**Super Toast Boy**

**Introduction**

Super Toast Boy is a platformer based heavily on Super Meat Boy. The goal is to guide Toast Boy through each level by carefully, and quickly, avoid traps, pitfalls, and enemies. To aid Toast Boy through this, he can make use of various power-ups such as jelly or butter, to get through certain obstacles.

**Features**

Toast Boy can walk, run, slide on walls, or jump from them to get higher or further. He also has the following power-ups available:

* Toaster
  + Toast Boy becomes extra crispy and can break through walls (still in development)
* Jelly
  + By covering himself in jelly, Toast Boy sticks to walls and ceiling with almost no sliding to get to hard to reach areas (still in development)
* Butter
  + Sliding across the floor on butter makes Toast Boy much faster and able to slide over sticky floors but he has to be careful not to slide off the edge of platforms.

**Object Organization**

Levels are comprised of a few standard pieces:

* Background
  + Some kind of background sprite to give the level some atmosphere.
* Toast Boy
  + Toast Boy contains a standard sprite renderer, an animator, and a collision box.
* Platforms
  + Each platform has a sprite and a basic collider to handle floor and ceiling collisions.
  + Nested in each platform is two smaller box colliders for detecting wall collisions so Toast Boy can slide or jump.
* Enemies
  + Chatter Mouth
    - A chatter mouth looks like a set of teeth chomping away and is tagged as an enemy for easy enemy collision checks.

**Scene Organization**

Scenes are currently broken down between levels and menus. Levels are named using the convention “Level\_N” where N is the index of the level in the lineup. When starting a level, Toast Boy is told which level it is then when he reaches the goal, we move to “Level\_N+1”.

At this point the game does not contain a pause menu but one will be added in the future.

**Mechanics Implementation Details**

*Buttons and flags are indicated using [[Button]] or [[Flag]]. Actual mappings may vary.*

* Toast Boy
  + Walking (on ground)
    - When the player is holding [[Left]], apply a negative horizontal acceleration.
    - When the player is holding [[Right]], apply a positive horizontal acceleration.
    - When the player is not holding [[Left]] or [[Right]], stop the player.
    - If the players velocity exceeds the maximum walking speed, set the velocity to the maximum speed.
    - Running can be activated at any time.
  + Walking (in air)
    - When the player is holding [[Left]], apply a negative horizontal acceleration.
    - When the player is holding [[Right]], apply a positive horizontal acceleration.
    - If the players velocity exceeds the maximum walking speed, set the velocity to the maximum speed.
    - Running can be activated at any time.
  + Running (on ground)
    - When the player is holding [[Left]], apply an amplified negative horizontal acceleration.
    - When the player is holding [[Right]], apply an amplified positive horizontal acceleration.
    - When the player is not holding [[Left]] or [[Right]], stop the player.
    - If the players velocity exceeds the maximum running speed, set the velocity to the maximum speed.
    - Running can be disabled and return to normal walking at any time.
  + Turning Around
    - When the player quickly shifts from holding [[Left]] to [[Right]], or vice-versa, apply an additional acceleration constant until the direction of the players velocity has changed.
  + Jumping
    - When the player presses [[Jump]], apply an upward force.
    - If the player releases [[Jump]] while still moving upward, set their vertical velocity to 0.
  + Wall Sliding
    - When the player collides with a wall in the air, prevent the player from apply a horizontal force for a limited amount of time.
    - Prevent player from dislodging from the wall unless they apply a force away from the wall or when they collide with the floor.
    - Lower gravity to allow player to slide further up wall when jumping and slide more slowly when moving down.
  + Wall Jumping
    - If the player is sliding against a wall and presses [[Jump]], apply a force both away from the wall and upward to allow the player to scale walls with repeated jumps.
* Enemies/Trap
  + Collision
    - When any enemy or trap collides with Toast Boy, reset Toast Boy to his original spawn position.
* Power-Ups
  + Jelly
    - Change Toast Boy's sprite to show him covered in jelly.
    - Make Toast Boy set his vertical velocity to 0 when a wall slide begins.
    - Set the gravity to a number close to 0 while still sliding.
  + Toaster
    - Change Toast Boy's sprite to make him appear he just came out of a toaster.
    - Set a flag on Toast Boy stating that when he collides with certain walls, those walls will break.
  + Butter
    - Change Toast Boy's sprite to show him covered in jelly.
    - Remove the instant stop when the player releases [[Left]] or [[Right]]
    - Increase the player's maximum speeds and acceleration.
    - Set a flag so that Toast Boy won't be affected by sticky surfaces.
* Platforms
  + State Flags
    - A platform with a [[Sticky]] flag will reduce the player's maximum speed, acceleration, and jump speed.
    - A platform with a [[Enemy]] flag will kill the player on contact.
  + Collision
    - When the player collides from above, stop the player's vertical velocity to allow the player to walk or run.
    - When the player collides from below, set the player's vertical velocity to 0 and let them fall.
    - When the player collides from the side, treat the platform as a Wall
* Walls
  + State Flags
    - A wall with a [[Sticky]] flag will reduce the player's wall slide gravity, slow the upward velocity, and slow their jump speed.
    - A wall with a [[Enemy]] flag will kill the player on contact.
  + Collision
    - When a player collides from the side, enter into a wall slide.
    - When a player collides from the top or bottom, treat the wall as a Platform.
* Goal
  + Collision
    - When the player collides with the goal, start the next level.

**Software Design**

* Player Controls
  + Controls are velocity based to provide a smooth physics experience and tight controls
  + Pressing [[Left]] or [[Right]] increases the horizontal velocity
  + Pressing [[Jump]] sets the vertical velocity to a set, instant value to launch the player
  + Wall jumping assigns a set vertical velocity and a set horizontal velocity, based on what side of the player the wall is on.
  + The player has 3 main states:
    - On the ground
      * This is determined by doing a linecast below the player to see if they're colliding with any ground.
    - On a wall
      * This is determined by doing a linecast to either side of the player and seeing if they're colliding with a wall.
    - In the air
      * This is true if both of the previous checks are not true.
* Terrain Collisions
  + Collisions are handled by Unity's internal collision detection system.
  + However, all ground and walls are on the “Ground” layer so we can perform a linecast from the edge of the player out to a small distance to determine if the player is on the ground or colliding with a wall.
* Resettable Objects
  + Resettable objects are objects which will need to be reset if the player dies, power-ups being an obvious example.
  + All resettable objects must be placed in a parent object matching the name such as “PowerUps” or “Enemies”
  + When the player dies, find all of those parent objects by name then re-activate each of their children. For resetting their original position, an additional ResetPosition component should be on the object. Objects which are not moved do not require this component.

**Changes Based on Playtesting**

After playtesting, we found that most everyone agreed on two things:

1. The controls feel floaty.
2. The levels are too hard.

* Levels were redone
  + Reduce the *opposition* by simplifying the opening levels.
  + Improved the players *decision making* by making at least 2 ways to complete most levels.
  + Improved the *variety of encounter* by introducing different challenges in each level to teach a new concept. Enemies aren't even introduced until later into the game.
  + We teach the player that their *position is their identity*.
* Controls were reworked entirely
  + The controls provide a far better *simulation* of being this character in this world.
  + Improved controls really drive home that *the player is their position.*