**Work in Progress Report 1**

**CHAUHUNEN**

**Github Link:** [**https://github.com/d7chau/Dino-Jump.git**](https://github.com/d7chau/Dino-Jump.git)

**Major Developments/Breakthroughs (reference specific code please):**

**-Sprite Movement**

if (Gdx.input.isKeyPressed(Input.Keys.RIGHT)) {

nYDinoX += nSpriteSpeed;

if (nYDinoWidth > 0) {

nYDinoWidth \*= -1;

nYDinoX -= nYDinoWidth;

}

}

if (Gdx.input.isKeyPressed(Input.Keys.LEFT)) {

nYDinoX -= nSpriteSpeed;

if (nYDinoWidth < 0) {

nYDinoWidth \*= -1;

nYDinoX -= nYDinoWidth;

}

}

**-Screen Wrap**

if (nYDinoX > Gdx.graphics.getWidth() + 50) {

nYDinoX = -50;

}

if (nYDinoX < -50) {

nYDinoX = Gdx.graphics.getWidth() + 50;

}

**-Gravity**

public void HandleFalling() {}

if (bCanFall) {

nYDinoY -= dFallSpeed;

dFallSpeed += dGravity;

}

if (nYDinoY <= 100) {

bCanJump = true;

bCanFall = false;

dFallSpeed = 0;

}

}

public void HandleJumping() {}

if (bCanJump) {

nCountJump++;

nYDinoY += dJumpSpeed;

dJumpSpeed -= dGravity;

if (nCountJump >= 40) {

bCanJump = false;

bCanFall = true;

dJumpSpeed = 20;

nCountJump = 0;

}

}

}

**-Spawning in platforms**

int[] arnNewPlatforms = new int[5];

for (int i = 0; i < arnNewPlatforms.length; i++) {

Random random = new Random();

int nPlatformRNG = random.nextInt(Gdx.graphics.getWidth() - 150);

arnNewPlatforms[i] = nPlatformRNG;

}

**-Hit Detection**

boolean isOverlapping = rectDino.overlaps(arnRectPlatform[i]);

if (isOverlapping) {

bCanJump = true;

bCanFall = false;

dFallSpeed = 0;

}

**Major Setbacks/Challenges (reference specific code please):**

-Having platforms spawn at the top of the screen and move toward the bottom of the screen (v2.1 Platforms)

-When sprite facing right and near a platform edge, it will fall through the platform as if the platform were not there

**Modifications to Specifications/Release Schedule:**

|  |  |
| --- | --- |
| **Release Name** | **New incremental features of this release** |
| v1.0  Sprite | Controlling the sprite so that it can move left and right with user input via left and right arrow keys. |
| **v.1.01**  **SpriteAppearance** | Sprite faces right when right arrow key is pressed, and horizontally flips and faces left when left arrow key is pressed (**added release**) |
| **v1.1**  **ScreenWrap** | Screen Wrap (if the sprite exits one side of the screen it will enter the opposite side) (**switched spots with release Gravity**) |
| **v1.2**  **Gravity** | Creating gravity in the game so that when the sprite jumps on a platform, they will go up, and eventually down (**switched spots with release ScreenWrap**) |
| **v2.0**  **HitDetection** | Generating platforms and when sprite is moving down onto platform it bounces up but when it is moving upwards through a platform, the sprite will pass through (**switched spots with release Platforms**) |
| **v2.1**  **Platforms** | Having platforms spawn at top of screen and move toward bottom of screen each time a sprite bounces on it (**switched spots with release HitDetection**) |
| v2.2  SpringsTrampolines | Springs and Trampolines (if sprite is in contact with the top of a spring or trampoline, it will propel the sprite in the y axis at a height higher than the normal bounce) |
| v3.0  Score | Tracking the score of the user (increases by a certain increment as the user continues to bounce on platforms upwards) |
| v4.0  RetryOrQuitScreen | Program allows user to choose to click the home button and go to the main menu, or to click the retry button to run the game once again |
| v4.1  MainMenu | Main menu (contains buttons to play screen, instruction screen, and customize sprite screen) |
| v4.2  InstructionScreen | Instruction Screen that teaches users how to control the sprite, and informs users about the rules and conditions of the game |
| v4.3  CustomizeScreen | Allows user to choose the colour of the sprite that they are playing with (Shop) |

**Description of scratch/test program**

**Sources:**

<https://youtu.be/T1aN--vTqLc>

**Describe the code and the lesson that you learned from it:**

-Collision detection

-Learned how to detect collisions between two transparent rectangles (rectangles are constantly in the same position as the sprites and have the same size as them acting as hitboxes)

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

-N/A

**Description of scratch/test program**

**Sources:**

<https://www.youtube.com/watch?v=fCsMiiup1OQ>

**Describe the code and the lesson that you learned from it:**

-Gravity

-Learned how to make the sprite jump and fall

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

-Modified code greatly as initial code from the tutorial did not work

**Peer Assesment:** 50/50