Multiplayer Platform Game
Group 19

Martin Petterson
Oskar Kvist
Christoffer Ekeroth
Misael Berrios Salas

February 25, 2008

Contents

1 Introduction

2 System Overview

3 Design Considerations

4 Graphical User Interface

5 Design Details

6 Functional Test Cases

Since almost all of the functional requirements in our software have to do with graphical representations of the game state and user interaction we will not use regular automated assertion tests. Rather, all tests will need to be performed by a tester who monitors and interacts with the game in order to assert that the test is successful. Our source code will include a preprocessor debug switch—activating the switch and compiling and running the game will result in starting a special debug version of the game.

The debug version of the game has all the features of the regular game, but with a few changes an additional features:

- The regular game stages are not available in the debug version. Instead, a selection of game stages constructed especially for debugging purposes are made available to the tester.
- The debug version skips all menus, instead starting directly on a default debug game stage. The tester can change between the available debug game stages by pressing buttons on the keyboard or the Xbox 360 gamepad.
- By default two players are instantiated, both of which can be controlled by the tester.
- The debug version is designed to be able to run on both the PC and the Xbox 360 console. The game shall run in windowed mode on the PC.
- The debug version can receive input from the keyboard and Xbox 360 gamepads. Several players can be controlled by the same keyboard, but two Xbox 360 controllers are needed in order to control both players.
- A player can be given additional lives at any time by pushing a button on the keyboard or the Xbox 360 gamepad

- A player can switch the contents of her first power-up slot between all available power-ups by pushing a button on the keyboard or the Xbox 360 gamepad, giving the tester unlimited access to all power-ups
- The debug game stage can be reinitialized at any time by pushing a button on the keyboard or the Xbox 360 gamepad
- The debug version displays the average time it took to render the last couple of frames. The amount of frames to be averaged is defined as a constant.

The following test cases will be distributed across several different debug game stages, designed to make testing convenient.

- **Functionality being tested**: That platforms are unaffected by gravity.
  Functional requirement: 2(b)ii.
  Expected output: The platforms remain suspended in the air.
  Procedure: Verify that the platforms remain suspended in the air.
- **Functionality being tested**: That players or monsters can't pass through platforms in any way.
  Functional requirement: 2(b)i
  Expected output: The players and monster(s) don't pass through the platforms in any way.
  Procedure: Verify that the player or monster(s) do not fall through the platforms. Try to run through a wall and verify that the player doesn't pass through it. Verify that the monster does not move through the platforms. Shoot the other player or the monster with the Boxing Glove power-up and verify that the player or monster does not pass through platforms when hit by the Boxing Glove power-up.
- **Functionality being tested**: That moving platforms move along a path correctly.
  Functional requirement: 2(b)iv
  Expected output: The moving platform moves along its path.
  Procedure: Verify that the platform moves along its path.
- **Functionality being tested**: That players or monsters can not pass through moving platforms in any way.
  Functional requirement: 2(b)i
  Expected output: The player ad monster do not pass through the moving platform in any way.
  Procedure: Jump onto the moving platform. Verify that the player does not fall through the moving platform. Try to run through a wall on the moving platform. Verify that the player cannot run through the moving platform.

- **Functionality being tested**: Gravitation in the game
  Functional requirement: 2(c)
  Expected output: Players and monsters fall towards the bottom of the screen when in midair.
  Procedure: Let the player walk off the platform and observe if the player starts falling. Observe wheter the monster falls down or not.

- **Functionality being tested**: The power-up dispenser shall cycle through available power-ups
  Functional requirement: 2(d)i.
  Expected output: The power-up dispenser cycles through all available power-ups, both in respect to the icons displayed and the power-up the player gets when touching the dispenser.
  Procedure: Observe the power-up dispenser and check if it cycles through all the power-up icons.

- **Functionality being tested**: The player shall gain the power-up that is currently shown on the power-up dispenser
  Functional requirement: 2(d)ii.
  Expected output: The player gets the power-up that is shown on the power-up dispenser when she touches the power-up dispenser
  Procedure: For each power-up displayed on the power-up dispenser, touch the power-up dispenser and confirm that you got the power-up corresponding to the power-up icon that was currently shown.

- **Functionality being tested**: The player shall only be able to use the same power-up dispenser once
  Functional requirement: 2(d)iii.
  Expected output: The player gets a power-up from the power-up dispenser the first time she touches it and no power-ups from touching it subsequent times.
  Procedure: Touch the power-up dispenser several times and confirm that the player got only one power-up.

- **Functionality being tested**: Traps shall cause the player to lose life when touching them.
  Functional requirement: 2(f)i.
  Expected output: The player loses a life when touching the trap.
  Procedure: Touch a trap and confirm that a life was subtracted from the player's life total.

- **Functionality being tested**: Traps shall be able to turn themselves on and off.
  Functional requirement: 2(f)ii.

Expected output: The player does not lose life when touching a trap in its off state.
Procedure: Touch a trap in its off state and confirm that no life was subtracted from the player's life total.

- **Functionality being tested**: Traps shall either move along a set path or have a fixed location
  Functional requirement: 2(f)iii.
  Expected output: Traps that move move along their paths correctly, traps that are not supposed to move stay stationary
  Procedure: Observe moving and non-moving traps and confirm that they move in the manner intended.

- **Functionality being tested**: Players shall be able to avoid traps either by jumping over them, waiting or going around them.
  Functional requirement: 2(f)iv.
  Expected output: A player passes through the game stage without losing life from touching traps.
  Procedure: Play each game stage and confirm that it would be possible to reach the finish point without losing life by means of traps.

- **Functionality being tested**: Monsters move back and forth on platforms.
  Functional requirement: 2(g)i.A.
  Expected output: The monster moves back and forth, turning around when reaching an obstruction or the end of the platform.
  Procedure: Observe the monster and confirm that it changes direction when reaching an obstruction or the end of the platform.

- **Functionality being tested**: Players shall be able to stun monsters.
  Functional requirement: 2(f)iii.
  Expected output: The monster gets stunned. The monster stops for the duration of the stun and the player does not lose life as a result of touching the stuenned monster.
  Procedure: Fire a Boxing Glove at the monster and confirm that it stops moving and that the visual stun effect is displayed. Touch the monster and confirm that this does not cause the player to lose life.

- **Functionality being tested**: Players should not be able to pass through monsters and monsters should not be able to pass through other monsters.

Functional requirement: 2(f)iv.A.
Expected output: Players and monsters do not pass through each other.
Procedure: Observe two monsters moving towards each other and confirm that they change direction when they reach each other. Try running through a monster and confirm that this is not possible. Shoot a Boxing Glove at a monster and confirm that the monster does not move through the other monster or a player when moved by the Boxing Glove.

- **Functionality being tested**: Losing life when running into monsters.
  Functional requirement: 2(f)iv.B.
  Expected output: The player loses a life when touching the monster.
  Procedure: Run into a monster and confirm that a life was subtracted from the player's life total.

- **Functionality being tested**: Jumping on monsters.
  Functional requirement: 2(f)iv.C.
  Expected output: The player loses a life when landing on the monster or the monster gets stunned, depending on the type of monster.
  Procedure: Jump on a monster. If it is a type of monster that the player can jump on without losing life, confirm that the player did not lose life and that the monster was stunned. If the monster was a type of monster that causes players to lose life when jumping on them, confirm that a life was subtracted from the player's life total.

- **Functionality being tested**: Losing life from monsters located above the player.
  Functional requirement: 2(f)ivD.
  Expected output: The player loses a life when touching the monster.
  Procedure: Let a monster higher up than the player touch the player and confirm that a life was subtracted from the player's life total.

- **Functionality being tested**: Using power-ups on monsters.
  Functional requirement: 2(f)v.
  Expected output: Power-ups affect the monsters in the same way they would a player.
  Procedure: Using the same procedure as when testing power-ups on

a player, test all power-ups on a monster and confirm that they have the same effect as they would have on a player.

- **Functionality being tested**: Players shall be able to avoid monsters either by jumping over them, waiting or going around them.
  Functional requirement: 2(f)vi.
  Expected output: A player passes through the game stage without losing life from touching monsters.
  Procedure: Play each game stage and confirm that it would be possible to reach the finish point without losing life by means of monsters.

- **Functionality being tested**: Vertical and horizontal movement shall be independent
  Functional requirement: 3(a)i.
  Expected output: The horizontal speed is the same before the jump, during the jump and after the jump if the horizontal input is constant and the player is not accelerating.

  Acceleration while standing still is the same on a platform as on the air.
  Procedure: Verify that the same horizontal speed is maintained while running, jumping and landing without letting go off the forward button on the game pad.

  Verify that the acceleration while standing still and then starting to run is equivalent to the acceleration while standing still, jumping and within the jump starting to run.

- **Functionality being tested**: 2 players can not occupy the same space
  Functional requirement: 3(a)ii.
  Expected output: The player trying to occupy the other players space will be pushed back in the same way as colliding with a wall.
  Procedure: Verify that when running/walking/jumping into another player, the two players do not overlap.

- **Functionality being tested**: When a player does not receive any horizontal input it starts decreasing its horizontal speed until the player stands still.
  Functional requirement: 3(a)iii.
  Expected output: The horizontal speed is decreased to 0 when no horizontal inputs are received from the player.
  Procedure: While in a horizontal movement, for example running, let go of the game pad and stop sending inputs.

- **Functionality being tested**: A player is always able to change his horizontal direction except while stunned.
  Functional requirement: 3(a)iv.
  Expected output: The horizontal direction is always changeable except when stunned.
  Procedure: While running/walking/standing still/falling/jumping try to change the horizontal direction of the player, verify that it is possible.

  Get in a stunned position, for example let player 2 jump on your head, and try to change the horizontal direction of the player, verify that it is not possible.

- **Functionality being tested**: Player being affected by gravity.
  Functional requirement: 3(a)v.
  Expected output: Since the gravity power affects the players downwards a player who walks over a edge should fall downwards until hitting a new platform or disappearing from the screen.

  A player who is affected by less gravity should be able to jump higher than a player with normal effect by gravitation.
  Procedure: Verify that the players starts falling after walking over an edge.

  Verify that there is a difference in the height of a jump from a player who is using the levitation power-up, who decreases the gravity, and a player who has no power-up activated.

- **Functionality being tested**: Maximum running speed.
  Functional requirement: 3(b)i.
  Expected output: When running a maximum speed is obtained. This speed may not be exceeded by any player, except when a player is using the speed boost power-up and a new maximum speed is set for this player.
  Procedure: Make the player run, and verify that after a little while, the player stop accelerating.

- **Functionality being tested**: Players accelerate and decelerate.
  Functional requirement: 3(b)ii.
  Expected output: When standing still and starting to run the speed shall increase constantly until reaching the maximum speed.

  When running and changing horizontal direction the the speed should constantly decrease in the first direction and then in the in the same way as above increase in the new direction.
  Procedure: Stand still and start moving in any horizontal direction, verify that the speed increases smooth and constant.

While running, change direction and verify that the speed in the previous direction starts decreasing down to 0 and then starts increasing to the new direction until reaching the maximum velocity.

- **Functionality being tested**: Player must be on a platform to be able to jump.
  Functional requirement: 3(c)i.
  Expected output: If a jump input is received the player will only gain new upward velocity if standing on a platform.
  Procedure: Walk off an edge and while falling press the jump button on the game pad and verify that nothing changes.

  While being in the air, press the jump button on the game pad again and verify that nothing changes.

- **Functionality being tested**: Player receives instantly a vertical upward velocity when a jump input is received. The velocity is affected by the gravity and therefore constantly reduced.
  Functional requirement: 3(c)ii.
  Expected output: When a jump input is received the player will jump, the velocity upwards will diminish until becoming negative making the player fall.
  Procedure: While moving or standing still push the jump button on the game pad and verify that it instantly affects the vertical upwards speed of the player. Also verify that the players vertical speed starts decreasing due to the gravity.

- **Functionality being tested**: Player shall be able to use a power-up from the inventory in any moment, except while stunned.
  Functional requirement: 3(e).
  Expected output: While equipped with a power-up it should be activated when the user pushed on the proper game pad button.
  Procedure: Pick a power-up and try using it while standing still, jumping, running, falling, using another power-up and while stunned. Verify that the power-up is activated in all situations except where the player is stunned, in this case nothing shall change.

- **Functionality being tested**: Whenever a player loses a life it shall become invulnerable for a short time, in that time she may not lose any live.
  Functional requirement: 3(f)i and 3(f)ii.
  Expected output: When a player is invulnerable nothing except falling out of the screen may make her lose a life.
  Procedure: 2 players needed, the first one shall go to the far end of the screen and collide with a monster, making her invulnerable for a short period of time. The second player should make the camera

follow him and leave the first player out of the camera scope. Verify that the first player loses a life when she is stunned and ends up out of the camera scope.

2 players needed, the first one shall go to the far end of the screen and collide with a monster, making her invulnerable for a short period of time. While the first player still is invulnerable the second player shall try to jump on the first players head or use power-ups on her and verify that nothing changes.

- **Functionality being tested**: When a player is invulnerable or stunned it should be visible for the bare eye.
  Functional requirement: 3(f)iii and 3(g)iv.
  Expected output: Invulnerability and stunned status is somehow visible.
  Procedure: A player runs into a monster and becomes stunned and there after invulnerable, verify that is is visible that the player is stunned and invulnerable.

- **Functionality being tested**: While stunned a player is unable to affect her own movements.
  Functional requirement: 3(g)i and 3(g)iii.
  Expected output: When stunned all player inputs are ignored.
  Procedure: Take a player and collide with a monster, when stunned try to jump, move or use a power-up, verify that it is not possible.

- **Functionality being tested**: While stunned a player starts decelerating.
  Functional requirement: 3(g)ii.
  Expected output: When a player has a speed and is stunned it should start losing speed.
  Procedure: While a player is running jump on him, making him stunned and verify that he still keeps moving at the same direction as before the collision but that the speed starts decreasing.

- **Functionality being tested**: Player should be able to stun each other by jumping on each others head.
  Functional requirement: 3(h).
  Expected output: When a player jumps upon another players head the target becomes stunned.
  Procedure: Jump on a players head and verify that she becomes stunned.

- **Functionality being tested**: If a player lands on another player or a monster she shall automatically jump.
  Functional requirement: 3(i).

Expected output: When landing on a opponent or monster the player will automatically jump without receiving any inputs.
Procedure: Jump on a player or a monster that gets stunned when jumped on, verify that the player makes a small jump after landing.

- **Functionality being tested**: If a player ends up outside the screen, she shall lose a life and then respawn. **Functional requirement**: 3(j).
  Expected output: The player will respawn and lose a life when ending up outside of the screen.
  Procedure: Move one player towards the right end of the screen and verify that the camera starts following her. Continue doing so until the other player ends up outside the screen. Verify that this causes the player that ended up outside the screen to lose a life. Verify that the player that ended up outside the screen respawns.

- **Functionality being tested**: A player shall be able to re spawn near the middle of the screen.
  Functional requirement: 3(k).
  Expected output: When a player falls down from a endless pit or ends up outside the screen she shall re spawn somewhere in the middle of the screen.
  Procedure: Fall down a endless pit having the camera in different positions and verify that you re spawn somewhere near the middle of the screen when possible.

- **Functionality being tested**: That the boxing glove power-up travels horizontally in the direction the player is facing, that the player/monster it hits gets a horizontal speed in the same direction and that the glove disappears when it hits an object.
  Functional requirement: 4(d)i.
  Expected output: The glove travels horizontally in the direction the player is facing, when it hist a monster/ player it gives them a horizontal speed in the traveling direction and then disappear, if it hits a wall it shall only disappear.
  Procedure: Spawn a player with a boxing glove power-up and shoot it at a monster/player/wall and very that the expected output happens.

- **Functionality being tested**: That the fetter power-up placed at the player using it and that it is affected by gravity.
  Functional requirement: 4(d)v.A and 4(d)v.B.
  Expected output: The fetter is spawned at the position of the player and is affected by gravity.

Procedure: Spawn a player with a fetter power-up. Jump into the air and use the power-up while in the air. Verify that the fetter is spawned at the position of the player and that it falls to the ground.

- **Functionality being tested**: That the speed boost power-up makes the player run faster.
  Functional requirement: 4(d)iii.
  Expected output: The player can run faster, but the time to accelerate to maximum speed, and the time it takes to stop is the same as before.
  Procedure: Spawn two players, one with a speed boost. Use the speed boost and start running with both players, verifying that the expected output is true.

- **Functionality being tested**: Test the fetters positioning while applied and how it affects the gravity and movement speed.

  Functional requirement: 4(d)v.A and 4(d)v.E.

  Expected output: The fetter when applied, shall be placed at the position of the player it is applied on.

  A player/monster with a fetter applied on her has a greater gravity force affecting her and a smaller maximum speed.
  Procedure: Spawn a fetter and a player. Make the player collide with the fetter and verify that the fetter is applied and that it is positioned on the player.

  When the fetter is applied try jumping with the player and verify that the hight of the jump is less that without the fetter. Also try running and verify that the maximum speed is now less.

  Make a monster collide with a fetter and verify that the speed of the monster is less when the fetter is applied.

- **Functionality being tested**: Fetter is affected by gravity.
  Functional requirement: 4(d)v.B.
  Expected output: When dropping a fetter from the air it shall fall.
  Procedure: Spawn a player equiped with the fetter power-up. Jump with the player and in the air activate the fetter power-up. Verify that the fetter falls down to the ground when dropped.

- **Functionality being tested**: The fetter shall not catch the player who activated it.
  Functional requirement: 4(d)v.D.
  Expected output: When colliding with a fetter you yourself have activated nothing should happen.

Procedure: Spawn a player with the fetter power-up. Activate the fetter and collide with it. Verify that nothing changes.

- **Functionality being tested**: The shrinking ray
  Functional requirement: 4(d)vi
  Expected output: The shrinking ray travels horizontally, disappears when it hits a player/monster/wall and shrinks players/monsters. Players that are shrunk have a slower maximum speed, and shorter jumping height.
  Procedure: Spawn a player with a shrinking ray, shoot it at a wall and verify the expected output.

  Spawn a player with a shrinking ray, shoot it at a monster and verify the expected output.

  Spawn two players, one with a shrinking ray. With one of the players, shoot the other player with the shrinking ray, walk/jump around with the other player and verify the expected output.

- **Functionality being tested**: Banana peels
  Functional requirement: 4(d)vii.A, 4(d)vii.B and 4(d)vii.D.
  Expected output: When activating the banana peel power-up 3 small banana peels will appear near the player who activated it. If the player is in the air the peels will fall down due to gravity. When the player who activated the power-up steps on the peels nothing happens.
  Procedure: Spawn a player with the banana peel power-up. Jump with the player and activate the power-up while in the air. Verify that three banana peels appear and are positioned near the player. Verify that the peels fall down to the ground. Also verify that when stepping on the peels nothing changes.

- **Functionality being tested**: Banana peels stunning power.
  Functional requirement: 4(d)vii.C.
  Expected output: When a player or a monster trips on a banana peel they will become stunned and have a lower acceleration and deceleration.
  Procedure: Spawn 2 players and a monster. Equip one player with a banana peel power-up. Activate the power-up and make the player and the monster trip on the banana peels and verify the expected output.

- **Functionality being tested**: That the amount of lives a player has is presented on the screen.
  Functional requirement: 5(a).
  Expected output: The right amount of lives a player has is presented on the screen.

Procedure: Walk straight into the monster and verify that you now have one life less than before.

Give the player an extra life and verify that you now have one more life.

- **Functionality being tested**: That a players power-ups are shown on the screen.
  Functional requirement: 5(a), 4(a), 4(b) and 4(c).
  Expected output: The right power-ups shown on the screen.
  Procedure: Take two power-ups from the power-up dispenser and verify that they are both shown. Use both power-ups.

  Do this procedure for all power-ups in the game, and verify that the icons for all the power-ups are right.

- **Functionality being tested**: That players occupy the same screen.
  Functional requirement: 5(b)
  Expected output: That the screen is not split up.
  Procedure: Spawn another player and verify that both players are on the same screen.

- **Functionality being tested**: That the camera doesn't zoom in/out.
  Functional requirement: 5(c)
  Expected output: The camera doesn't zoom.
  Procedure: Spawn another player, and run with one of the players to the right and verify that the camera doesn't zoom.

- **Functionality being tested**: That the camera follows the player in the lead, but gives a buffer of 1/3 of screen.
  Functional requirement: 5(d)
  Expected output: The camera moves only if the leader moves, and it keeps a distance of 1/3 of the screen as a buffer.
  Procedure: Spawn another player, and run with one of the players to the right. Verify that the camera follows the player in the lead, and that it keeps a buffer of 1/3 of the screen to the right.

- **Functionality being tested**: That a player dies if he gets outside the screen.
  Functional requirement: 5(e)
  Expected output: The player dies when he is outside the screen.
  Procedure: Spawn another player, and run with one of the players to the right until the other player ends up outside the screen. Verify that the other player looses a life and respawns.

- **Functionality being tested**: That the game stops if all players have finished/died.

Functional requirement: 6(a)
Expected output: The game ends and the end game screen is presented.
Procedure: Spawn another player and jump into the finishing portal with both of them. Validate that the end game screen is presented. Do the same thing again, but let one of the players die after the other player has finished.

- **Functionality being tested**: That players start with 3 lives.
  Functional requirement: 6(b) and system requirement 6(a).
  Expected output: It is presented in the GUI that the player has 3 lives.
  Procedure: Validate that the player starts with 3 lives.

- **Functionality being tested**: That a player dies if he doesn't have any lives left.
  Functional requirement: 6(b) and system requirement 6(a).
  Expected output: The player dies after losing a life 3 times.
  Procedure: Walk into a monster 3 times and validate that the player has died, and forfeited the race.

- **Functionality being tested**: That if only one player is left, he wins the race.
  Functional requirement: 6(c)
  Expected output: The only player left wins.
  Procedure: Spawn another player. Walk into a monster 3 times with one player and validate that the race has ended, and that the player left has won.

- **Functionality being tested**: That the game has music and that it is at least 2 minutes long.
  Functional requirement: 7(a)
  Expected output: The player dies after loosing a life 3 times.
  Procedure: Walk into a monster 3 times and validate that the player has died, and forfeited the race.

- **Functionality being tested**: That there are sound effects in the game.
  Functional requirement: 7(b)
  Expected output: There are sound effects for jumping, losing a life, reaching the goal, using a power-up, landing on another player, landing on a platform.
  Procedure: Verify that there is a sound effect after these procedures:
  - Jump with the player
  - Walk into a monster

  - Take a power-up from a power-up dispenser, and use it
  - Jump and land on a platform
  - Spawn another player and jump on him
  - Jump into the finishing portal

- **Functionality being tested**: Two players cannot select the same character
  Functional requirement: 8(a)
  Expected output: After one player has chosen a character, another player cannot choose the same character.
  Procedure: Start the game with two gamepads. Choose the first character with the first game-pad and validate that the same character cannot be choses with the other gamepad.

- **Functionality being tested**: There are at least 3 game stages to choose from.
  Functional requirement: 8(b)
  Expected output: In the game stage selecting screen, you are able to select at least 3 different game stages.
  Procedure: Start the game with two gamepads, choose characters for both players and validate that there are at least 3 different game stages that are selectable.

- **Functionality being tested**: Any player can pause the game.
  Functional requirement: 9(a)
  Expected output: The game is paused
  Procedure: Start a race with two players. Try pausing and unpausing the game with both players.

- **Functionality being tested**: Only the player who has paused can unpause.
  Functional requirement: 9(b)
  Expected output: Only the player who has paused can unpause.
  Procedure: Start a race with two players. Try pausing with one player and unpausing the game with the other players.

- **Functionality being tested**: That the pause-screen contains "Resume", "Restart" and "Abort".
  Functional requirement: 9(c)
  Expected output: The pause-screen contains "Resume", "Restart" and "Abort".
  Procedure: Start a race with two players. Pause the game and validate that "Resume", "Restart" and "Abort" are all options.

- **Functionality being tested**: That there is a end game screen which shows finishing places.

Functional requirement: 10(a)
Expected output: You can see an end game screen with finishing places.
Procedure: Start a race with two players. Finish the race with both players and validate that there is an end game screen with finishing places.