Buse Çarık

Assignment Features:

1. I implement virtual joystick, on screen gamepad and accelerometer according to the article in the student portal. I create a start screen. In that screen, player can choose which type of controller that he/she wants. When he/she presses the start button, game starts. In the start screen, I use three radio buttons. As a default, I choose virtual joystick without a reason. When the checked button is changed, I store the chosen one's value in a string and send it to the GameActivity class with intent. According to value that come from the intent, I create the controller.
2. I create a new static enemy class that is called Enemy. It is extended from StaticEntity class. To detect who is colliding, I add a new feature to the Entity class which is _type. In the player class, when the player collides with something, I put a condition to detect if it collides with enemy. If the player hits to the enemy, I check the time. If the player completes the recovery time from last hit, its health is decreased, update the timer for the detect the recovery frame, remove the enemy from the screen and make a sound effect. For the recovery frames, I create a long timer object which is set to the current time of the device. In the onCollision() method, first I check the last time of crash. If the current time is bigger than the last crash time plus recovery frame time, then it is not in the recovery frame so it can collide. I update the hit time and do the other operations. During the recovery frames, do not enter the collision case. For the effect of the recovery frames, I change the alpha value in the paint object. I create a boolean variable which is _collision to detect if the game is in the recovery frames. In the collision situation, I update the value of the collision and I create a new method which is checkTimer(), to check if the recovery time is finished to change the value of collision. I check the current time and the time last crash and value of the collision for the first case. If the time for recovery is finished, it updates the value of the collision variable. Also, to change the alpha value according to recovery process, I create another method which is setAlphaValue(). First, it calls the checkTimer() method, then according to the value of the collision, it updates the alpha value on paint object. And in the render() method of the player, I call the setAlphaValue(). If the player's health is less than 1, then the it restarts the level. In the Game class, I check the player's health, if it is less than 1, make the game over sound and call the restart method. In the restart method, make the start over sound effect and call the LevelManager's restart method. In that method, first reset the number of collected coins and reload the map for that level. I store the level number in the LevelManager class with _levelNumber variable. When the level is changed, this value is also updated. Also, when the player hits an enemy, I add a jump effect for the physics reaction. When it hits, in the update method of the player, it calls hurt function, and change the y velocity and add to the y coordinate of the player.
3. I create a new class for dynamic collectible entity which is called Coin. When the player hits a coin, its number of collected coins is increased. In the level manager, I

create variables that keep the number of collectibles. When the map is loading and entities are created, I count the total number of in that level. I also have two more variables for the coins; one of them for the collected coins for that level and the other one is for collected coins in the all game. When the player hits a coin, in the Coin's onCollision method, if the coin hits to the player, it increases the number of coins that for both coin variables, make a sound effect and add a physic with jumping effect. I add y velocity and in the update method of the coin, I check the collision case and if the coin collides, it adds y value until some point and then remove the coin from entity list. I show the collected coins and remaining coins on the screen with Render class and values are changed when the player picks a coin, or it dies. When the player passes the next level, he/she does not lose collected coins, they are also transferred to the next level, that's why I create the levelCollectedCoin and collectedCoin. If the player dies, collectedCoin is reset. Otherwise, it continues to count collected coins in each level. levelCollectedCoin is for determining the remaining coin in that level. In the update method of level manager, I update the remaining coin with a small math operation. I subtract the levelCollectedCoin from the totalCoin in that level.

4. I create a new RenderHud class to display the number of collectibles, player' health and level number. I create a paint object. I add a new font resource. To use that I used the Typeface class. I get the font from assets and set to the paint object. To arrange the text's size, align and colour, I use the paint object in the constructor of the RenderHUD class. To draw the text to the screen, I create a method which isContinue(). In that method, I take the player's health, collected coin, remaining coin and level number as parameters also the canvas. With canvas' drawText method I draw them to the screen. I refactor all the user-facing Strings into string resource. I get the value as a parameter in the string resource. (%1$d with that way in the resource file). I call that method in the Game class' render method, if the game is not over, it calls that method.
5. I replace the maps of the TestLevel with plaintext files. I create 3 plaintext resources for 3 levels. In the level manager, I read those files to the map according to the level number. I create a SparseArray to store the level' files names according to the level number. When it is loading the maps, it opens the file and read according to the level number. I borrow the code of the reading a file from the internet. I use BufferedReader and InputStreamReader. They open the file and fill the map. Then call the loadMapAssets method to get sprite name and create the entities. Also, instead of TestLevel, I use LevelInfo class for sprite name and to get sprite name.
6. I add sound effects for collision, death, boost and start game. I create Jukebox class for that. Mostly I use your code sample. I create a soundpool object. And for representing each situation, I create a new type of enum which is GameEvent. I store the GameEvent and id of sounds in to a HashMap with loadSoundEffects() and loadEventSound() methods. I read ids from assets with AssetFileDescriptor. I create a play() method to play the effects. In the Game class, I create a method to play the sound according to the event. It takes GameEvent as a parameter and call the play() method of the _jukebox object to play the sound. For the background music, I make

changes from the sample code from the student portal. I create an SparseArray for keeping the tracks' file name according to the level number to play different music for each level. loadMusic method takes the level number as a parameter. In that way, it loads the background music according to the level. I create new method in Jukebox class to play music according to level. That method gets the level number, load the music for that level number and start the play the music with musicBgMusic method. This method is called in the Game class. I crate a new method there. When the level is changed, that method is called, first call the pauseBgMusic method to pause the media player, then call the onLevelChangeMusic method to start the new level's background music. I add two buttons to the game screen to turn on or off the sound and music. In the Game class' update method, I check if the buttons are clicked. If they are pressed, call the toggleMusic/SoundStatus methods. I also change the icon on the buttons according to the state.

VG 1. When the player collects all the coins and reaches the door, he/she passes to next level. I create three levels for that game. If the player can finish three levels, the game starts from the beginning. I create a new entity which is door. When player hits to door, it checks if all the coins are collected. If not, nothing happened. I also add new entities for the next levels. I add a new class for health that is called Heart. When player collects the heart, he/she gains health. And I add another enemy for the third level. For the next level, if all conditions are satisfied, door class calls the levelUp method from the Game class. If the player is still alive, increase the level number, add sound effects, and call the level manager to load the next level's map. It also passes the player's health and collected coins to the next level. For each level, I create different plaintext files.

Resources, Assets and Codes That Are Not Mine:

Font

- https://www.dafont.com/theme.font.

Drawables:

- https://opengameart.org/content/platform-pack

Sound effects:

- https://gamesounds.xyz/?dir=Sonniss.com%20-%20GDC%202016%20-%20Game%20Audio%20Bundle/ShapingWaves%20-%20%208-BIT%20GAMES
- https://gamesounds.xyz/?dir=Sonniss.com%20-%20GDC%202018%20-%20Game%20Audio%20Bundle/Gamemaster%20Audio%20-%20Guns%2C%20Bullets%20and%20Explosions
- https://www.bfxr.net/

Background Musics:

- https://gamesounds.xyz/

Codes:

- _typeface = Typeface.createFromAsset(context.getAssets(), "fonts/theme.ttf"); and
- _paint.setTypeface(_typeface); are from https://stackoverflow.com/questions/27588965/how-to-use-custom-font-in-a-project- written-in-android-studio.
- https://stackoverflow.com/questions/9544737/read-file-from-assets
- https://stackoverflow.com/questions/24291721/reading-a-text-file-line-by-line-in-android
- https://stackoverflow.com/questions/4233873/how-do-i-get-extra-data-from-intent-on-android