Pathfinders Test Plan M3

|  |  |
| --- | --- |
| **Test Case** | **Steps & Expected Outcomes** |
| 1. Story | 1. Launch game    * **EXPECT**: The player to be introduced to the story for the game by Oliver and an Old Man. Skip by pressing Z |
| 2. Sound | 1. Launch game    * **EXPECT**: music to play in the background 2. Touch an enemy or fall off the scene    * **EXPECT**: Death sound to play 3. Touch a checkpoint flag  * **EXPECT**: Checkpoint sound to play  1. Touch trophy (end of level)  * **EXPECT**: Level end sound to play  1. Touch paint (extra paint)  * **EXPECT**: Power up sound to play |
| 3. Parallax | 1. Launch game and move player around scene    * **EXPECT**: A new background with trees and clouds, moves with player movement |
| 4. Complex Prescribed Motion | 1. Play the game up to level 2. Enemy entity appears that can shoot projectiles at player using Bezier curve as path   **EXPECT:** Projectile to break upon entity collision. If player gets hit, player dies, and game restarts at start of level |
| 5. Drawing System | 1. Launch game.  * **EXPECT**: Drawing should be functional: player can use it to avoid enemies by blocking line of sight with player. Can also be block paint can from running away |
| 5. Player Movement System | 1. Launch game, move player using wasd keys and test jump height based on length of holding space bar  **EXPECT**: Player jump height to increase the longer the space bar is held |
| 5. Gameplay Progression | 1. Launch game, move player through each level by touching the trophy at the end of each game. Make it through levels 0 and 1 to make it to level 2   * **EXPECT:** boulders and new projectile entities to appear  1. Make it to level 3  * **EXPECT**: Level 3 to have platforms disappear after a short period of time |

Pathfinders Test Plan M2

|  |  |
| --- | --- |
| **Test Case** | **Steps & Expected Outcomes** |
| 1. Game logic response to user input | 1. Launch game and reach the second level by touching the trophy. Observe the falling boulders.    * **EXPECT**: The player should be able to encounter a new level and see the boulders changing path when the player is in sight. 2. Reach the third level by touching the trophy again and try to touch the paint can.    * **EXPECT:** The paint can should move away from the player |
| 2. Sprite sheet animation | 1. Press the D key.    * **EXPECT**: the player stickman character moves to the right with running animations. 2. Repeat 1,2 but with the A key.    * **EXPECT**: Same as 1, but the character faces left while running. |
| 3. New integrated assets | 1. Launch game    * **EXPECT**: A new background with mountains should be visible |
| 4. Mesh-based collision detection | 1. Change the paint.obj mesh with another mesh and check the collisions between the player and the new mesh.    * **EXPECT:** Collisions should fit the expected convex hull for the new mesh. |
| 5. Base user tutorial/help | 1. Launch game.    * **EXPECT**: Basic tutorial and help should be displayed on the screen. |
| 6. FPS counter | 1. Launch game.    * **EXPECT:** The FPS counter should be displayed on the top left of the window bar. |
| 7. Advanced graph search algorithm | 1. Launch game and reach the second level by moving to the trophy in the first level.    * **EXPECT:** Player should be able to encounter a new level. 2. Use the movement key (WASD and space) to move left or the right of newly spawned boulder.    * **EXPECT:** The boulder should search and chase the player based on player’s position |
| 8. Basic drawing on screen | * 1. Launch game.   2. Move the mouse while pressing left mouse button on the screen. * **EXPECT:** Lines should be drawn on the screen following pencil icon. |

Pathfinders Test Plan M1

|  |  |
| --- | --- |
| **Test Case** | **Steps & Expected Outcomes** |
| 1. Textured Geometry Rendering | 1. Launch game    * **EXPECT**: No visual artifacts observed, no object flickering, assets are rendered onscreen. Player is not rendered overlapping with terrain (i.e. ground, walls). 2. Exit game. 3. Launch game again.    * **EXPECT:** Same outcome as 1.), i.e. no inconsistencies. |
| 2. Basic 2D Transformations & Keyboard/Mouse control | 1. Press the D key.    * **EXPECT**: the player stickman character moves (translation) right so long as the key is pressed. 2. Let go of the D key.    * **EXPECT**: the player stickman character begins to slow down to a stop due to friction. 3. Repeat 1,2 but with the A key.    * **EXPECT**: Same as 1,2 but moving left. |
| 3. Random/coded Action & Key-frame/State Interpolation | 1. Launch game and wait for ~10 seconds.    * **EXPECT**: A boulder sprite spawns from the top of the screen and begins to fall downward.    * **EXPECT:** The boulder should be moving slightly horizontally towards the player’s position at the time the boulder was spawned      + i.e., horizontal motion is interpolated between spawn position and player position at spawn |
| 4. Gravity | 1. Launch game.    * **EXPECT:** Random boulders are falling with consistent motion.    * **EXPECT:** Player character starts above platform and slowly drops due to gravity 2. Press the SPACE key to jump.    1. **EXPECT**: The player should accelerate upward momentarily, then come back down. 3. Move off platform.    * **EXPECT:** Player begins to fall continuously (see below). |
| 5. Game-space boundaries | 1. Use the movement keys (WASD) to move off the platform into empty space    * **EXPECT**: Player falls continuously until out of scene, then “dies”, then the game restarts. |
| 6. Collision | 1. Launch game.    * **EXPECT:** The player stickman character drops onto the platform, is stopped, and stays grounded 2. Use the movement keys (WASD) to move into a falling boulder.    * **EXPECT:** Collision occurs, the player becomes uncontrollable (dies), and the scene fades. |
| 7. Save/Load | 1. Press ‘L’ on the keyboard, making sure $PROJ\_DIR/save.json does not exist    * **EXPECT:** Nothing should happen 2. Move to a flag checkpoint in the level 3. Restart the game (‘R’), then press 'L' on the keyboard    * **EXPECT**: Player position is reset to the previous checkpoint position 4. Exit the game, re-launch, press 'L'    * **EXPECT**: saved state persists and is loaded, player position resets to most recently-visited checkpoint |