CPSC386 - Assignment1 – Daniel Wu

Scripts/Components listed and described:

- **Bar.cs** – this script was made togerthe with professor May's help. It controls one specific behavior of how the ball bounce at the paddle. The farther the ball hits the paddle from the center, the higher angle reflection will be added. Example: if the ball hits the side of the paddle, the ball will bounce to the left.
- **BottomWall.cs**: this script had little modification and will end the game. When the ball hits the bottom wall, also known as Pit (invisible to the player), it will load the "gameover" scene. This condition is needed since the ending condition of Endless game is letting the ball hit the pit.
- **BouncingBall.cs**: This customized function will detect collisions with the blocks, and with the pit. When colliding with a block, it calls "EnemyManager.HandleEnemy". When it hits the pit in game2, it will retrieve the score and will check if it is better than the current high score.
- **ClockController.cs**: This script is made by Daniel and will start the clock when the user presses the key. Then, it will display the time on screen similarly to a running time. It will also provide the running time as a float variable to compare it with highest score.
- **EnemyManager.cs**: this customized script handles the enemy spawning. It will destroy the game object when it is not needed (time battle mode) or keep it in a pool when it is expected to be created again (endless mode). At time battle mode, there is a counter that is increased each time a block is eliminated, and when it reaches the maximum enemy's parameter, the game will end. While ending the game, it will retrieve the amount of time since the start of the game and will compare with the high score. If the value is faster them the best time, it will update the register. At Endless mode, one enemy will be generated per second until the limit of 10 enemies. A counter will be increased each time the player eliminates a block, and the value will be updated on screen. This script also has a timer that controls how fast it spawns blocks, a current spawned variable that controls how many were created and a maximum spawn variable that will keep enemies on screen under this limit. All those three variables work together to keep enemies under control.
- **Gameover.cs**: Daniel's script will show a message when the player gets a better high score at "gameover" scene.
- **GlobalVariables.cs**: Daniel's script will save some information that is needed between scenes, like to mute the sound, and if a high score is obtained (to display a message that highscore was updated).
- **Highscore_bt.cs**: Daniel's script will reset the high score file and update the information on screen.
- **Highscore_init.cs**: Daniel's script will create two parameters on PlayerPrefs on main screen if they do not exist. One is to store the best time in Time Battle mode, and another is to store the maximum blocks eliminated in Endless mode.
- **Highscore.cs**: Daniel's script will load the best results from both playing modes.
- **InputExample.cs**: this customized script will allow an initial impulse only when the ball is not moving. It also keeps the ball over the paddle when the ball's speed is zero, even if the player moves the paddle. It also set a random trajectory when giving the initial impulse. It also keeps the paddle between the walls.

- **SceneLoader.cs**: This customized script will load a scene that is previously set by its name or number. It also detects if the next scene is a game (1 or 2) and will reset the variable of high score obtained in case it is a game. Lastly, when it is a game, it will load the game mode selected as “LoadSceneMode.Single” and the baseLv scene as “LoadSceneMode.Additive”. Game1 and game2 contains their own rules and game objects while baselv scene contains static elements that are common on both games mode, like the top, left, right and bottom walls, and the back button.
- **SoundController.cs**: Daniel’s script will mute the music if the global parameter is set to true.
- **SoundTxtSwitcher.cs**: Daniel’s script will switch the text of a button in MainMenu scene between “sounds on” and “sounds off” when it is pressed.
- **UnitPool.cs**: This script had no modification and will create new blocks based on enemyPrefab and will keep a pool of it. It will disable game objects when it is not needed and activate it again when it is needed. It can also destroy it when the game object is not needed anymore.

GameObject composition detailed (what personal components make it “work”)

- **bgMusic**: SoundController.cs (Daniel)
- **PlayerInput**: ClockController.cs (Daniel), InputExample.cs (customized), Player Input (default)
- **Dynamic Sprite – bar**: Bar.cs (Daniel + professor)
- **Dynamic Sprite – ball**: BouncingBall.cs (customized)
- **EnemyManager**: EnemyManager.cs (customized)
- **Button-Clear**: Highscore_bt.cs (Daniel)
- **HSUpdate**: Gameover.cs (Daniel)
- **HighscoreController**: Highscore.cs (Daniel)
- **Button-play, Button-play2**: SceneLoader (customized)
- **Button-sounds**: SoundTxtSwitcher.cs (Daniel)
- **highscore_init**: Highscore_init.cs (Daniel)

Brief description of each scene

- **BaseLv** – This scene provides static objects that will be used in all games (Time Battle and Endless mode). Game objects like the four walls and the back button will be loaded in this scene.
  - *Static Sprite – wallR, Static Sprite – wallL, Static Sprite – wallR*: they are static objects that bounce back the ball and keep the ball/paddle inside the visible are of screen.
  - *Static Sprite – wallB*: this object is not visible and is the pit. When the ball touches this object, the gameover scene is loaded. Player could not directly interact with this object.
  - *Button-Back*: when pressed, this button will load the mainMenu scene.

- **Credits** – A simple scene which give credits to the creator of this game and plays background music.

- *Credits*: displays a text with no direct interaction.
- *Button-Back*: this button will load the mainMenu scene.
- *bgMusic*: plays music but no direct interaction. Could be muted and unmuted at MainMenu scene.
- *bgImage*: displays background image but no direct interaction.

- **Game1** – A Time Battle mode game with background music. There are 24 blocks which must be destroyed. A time counter will register how much time is needed to finish it. The game ends when all blocks are eliminated or when the ball falls into the pit. The back button will return to MainMenu scene. This scene must be loaded together with baseLv scene.
  - *PlayerInput*: only some keys are recognized (left, right, a, d, and spacebar). Mouse is used to navigate and click the buttons.
  - *Dynamic Sprite – bar:* It could move left and right, limited to the space between the walls.
  - *Dynamic Sprite – ball:* It could bounce between the walls, bounce when hit a block, and bounce differently depending on where it hits the paddle. The game always ends when the ball falls into the pit (and hit the bottom invisible wall). The ball starts moving when the player press “Spacebar”. It should move together with the paddle when the ball is not moving and independently after the initial movement.
  - *Text-clear:* This text is invisible until all blocks are eliminated. Game is paused when this message is displayed.
  - *Text-HSUpdated:* this text is invisible until the player get a score/time better than the high score. It only appears when the game ends.
  - *Text-counter:* this text displays the time since the game starts and should change every second. It should only start counting when the player presses any key.
  - *Text-description:* this is a static description with no interaction at all.
  - *EnemyManager:* This game object will log when a block is destroyed, monitor when the game ends at this scene, pause the game when it ends, and displays the “Level Clear” message. It also compares the player’s time with highscore, update it if it is lower, and display “Highscore updated” message.
  - *Button-Back*: same as Credits.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.

- **Game2** – A Endless mode game with background music. Enemies will continuously spawn every second until the limit of 10 blocks on screen. The game ends when the ball falls into the pit. The back button will return to MainMenu scene. This scene must be loaded together with baseLv scene.
  - *PlayerInput*: same as game1.
  - *Dynamic Sprite – bar:* same as game1.
  - *Dynamic Sprite – ball:* same as game1.
  - *Text-counter:* this text displays the score (enemies eliminated) and should be updated each time a block is destroyed.
  - *Text-description:* this is a static description with no interaction at all.
  - *EnemyManager:* This game object will notify when a block is released at the log. It controls the enemy respawn as described before, using pool to store it and deactivating an object instead of destroying it.
  - *Button-Back*: same as Credits.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.

- **GameOver** – a simple scene with a background music that shows when the ball falls into the pit.
  - *Credits:* this is a static description with no interaction at all.
  - *Text-HSUpdated: same as game1*
  - *Button-Back*: same as Credits.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.
  - *HSUpdate: No interaction with the player, but it will allow Gameover Scene to display the "highscore updated" message.*

- **Highscore** – this scene will load the best time and best score stored in PlayerPrefs with a background music. It also allows the user to reset the values to default.
  - *Text_TimeBattle, Text_Endless:* this is a static description with no interaction at all.
  - *Results_TimeBattle, Results_Endless:* these values are updated from values stored at PlayerPref. It is the best time and best score achieved in the game.
  - *Button-Back*: same as Credits.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.
  - *HighscoreController:* No interaction with the player, but it will load values stored at PlayerPrefs and update Results_TimeBattle, Results_Endless values.

- **Instructions** – a simple scene with instructions about how to play the game with a background music.
  - *Text-Left, Text-Right, Text-Spacebar:* No interaction with the player, but it will display static texts.
  - *Image-Left, Image-Right, Image-Spacebar:* No interaction with the player, but it will display static images.
  - *Text-Objective:* No interaction with the player, but it will display static text.
  - *Button-Back*: same as Credits.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.

- **MainMenu** – this scene will allow the player to click buttons and navigate through the app and plays a background music.
  - *Button-Instructions*: press to load instruction scene.
  - *Button-play*: press to load game1 + baselv Scene.
  - *Button-play2*: press to load game2 + baselv Scene.
  - *Button-sounds*: press to change button text and to mute/unmute music.
  - *Button-Credits*: press to load credits scene.
  - *Button-HighScore*: press to load highscore scene.
  - *Text-Title*: just display the game title but no direct interaction.
  - *bgMusic*: same as Credits.
  - *bgImage*: same as Credits.
  - *highscore_init:* a highscore message will display only when player achieve a better result. Message will be displayed at game over or level clear screen.

Source Code for all scripts included at end or included as an archive file i.e. zip (only scripts are required. Sending the assets is appropriate, but **points will be deducted** if the entire project's files are submitted)