

`finalBossAttack()`

- Checks if the final boss attack actually damages the monument's health by the right amount
- This makes sure that the final boss attack works for each difficulty by comparing the monument health to the original monument health - boss attack damage. If they are the same it works.
- This is important to the implementation of the game because the witch should damage the monument by the right amount for each difficulty.

`cannon1UpgradesBalance()`

- Checks if the upgrade function subtracts the player balance after a cannon1 (lowest level cannon) has been upgraded for all difficulties.
- This makes sure that the set balance part of the upgrade function works for cannon1 (lowest level cannon) for all difficulties by comparing the player balance after an upgrade to a dummy player's balance - upgrade cost. If they are the same it works.
- This is important to the implementation of the game because the player balance should be subtracted correctly every time a cannon1 (lowest level cannon) is upgraded for each difficulty.

`cannon2UpgradesBalance()`

- Checks if the upgrade function subtracts the player balance after a cannon2 (middle level cannon) has been upgraded for all difficulties.
- This makes sure that the set balance part of the upgrade function works for cannon2 (middle level cannon) for all difficulties by comparing the player balance after an upgrade to a dummy player's balance - upgrade cost. If they are the same it works.
- This is important to the implementation of the game because the player balance should be subtracted correctly every time a cannon2 (middle level cannon) is upgraded for each difficulty.

`cannon3UpgradesBalance()`

- Checks if the upgrade function subtracts the player balance after a cannon3 (highest level cannon) has been upgraded for all difficulties.

- This makes sure that the set balance part of the upgrade function works for cannon3 (highest level cannon) for all difficulties by comparing the player balance after an upgrade to a dummy player's balance - upgrade cost. If they are the same it works.
- This is important to the implementation of the game because the player balance should be subtracted correctly every time a cannon3 (highest level cannon) is upgraded for each difficulty.

cannon1UpgradesAttackDamage()

- Checks if the upgrade function correctly updates the attack damage after a cannon1 (lowest level cannon) has been upgraded for all difficulties.
- This makes sure that the attack damage part of the upgrade function works for cannon1 (lowest level cannon) for all difficulties by comparing the cannon's attack damage after an upgrade to a dummy cannon's attack damage * upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack damage should be incremented correctly every time a cannon1 (lowest level cannon) is upgraded for each difficulty.

cannon2UpgradesAttackDamage()

- Checks if the upgrade function correctly updates the attack damage after a cannon2 (middle level cannon) has been upgraded for all difficulties.
- This makes sure that the attack damage part of the upgrade function works for cannon2 (middle level cannon) for all difficulties by comparing the cannon's attack damage after an upgrade to a dummy cannon's attack damage * upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack damage should be incremented correctly every time a cannon2 (middle level cannon) is upgraded for each difficulty.

cannon3UpgradesAttackDamage()

- Checks if the upgrade function correctly updates the attack damage after a cannon3 (highest level cannon) has been upgraded for all difficulties.

- This makes sure that the attack damage part of the upgrade function works for cannon3 (highest level cannon) for all difficulties by comparing the cannon's attack damage after an upgrade to a dummy cannon's attack damage * upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack damage should be incremented correctly every time a cannon3 (highest level cannon) is upgraded for each difficulty.

cannon1UpgradesAttackSpeed()

- Checks if the upgrade function correctly updates the attack speed after a cannon1 (lowest level cannon) has been upgraded for all difficulties.
- This makes sure that the attack speed part of the upgrade function works for cannon1 (lowest level cannon) for all difficulties by comparing the cannon's attack speed after an upgrade to a dummy cannon's attack speed / upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack speed should be updated correctly every time a cannon1 (lowest level cannon) is upgraded for each difficulty.

cannon2UpgradesAttackSpeed()

- Checks if the upgrade function correctly updates the attack speed after a cannon2 (middle level cannon) has been upgraded for all difficulties.
- This makes sure that the attack speed part of the upgrade function works for cannon2 (middle level cannon) for all difficulties by comparing the cannon's attack speed after an upgrade to a dummy cannon's attack speed / upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack speed should be updated correctly every time a cannon2 (middle level cannon) is upgraded for each difficulty.

cannon3UpgradesAttackSpeed()

- Checks if the upgrade function correctly updates the attack speed after a cannon3 (highest level cannon) has been upgraded for all difficulties.

- This makes sure that the attack speed part of the upgrade function works for cannon3 (highest level cannon) for all difficulties by comparing the cannon's attack speed after an upgrade to a dummy cannon's attack speed / upgrade multiplier. If they are the same it works.
- This is important to the implementation of the game because the cannon's attack speed should be updated correctly every time a cannon3 (highest level cannon) is upgraded for each difficulty.