Biscuit Blob™

Created by:

Julia Wozniak - Bucks ID 22117252
Bartłomiej Wojda - Bucks ID 22204382
Sam Eldridge - Bucks ID 22117253

Introduction to the game

The storyline of the game is the blob trying to find his biscuit, with many obstructions along the way. The main character is a green blob, who is trying to make his way to the portal, to get to the biscuit. When the biscuit is obtained, the game is won. If the player runs out of time or touches the enemy (without the sword) the game will be lost. The game can be played again in a loop if the player simply presses the ‘play again’ button.

Game Instructions

The game consists of two main levels. The objective is for the blob to get the biscuit, and to do that he has to get through the levels. In order for the portal to the next level to open, the player must collect all the jewels. In the first level, the player must collect a potion to receive a jump boost, and this way the player can collect all of the jewels. The clocks are for extra time in the game, they add 10 seconds to the game time. In top left of the screen, there is a time and score count. Score count is measured with the number of jewels picked up by the player. Once the time runs out, the player has lost the game. In the second level, the player can pick up the sword to kill enemies, which is not an option in the first level. Once the player has gone through the portal into the last screen and collects a biscuit, the game is over, and the YOU WON screen is displayed, with a choice to play again.

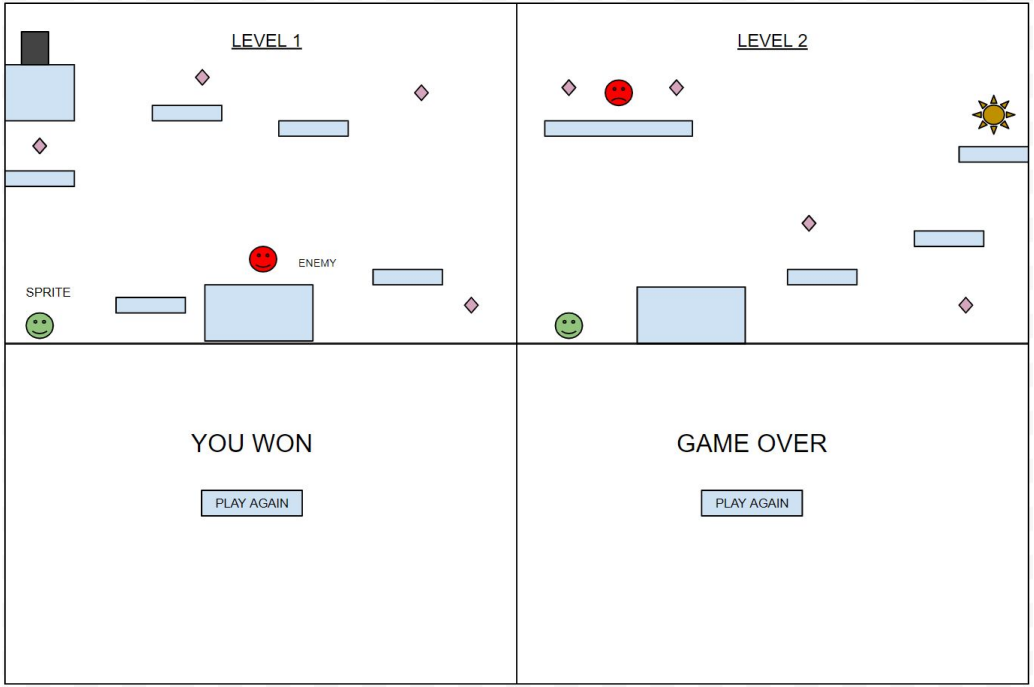
Platform

It was decided by all of the group members to use Greenfoot software, which is a game development environment. It uses Java language which has been previously used by all members, and they all feel comfortable with it. Github was also used in this project as a platform for all of the students participating in this project to share their work with others. There are three contributors for the GitHub, and all of the changes made are recorded on the platform with names and description if added by the contributor.

The GitHub repository link :
https://github.com/CoffeeAddicts/GreenFoot-Game

Initial Design

Rough idea for levels, level 1 & 2, the ‘win’ and ‘game over’ screen, with the button to play the game again.



Graphics

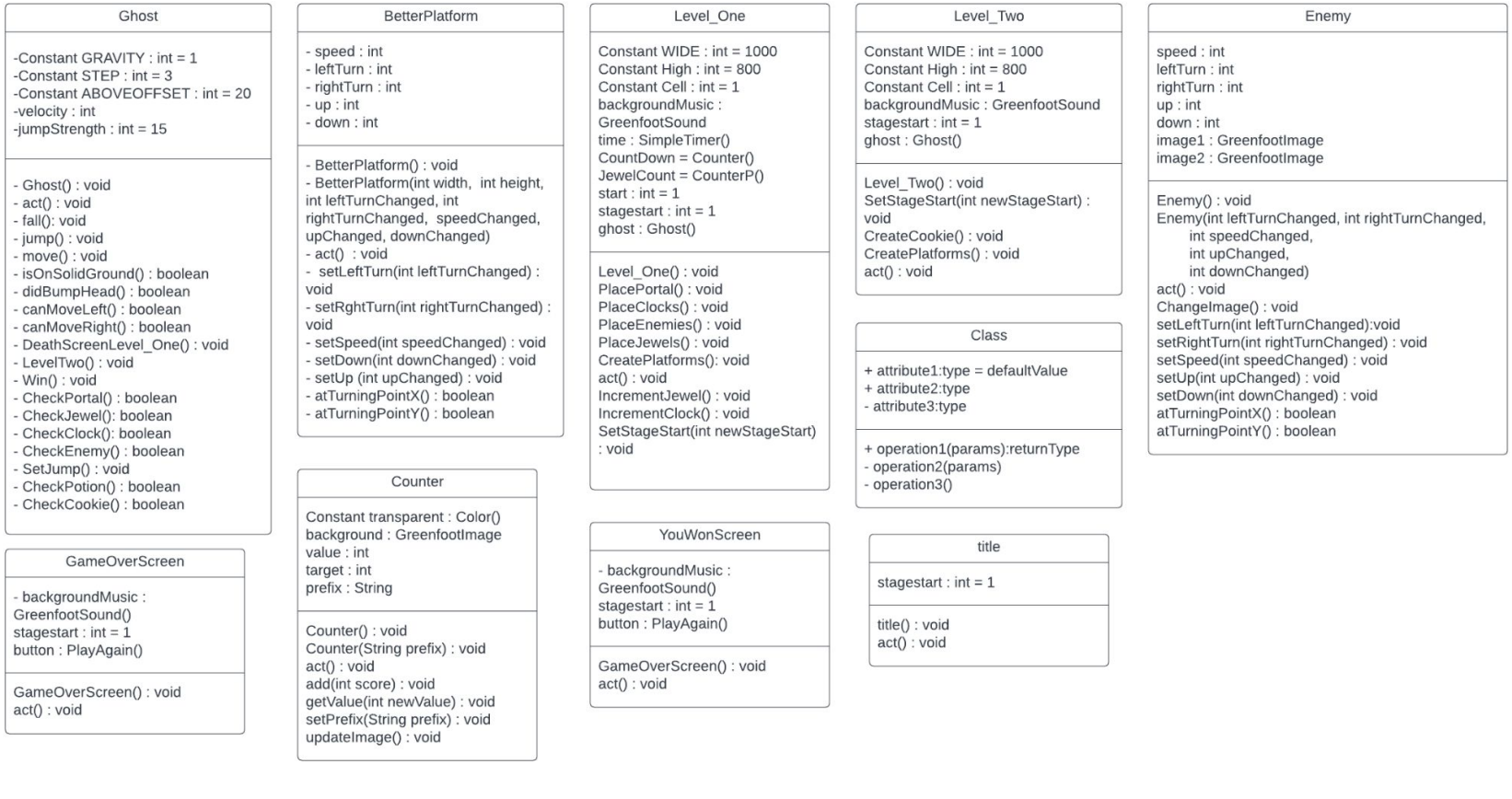
All of the graphics used in the game are from ‘opengameart.org’ website, which allows users to download different graphics which then can be used for them all copyright free, as long as it is for non-profit purposes.

Other graphics were created by the group members or used from the Greenfoot provided graphics. All of the sound effects in the game is also copyright free.

Some of the initial designs might be changed in the end result.

Design Considerations

Prior to coding, all of the classes which were needed for this project had to be identified. The classes used in coding of this game can be found in the following slide - slide 7 . All of the classes and attributes and behaviours were planned out before creating the game. We knew we would need a class for a main character. This class would need to have variables which could be changed from outside the class. For example the jumpStrength variable which determines how high the character can jump. This was changed when the Potion is picked up.


Ghost
-Constant GRAVITY : int = 1
-Constant STEP : int = 3
-Constant ABOVEOFFSET : int = 20
-velocity : int
-jumpStrength : int = 15
- Ghost() : void
- act() : void
- fall(): void
- jump() : void
- move() : void
- isOnSolidGround() : boolean
- didBumpHead() : boolean
- canMoveLeft() : boolean
- canMoveRight() : boolean
- DeathScreenLevel_One() : void
- LevelTwo() : void
- Win() : void
- CheckPortal() : boolean
- CheckJewel(): boolean
- CheckClock(): boolean
- CheckEnemy() : boolean
- SetJump() : void
- CheckPotion() : boolean
- CheckCookie() : boolean
GameOverScreen
- backgroundMusic : GreenfootSound()
stagestart : int = 1
button : PlayAgain()
GameOverScreen() : void
act() : void
BetterPlatform
- speed : int
- leftTurn : int
- rightTurn : int
- up : int
- down : int
- BetterPlatform() : void
- BetterPlatform(int width, int height, int leftTurnChanged, int rightTurnChanged, speedChanged, upChanged, downChanged)
- act() : void
- setLeftTurn(int leftTurnChanged) : void
- setRghtTurn(int rightTurnChanged) : void
- setSpeed(int speedChanged) : void
- setDown(int downChanged) : void
- setUp (int upChanged) : void
- atTurningPointX() : boolean
- atTurningPointY() : boolean
Counter
Constant transparent : Color()
background : GreenfootImage
value : int
target : int
prefix : String
Counter() : void
Counter(String prefix) : void
act() : void
add(int score) : void
getValue(int newValue) : void
setPrefix(String prefix) : void
updateImage() : void
Level_One
Constant WIDE : int = 1000
Constant High : int = 800
Constant Cell : int = 1
backgroundMusic : GreenfootSound
time : SimpleTimer()
CountDown = Counter()
JewelCount = CounterP()
start : int = 1
stagestart : int = 1
ghost : Ghost()
Level_One() : void
PlacePortal() : void
PlaceClocks() : void
PlaceEnemies() : void
PlaceJewels() : void
CreatePlatforms(): void
act() : void
IncrementJewel() : void
IncrementClock() : void
SetStageStart(int newStageStart) : void
YouWonScreen
- backgroundMusic : GreenfootSound()
stagestart : int = 1
button : PlayAgain()
GameOverScreen() : void
act() : void
Level_Two
Constant WIDE : int = 1000
Constant High : int = 800
Constant Cell : int = 1
backgroundMusic : GreenfootSound
stagestart : int = 1
ghost : Ghost()
Level_Two() : void
SetStageStart(int newStageStart) : void
CreateCookie() : void
CreatePlatforms() : void
act() : void
Class
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
title
stagestart : int = 1
title() : void
act() : void
Enemy
speed : int
leftTurn : int
rightTurn : int
up : int
down : int
image1 : GreenfootImage
image2 : GreenfootImage
Enemy() : void
Enemy(int leftTurnChanged, int rightTurnChanged, int speedChanged, int upChanged, int downChanged)
act() : void
ChangeImage() : void
setLeftTurn(int leftTurnChanged):void
setRightTurn(int rightTurnChanged) : void
setSpeed(int speedChanged) : void
setUp(int upChanged) : void
setDown(int downChanged) : void
atTurningPointX() : boolean
atTurningPointY() : boolean

Game Development

One of the biggest issue encountered was the platforms. It became a great challenge to make them solid from all angles. After a few tutorials the platforms were finally able to be rigid from all sides and even improved with movement meaning the platforms could themselves be used as obstacles. Spikes and live hearts were added as actors for the game, however with the time limit we had those were not used in the actual game. With more time given, there would be more levels added and more actors and obstructions in border to make the game more entertaining.

Game Screen

The game screen was created for the game to look more presentable and professional, with some graphics used in the game. As the user presses anywhere on the screen the game will play from level one.

Gameplay Demonstration

The game has been tested as it was developed, with all of the features being developed and improved along the way. As the game came together the fixes were made. The final result can be found in the repository on GitHub, along with all of the code for the presentation and other game resources can be found in the link below.

GitHub video demonstration of the game, as an MP4 file :
https://github.com/CoffeeAddicts/GreenFoot-Game/blob/main/Biscuit%20Blob%20Demo.mp4

Evaluation

A lesson which we have all learned is how to properly work together. Initially at the beginning of the project it was either one person doing the work at a time or everyone with little amount of communication. However, by the end of the project we learnt how to properly communicate and use GitHub to our advantage to be able to work together simultaneously. The game project was an overall success as the game was created with all of the basic requirements needed, and it is completely functional as desired.

Evaluation

Improvements for next include having more complex levels and a tutorial for the user to use before getting into the game. In the current state the game is quite short, having only two levels. It would be more interesting in future improvements to add more levels, with more enemies and adding lives. Some of the improvements would be adding lives and extra enemies, such as spikes. Those graphics were already added to the game, however there was not enough time allocated for all the resources to be used.