include, no particle effect, fire trailing, smoke and 'aura.' Should a user wish to exit the game, they must press the Escape button.

How you modelled objects and entities

As none of us had any experience with 3D modelling, the geometry was kept fairly simple. The tiles were simple cubes (as they would be anyway), the player was a sphere, and all the powerups were spheres (with computer-drawn textures applied to differentiate them).

As mentioned later in evaluation section, the point-increasing power-ups were changed from spheres to gems to more easily tell them apart from regular power-ups. This was done by importing a free package from the Unity Asset Store.

Fire and Smoke effects are made by creating a new particle system with Box shape, gravity modifier to 1 and start speed to 5 to create floating particles around the object. The fire and smoke particles change size over their lifetime from large, decreasing their size until they disappear. Their colour changes over their lifetime from red to yellow for fire and black to grey for smoke. This is done to enhance effect of the colours. Finally, noise with 3.5 strength is added to cause turbulent motion of particles.

The Ring effect is also made using a particle system. This is done by, making the shape a circle, setting the radius and choosing the "emit from edge" option to keep the particles circling around the ball. The particle's start lifetime is set to 1 to keep the particles near the ring and set shape mode to "loop" to create a looping circle. Then the rate over time is adjusted along with the color over lifetime with a dark green at the front, light green in middle and back to dark green at the end to create the color ring effect.

The menus are created using a Canvas and controlled by a MenuManager script which responds to the player's choices. Finally, a move in and out custom animation is used for the menu transitions.

How you handled the graphics pipeline and camera motion

The camera follows a third-person approach, where it follows the user, with a locked y axis, until the user falls off the platform, at which point it locks the remaining axes. This is done in CameraScript.cs.

Descriptions of how the shaders work

The ripple water works in conjunction with a Collision Script. The collision script calculates wave amplitude, origin and distance from origin based upon where a collision between the water and an object occurs, and the velocity of the colliding object. The amplitude is decreased every frame, while the distance the wave has traveled increases. These calculations are given to the material where the shader then calculates the displacement of the mesh's vertices caused by the collisions (up to 8). There is also a diagonal wave that moves through the water at all times. Not only this, the shader also applies a texture (which is multiplied with a chosen colour) and a bump map. The glossiness and smoothness of the shader can also be customized.

The gem shader does not cull the back side of its model, as it's partially see through. The shader renders the backside of the gem the first pass through, and renders the front side on the second pass.

Description of the querying and observational methods used, including: 1) description of the participants (how many, demographics), description of the methodology (which techniques did you use, what did you have participants do, how did you record the data), and feedback gathered.

Participants Over 50:

Participant 1: Female, technologically inept, no game experience. (Testing Round 1)

Participant 2: Male, Technologically capable, some game experience. (Testing Round 2)

Participant 3: Male, Technologically capable, extensive game experience. (Testing Round 3)

Participants 25 and Under (All Technologically Capable):

Participant 4: Male, extensive gaming experience. (Testing Round 1)

Participant 5: Female, minor gaming experience. (Testing Round 1)

Participant 6: Male, extensive gaming experience. (Testing Round 2)

Participant 7: Male, limited Gaming experience. (Testing Round 2)

Participant 8: Female, extensive gaming experience, Some Programming and design knowledge. (Testing Round 3)

Participant 9: Female, limited gaming experience. (Testing Round 3)

Participant 10: Male, Extensive gaming experience, Computing Major. (Testing Round 3)

Participant 11 Male, minor gaming experience, Computing Major. (Testing Round 3)

Note: gaming experience scale: Extensive > Minor > Limited > No Experience

It is Important to note that various participants had slightly different versions of the game, depending on the testing round. Each round took feedback from the previous rounds and implemented changes reflecting the feedback. These changes are outlined in a later section.

Methodology:

Upon agreeing to trial the game, the participants were presented with the game, not told of the instructions or the goal. This was done to test whether or not the game design appeared intuitive. At this stage, if they would like clarification they could ask as the observational evaluation was done using a co-operative method. Co-operative evaluation as users naturally seemed to be thinking aloud and ask questions while trying the game. This allowed us to answer questions and take notes on future changes as the user played the game. Furthermore, it allowed the user to extensively criticise the game and various interactions, interface and gameplay features.

Upon completing their trial, the users were then queried, using a very informal interview. The questions were customized to how the user reacted to the game, however the first question was always, "Do you have any suggestions?" Another common question was about the clarity of the powerups, questioning whether they understood their purpose or not. The remainder of the questions were based off of comments made during their trial, asking for clarification or elaboration. This method has multiple advantages, particularly in allowing users to fully voice their opinions, while best suiting their experience of the game. However, this method can be quite subjective if the interviewer isn't careful.

Document the changes made to your game based on the information collected during the evaluation.

Post Round 1 Changes

Following the first round of testing, feedback was mostly relating to the difficulty of the game, with users complaining about the game being too difficult. Additionally, users reported not knowing what each of the powerups did. As a result, the starting speed was decreased from 15 to 10, while increasing collision box sizes to allow users more leniency on edges. Furthermore, the materials and shapes of the powerups changed. The Fast power-up went from being a yellow ball, to being a ball with a lightning bolt. The slow power-up went from a grey ball to a black ball with a turtle on it. Whereas the points power-ups went from different coloured squares to utilizing the Unity Gem package from the asset store, thus changing the shape, shader and materials of the powerups.

Post Round 2 Changes

Upon the completion of round 2, the users still complained about it being difficult to know what each power-up does, stating that it was somewhat clear, however the exact intention was unclear. As a result, text was added after picking up a power-up, stating the effect of the power-up. There were additional complaints about difficulty, however they weren't as pronounced as the first round's complaints.

Post Round 3 Changes

Upon the completion of the third round of changes, the feedback received was generally positive, with only two participants able to pass 50 points on their first three attempts. The feedback received was once again that it started off too difficult. This resulted in the starting speed once again dropping from 10 to 9. A speed of 9 was chosen as any slower was found to make the game too slow. However, while speed remained an issue, complaints about the power-ups were non-existent as users could now view their actions in the instructions, but most importantly as text upon picking it up. Finally, we added music to the final game as it was a minor piece of feedback we received from players during our third round, with many stating that 'it felt like something was missing.'

A statement about any code/APIs you have sourced/used from the internet that is not your own

Idea for the game and guide adapted from InScope Studios' Endless Runner guide on Youtube:

https://www.youtube.com/watch?v=gRVS8XJdSOU&list=PLX-uZVK_OK_73EIM5VvzfrBUDqztzbARm

Texture, shader and creating ripple of water learned from guide on Youtube:

<https://www.youtube.com/watch?v=-R7TAVNtYMA>

Texture, Models and shaders for gems used from Unity Asset Store:

<https://www.assetstore.unity3d.com/en/#!/content/3>

Fire and Smoke Particle System inspired by a Youtube tutorial on Trailing effect and got modified by us

<https://www.youtube.com/watch?v=agr-QEsYwD0>

Ring Particle System adapted from a Youtube tutorial video with some modification

<https://www.youtube.com/watch?v=CVsZ98TSEwI>