

Testing Plan:

1. Each person writes tests first for each ticket/method they have been assigned (unit) then implements the method,
2. Then as each method is finished, test the class as a whole seeing how all methods interact with each other (integration). At this stage mock tests and mock objects will be used to reduce dependency on other team members.
3. Then each member writes system tests to see how their part interacts with the dungeon controller (system). We will also be testing the game by playing it on the frontend as we add new features.

NewGameTest (System test)

- Test it can read json file
- Test the game can start normally
- Test exceptions when config / dungeon files not exist

ControllerTest (System Test)

- Test the map can return modal normally

BuildTest (System Test)

- Test player cannot build item with exception - the buildable type is not bow / shield
- Test player cannot build item with exception – not enough material
- Test player successfully builds shield with 2 woods + 1 treasure
- Test player successfully builds shield with 2 woods + 1 key
- Test player successfully builds bow with 3 arrows + 1 wood
- Test player successfully builds 2 bows

TickItemUsedTest (System test)

- Test can consume a potion
- Test can consume multiple potions
- Test potion lasts one tick and is then changed
- Test exception when the string is not bomb / potion
- Test exception when the id is not exist

TickMovementTest (Integration test)

- Test player's basic movement
- Test player can not open the door with wrong key
- Test player blocked by a wall
- Test player can collect item but cannot collect 2 keys
- Test player can push boulder

Doors And Keys Test (Unit Test)

- Test the player cannot collect 2 keys at a time
- Test the player cannot open the door with wrong key
- Test the player cannot open the door without key
- Test the player can open the door normally

GoalTest (Integration Test)

- Test different combinations of AND or OR operators
 - Supergoal: OR
 - Two Ands subgoals
 - One Or and one AND subgoal
 - Simple goal and list
 - Supergoal: And
 - Two ORs subgoals
 - One Or and one AND subgoal
 - Simple goal and list
- Test disjunction operators (4 ORs)
- Test the player cannot exit without doing any other tasks
- Test the player can undone the goal by pushing the boulders away

PortalTest (Unit Test)

- Test the player can teleport normally
- Test the player cannot teleport if they end up on a wall
- Test the player can only teleport to corresponding colour

MercenaryTests (Unit + Integration Test)

- Test mercenary moves towards player
- Test bribing mercenary
- Test bribe fails because of not enough gold but within radius
- Test bribe fails because not within radius
- Test mercenary is able to move around walls
- Test mercenary is not able to move past a boulder
- Test mercenary is not able to move past a zombie toast spawner
- Test mercenary runs away from player when player is invincible
- Test mercenary movement when it is an ally

PlayerMovementTests (Integration Test)

- Test the player can move up / down / left / right normally
- Test the player can collect item normally
- Test the player can collect item on switch
- Test player can move into multiple enemies

BoulderTest (Integration + Unit Test)

- Test the player can push the boulder normally
- Test the player can only push one boulder at a time
- Test the player can push the boulder onto item
- Test the player can trigger the switch by pushing a boulder onto it
- Test the player cannot push the boulder to the wall / a locked door
- Test when the player push the boulder away, it should untrigger the switch

BattleAllyTest (Integration + System Test)

- Test battle with no weapons and one ally bonus
- Test battle with no weapons and multiple ally bonus
- Test battle with shield, bow, sword, ally bonus
- Test player bribes merc then battle
- Test the calculation of the ally battle

BattlePotionsTest (Integration + System Test)

- Test battle with invincible potion zombie
- Test battle with invisible potion zombie
- Test battle with invincible potion all enemies
- Test battle with invisible potion all enemies
- Test battle with invincibility potion reach duration mid battle
- Test battle with invisibility potion reach duration
- Test battle after invisibility potion
- Test the player becomes invincible when battle
- Test the player becomes invisible when battle
- Test using different potions in a battle

SpiderTests (Unit Test)

- Test movement sequence is correct assuming no collisions
- Test movement sequence when boulder blocks path
- Test movement through static entities

ZombieToast Test (Unit + Integration Test)

- Test movement changes position value for zombie toast
- Test collision with boulders
- Test collision for locked doors
- Test collision for unlocked doors
- Test collision for walls
- Test collision for portals
- Test collision for zombie toast spawner
- Test zombie runs away from player when player is invincible

ZombieSpawner Test (Unit + Integration Test)

- Test correct number of zombies spawned for the spawn rate
- Test found open square
- Test no open square
- Test no open square boulders and walls
- Test pushing boulder can spawn now
- Test if player is not cardinally adjacent to the spawner when trying to destroy it
- Test if player does not have a weapon but is within range and attempts to destroy a spawner
- Test player can destroy spawner when cardinally adjacent and has a weapon

Bomb test (Integration Test)

- Test given id not in player's inventory
- Test given id not correspond to a bomb
- Test bomb successfully explode with radius 3
- Test inactive switch and trigger bomb when activate
- Test bomb won't affect inventory
- Test bomb explodes with radius 0
- Test bomb explodes with radius 1

BattleTest (Integration + Unit Test)

- Test battle with no weapons vs zombie and zombie dies
- Test battle with multiple enemies
- Test battle with no weapons vs zombie and player dies
- Test battle with no weapons vs spider and spider dies")
- Test battle with no weapons vs mercenary and mercenary dies
- Test battle with sword
- Test battle with sword durability maxed during battle
- Test battle with bow
- Test battle with Shield
- Test battle with multiple bows
- Test battle with multiple shields
- Test battle with shield, bow, sword
- Test player battle spider and spider dies
- Test player battle zombie and player dies
- Test player battle zombie with sword
- Test player battle zombie with bow
- Test player battle zombie with shield
- Test player battle with bow, shield, sword
- Test player battle with bow, shield, sword, with varying durability
- Test player battle multiple enemies

Player and Enemy Test (Integration + Unit Test)

- Test player can move into enemy in unlocked doorway
- Test player can move into an enemy and collect item
- Test player can move into an enemy and collect item reverse
- Test player can move into multiple enemies and collect item
- Test player can drink potion
- Test potion effect wears off
- Test potion effects can be queued

Player Interaction Test (Integration + Unit Test)

- Test player can collect them
- Test player can collect multiple treasures
- Test player can collect item on a switch
- Test player can collect item on a switch reverse

Milestone 3 Tests

Zombie Spawner extra tests (Unit + integration)

- Test cannot spawn into wall with spider on it
- Test cannot spawn into boulder on top of switch
- Test can spawn open square up
- Test can spawn open square down
- Test can spawn open square left
- Test can spawn open square right

Swamp Tile Tests (unit + integration tests)

- Test mercenary moves towards player through swamp tile
- Test mercenary takes shortest path to player with swamp tile
- Test zombie movement when stuck in swamp tile
- Test spider movement sequence is correct for swamp tiles
- Test hydra movement when stuck in swamp tile
- Test assassin moves towards player through swamp tile
- Test assassin takes shortest path to player with swamp tile

Mercenary Extra Movement Tests (unit + integration tests)

- Test mercenary takes shortest path to player
- Test mercenary takes shortest path to player (left) when moving around walls
- Test mercenary takes shortest path to player (right) when moving around walls

Assassin Tests (Integration + unit test)

- Test assassin moves towards player
- Test bribe fails because not enough gold but within radius
- Test bribe fails because not within radius
- Test assassin is able to move around a boulder

- Test assassin is able to move around a zombie toast spawner
- Test assassin takes shortest path to player
- Test assassin takes shortest path to player (left) when moving around walls
- Test assassin takes shortest path to player (right) when moving around walls
- Test assassin moves towards from player when player is invisible and in range
- Test treasure used to bribe is removed from players inventory even when failed
- Test treasure used to bribe is removed from players inventory
- Test assassin moves towards player through swamp tile
- Test assassin takes shortest path to player with swamp tile
-

Hydra Tests (unit + integration tests)

- Test movement changes position value
- Test collision for boulders
- Test collision for locked doors
- Test collision for walls
- Test collision for portals
- Test collision for zombie toast spawner
- Test hydra runs away from player when invincible
- Test stuck in swamp tile
- Test battle with hydra 100% health rate
- Test battle with hydra 0% health rate
- Test battle with hydra 20% health rate
- Test battle with hydra 50% health rate
- Test battle with hydra 80% health rate

Midnight Armour Test (unit + integration tests)

- Test battle unit test
- Test build armour with sword and stone
- Test build invalid exception insufficient items
- Test build invalid exception zombies exist
- Test provides extra attack damage and armour

Persistence Test (integration + system test)

- Test can save game
- Test LoadGame with exception
- Test saving and loading a simple maze map
- Test saving and loading a map with boulders
- Test saving and loading a map with enemies and collectables
- Test saving on a map containing all types of entities

Rewind Test (integration test + system tests)

- Test rewind tick less than one
- Test tick not occurred

- Test tick occurred
- Test inventory persists
- Test old player exists when time travelled
- TestOld Player Have Same Path AsBefore
- Test Bomb Game State
- Test Door & Key Game State
- Test Time Travel to initial state
- Test Time travel one tick
- Test Game state after tick occurs
- Test Older player reached from map
- TestOld Player disappear at right tick

Sceptre Test (unit + integration)

- Test build sceptre with wood, key, stone
- Test build sceptre with wood, treasure, stone
- Test build sceptre with arrow, key, stone
- Test build sceptre with arrow, treasure, stone
- Test build sceptre with arrow, stone, stone
- Test build sceptre with wood, stone, stone
- Test invalid exception 2 arrows , treasure
- Test player cannot build item with exception
- Test mind control mercenary
- Test mind control assassin
- Test mind control assassin then battle while under control

SunStone (Unit + integration + acceptance)

- Test pick up sunstone
- Test open door with sunstone and retain sunstone
- Test use a sunstone to open door rather than a key
- Test can build a shield with sunstone
- Test build shield with key + sunstone
- Test sunstone cannot be used to bribe mercenary
- Test sunstone counts towards treasure goal
- Test the mercenary will still follow the player
- Test sunstone interactions running frontend

Time Travelling Via Portal Tests (integration tests)

- Test initial game state
- test time travel after 30 ticks
- Test older player moves into portal
- Test older player disappears immediately

Time Travelling Portal Test (unit tests)

- Test Portal Exists

- Test Portal Teleports

Time Turner Test (unit tests)

- Test collect time turner
- Test door state
- Test activate switch
- Test explode bomb
- Test rewind after explode bomb

Logic Switches (Unit + Integration)

- Test boulder triggers a switch
- Test can activate a light bulb
- Test light bulb can be lit up by placing a boulder on an cardinally adjacent switch
- Test xor light bulb can be lit up by placing a boulder on an cardinally adjacent switch
- Test logic AND light bulb can be lit up by adjacent switches
- Test logical switch can be activated by the boulder regardless of logic
- Test can logic switch can be activated by the electrically
- Test all the other entities cardinally adjacent to the wire are activated
- Test Switch Door can be opened by the wire
- Test Switch Door can be opened by the key
- Test Switch Door can be opened by the active switch

Generate Dungeon Test (integration tests)

- Test IllegalArgumentException with not exist config file
- Test the basic game generation
- Test the map can be generated with negative coordinate
- Test all non-wall grids are reachable
- Test all wall grids (boundary) are not reachable