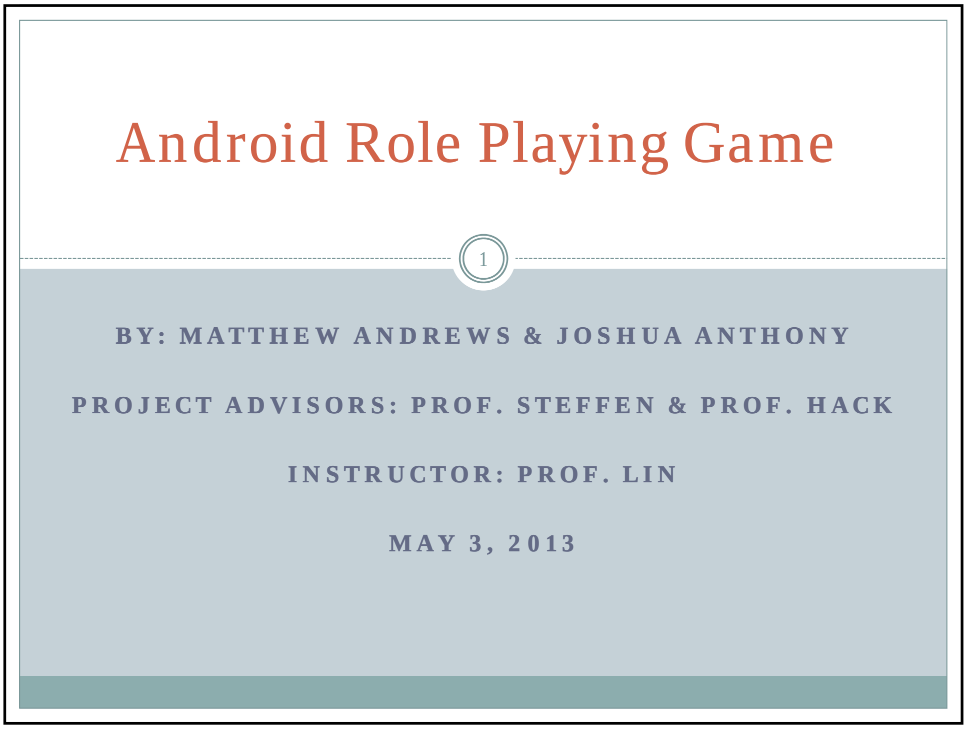


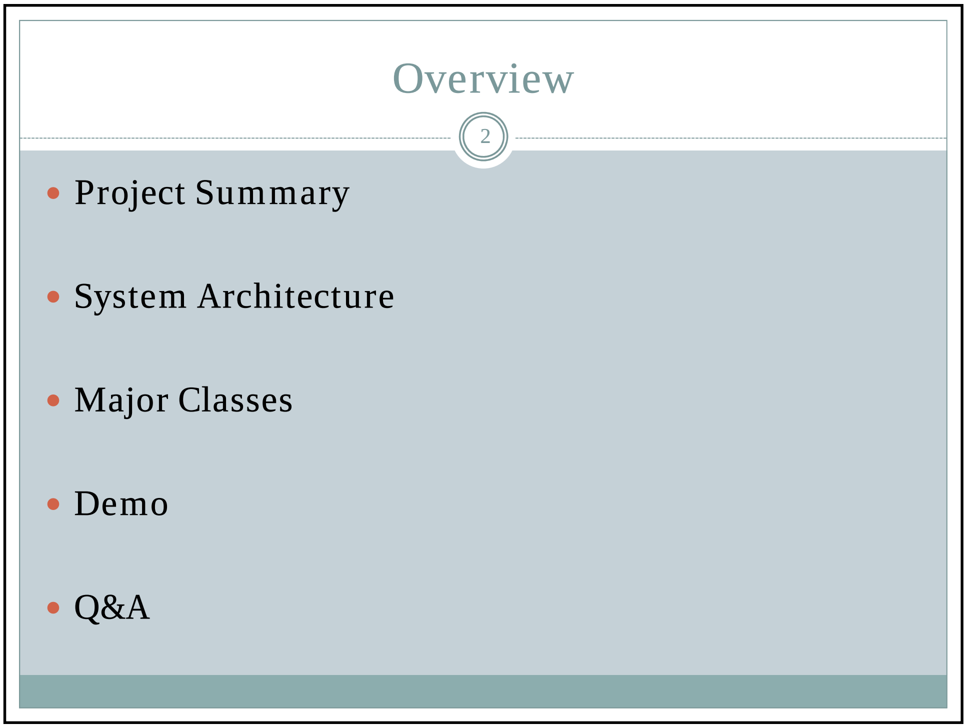
Android Role Playing Game

BY: MATTHEW ANDREWS & JOSHUA ANTHONY

PROJECT ADVISORS: PROF. STEFFEN & PROF. HACK

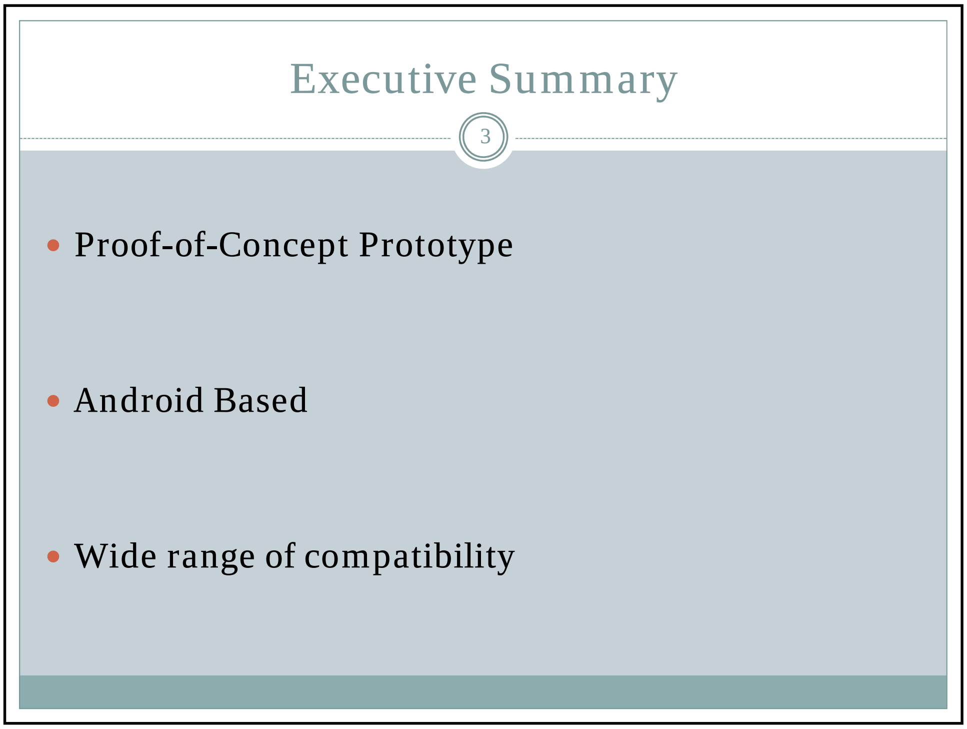
INSTRUCTOR: PROF. LIN

MAY 3, 2013



Overview

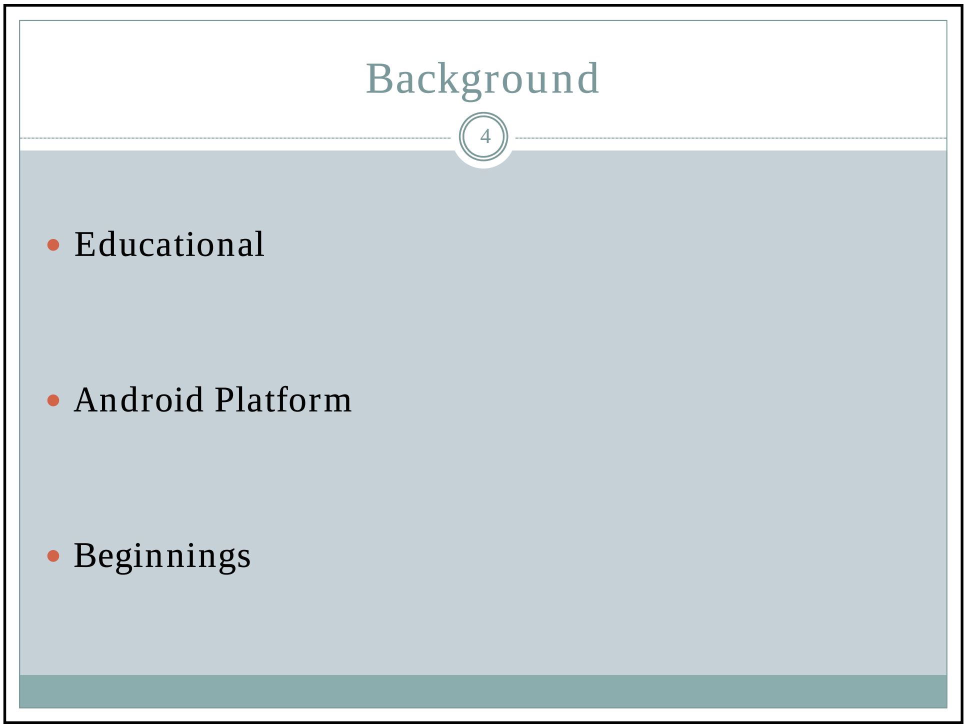
- Project Summary
- System Architecture
- Major Classes
- Demo
- Q&A



Executive Summary

3

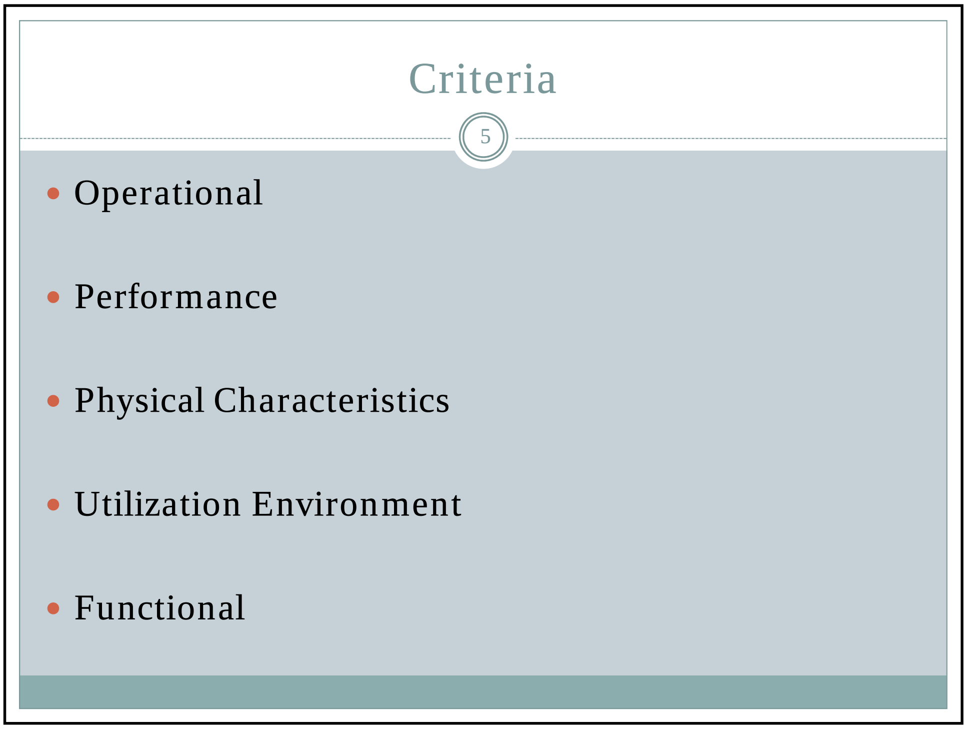
- Proof-of-Concept Prototype
- Android Based
- Wide range of compatibility



Background

4

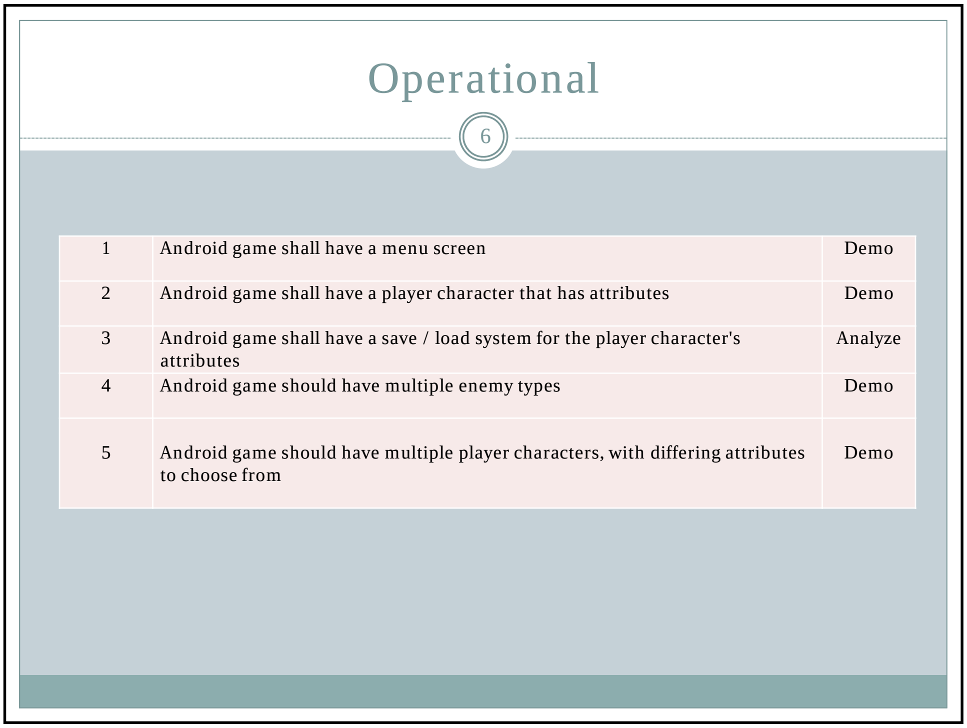
- Educational
- Android Platform
- Beginnings



Criteria

5

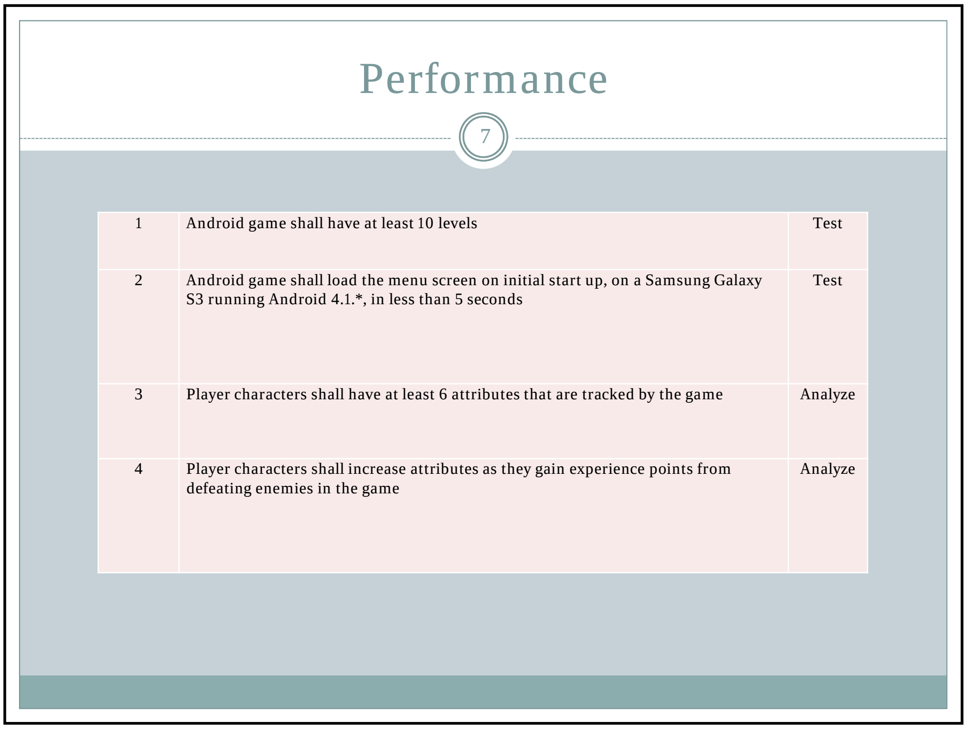
- Operational
- Performance
- Physical Characteristics
- Utilization Environment
- Functional



Operational

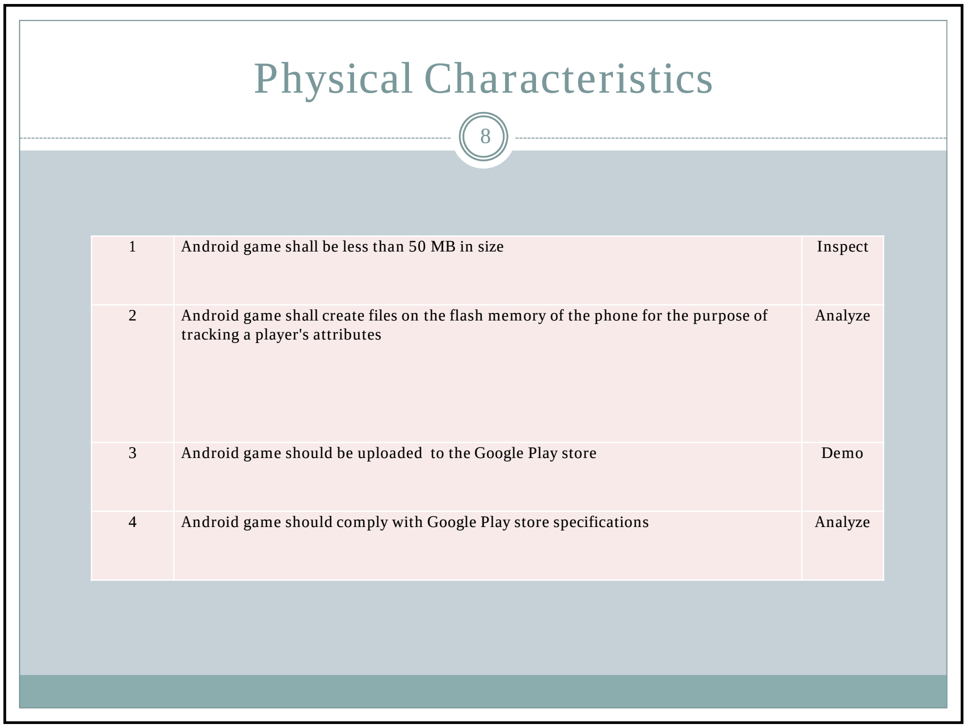
6

1	Android game shall have a menu screen	Demo
2	Android game shall have a player character that has attributes	Demo
3	Android game shall have a save / load system for the player character's attributes	Analyze
4	Android game should have multiple enemy types	Demo
5	Android game should have multiple player characters, with differing attributes to choose from	Demo



Performance

1	Android game shall have at least 10 levels	Test
2	Android game shall load the menu screen on initial start up, on a Samsung Galaxy S3 running Android 4.1.*, in less than 5 seconds	Test
3	Player characters shall have at least 6 attributes that are tracked by the game	Analyze
4	Player characters shall increase attributes as they gain experience points from defeating enemies in the game	Analyze



Physical Characteristics

1	Android game shall be less than 50 MB in size	Inspect
2	Android game shall create files on the flash memory of the phone for the purpose of tracking a player's attributes	Analyze
3	Android game should be uploaded to the Google Play store	Demo
4	Android game should comply with Google Play store specifications	Analyze

Environment & Functional

9

- Environment

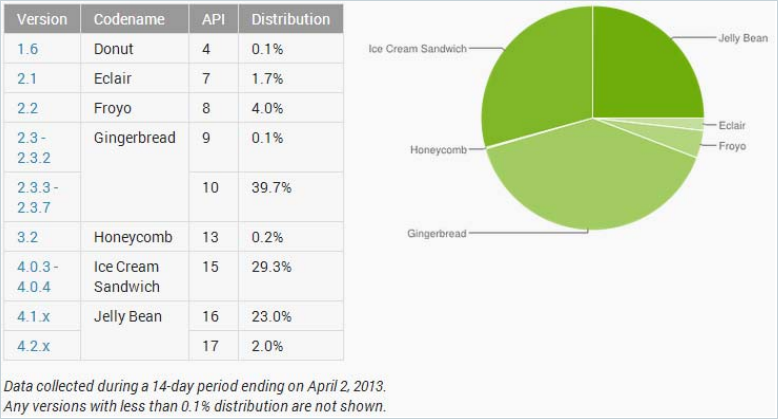
1	Android game shall run on an emulated environment of Android 4.1.*	Analyze
2	Android game shall run a Samsung Galaxy S3, running Android 4.1.*	Analyze

- Functional

1	Android game shall utilize a text file to save player attributes	Demo
2	Android game shall have enemies that move towards the player every movement of the game	Demo
3	Android game shall contain a system that measures player's and an enemy's attributes against each other and determines the winner of the struggle.	Analyze

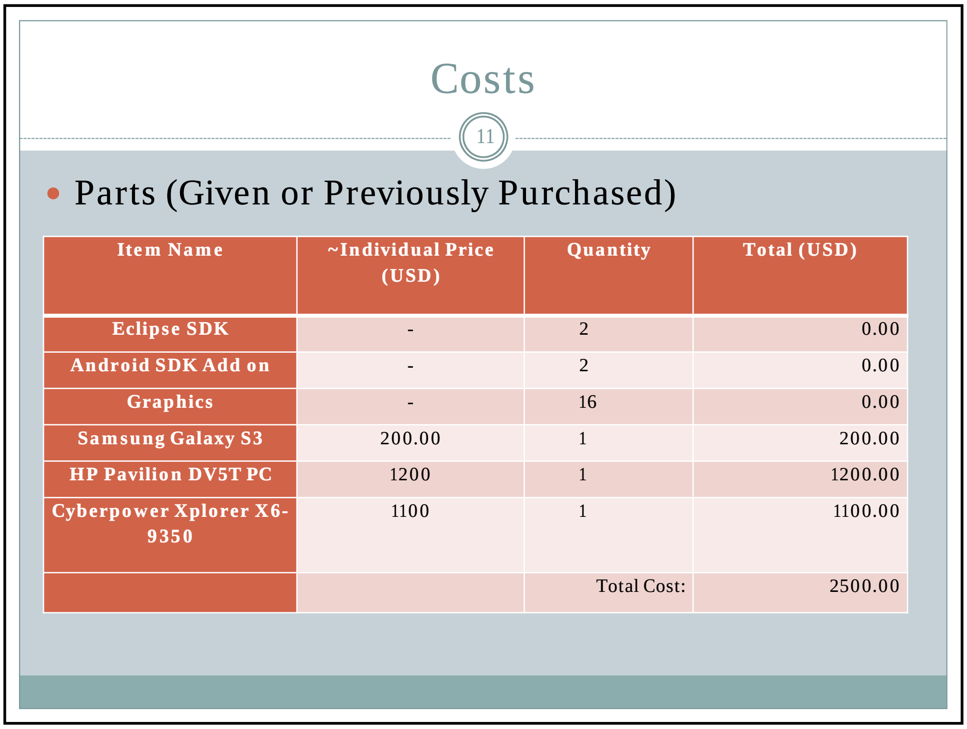
Why Jelly Bean?

10



Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.7%
2.2	Froyo	8	4.0%
2.3 - 2.3.2	Gingerbread	9	0.1%
2.3.3 - 2.3.7		10	39.7%
3.2	Honeycomb	13	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29.3%
4.1.x	Jelly Bean	16	23.0%
4.2.x		17	2.0%

Data collected during a 14-day period ending on April 2, 2013.
Any versions with less than 0.1% distribution are not shown.

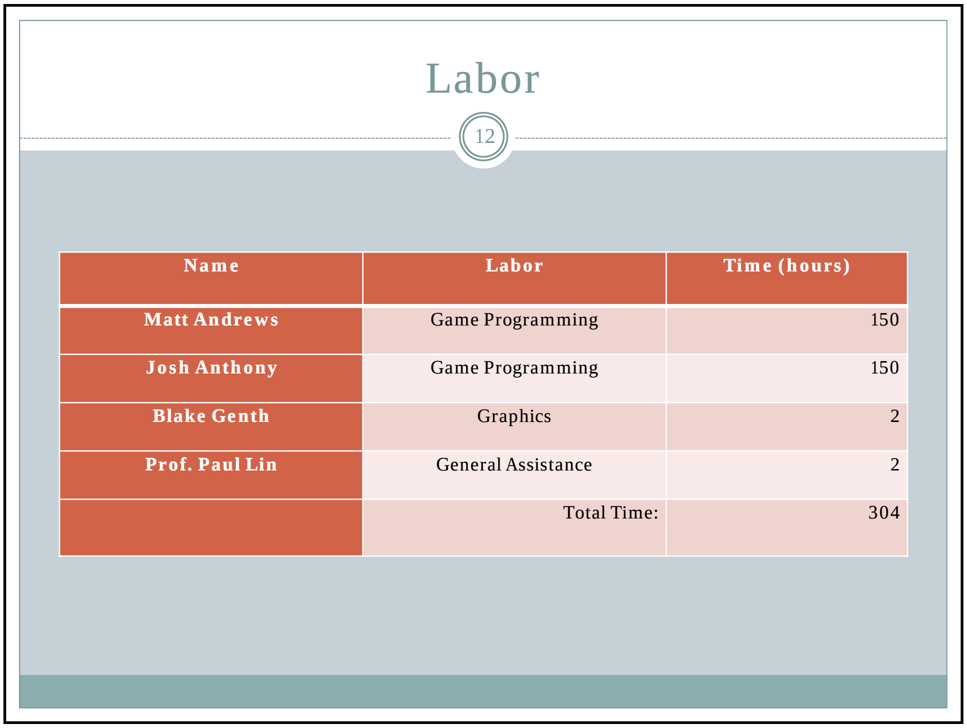


Costs

11

- Parts (Given or Previously Purchased)

Item Name	~Individual Price (USD)	Quantity	Total (USD)
Eclipse SDK	-	2	0.00
Android SDK Add on	-	2	0.00
Graphics	-	16	0.00
Samsung Galaxy S3	200.00	1	200.00
HP Pavilion DV5T PC	1200	1	1200.00
Cyberpower Xplorer X6-9350	1100	1	1100.00
		Total Cost:	2500.00



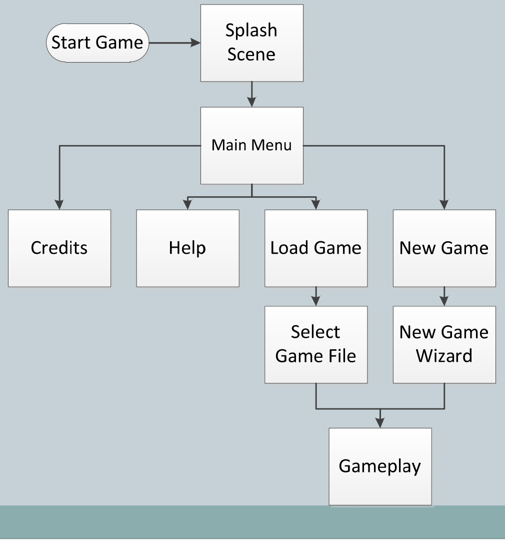
Labor

12

Name	Labor	Time (hours)
Matt Andrews	Game Programming	150
Josh Anthony	Game Programming	150
Blake Genth	Graphics	2
Prof. Paul Lin	General Assistance	2
	Total Time:	304

Software Architecture

13



Start Game

Splash Scene

Main Menu

Credits

Help

Load Game

New Game

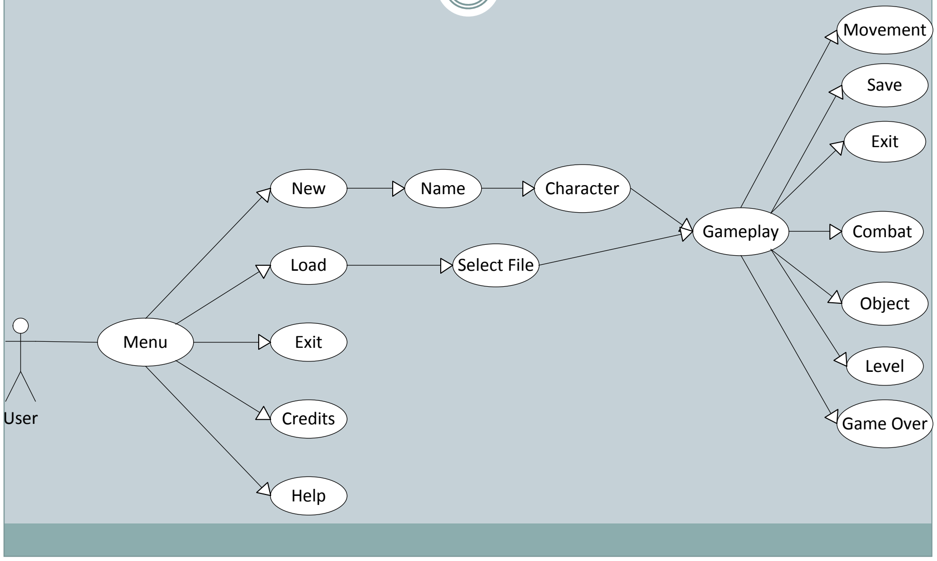
Select Game File

New Game Wizard

Gameplay

Software Architecture

14



User

Menu

New

Load

Exit

Credits

Help

Name

Character

Select File

Gameplay

Movement

Save

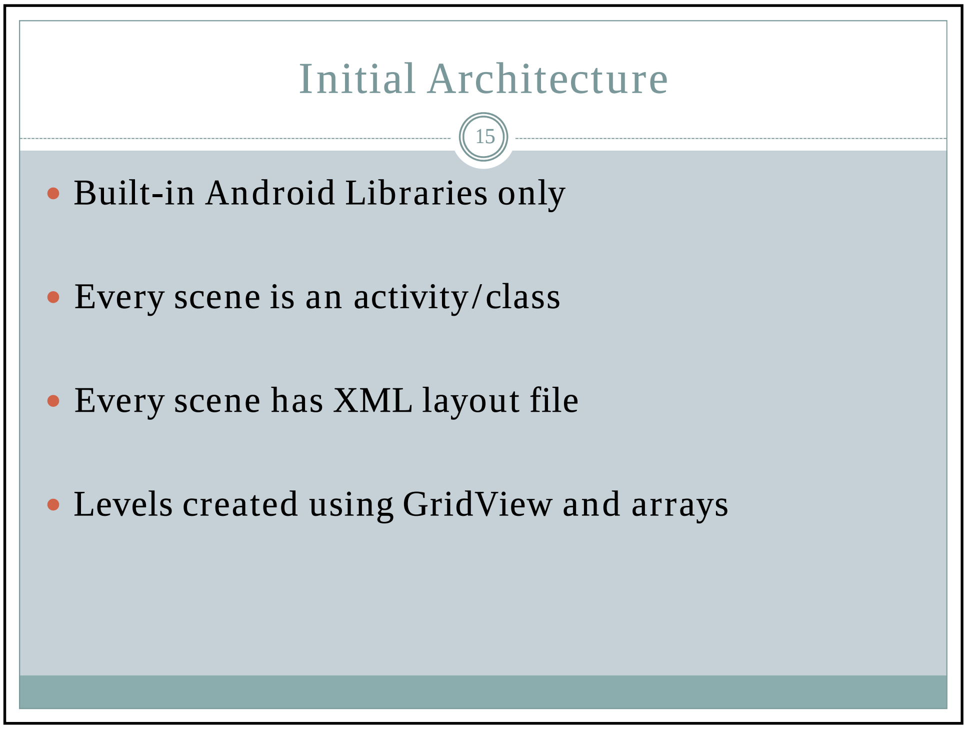
Exit

Combat

Object

Level

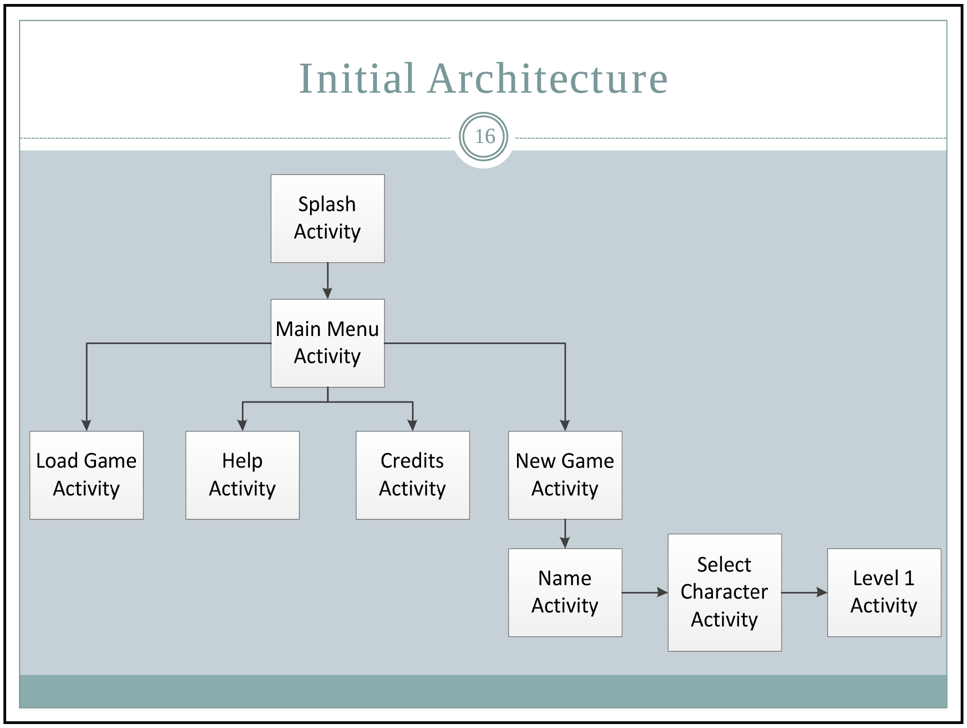
Game Over



Initial Architecture

15

- Built-in Android Libraries only
- Every scene is an activity/class
- Every scene has XML layout file
- Levels created using GridView and arrays



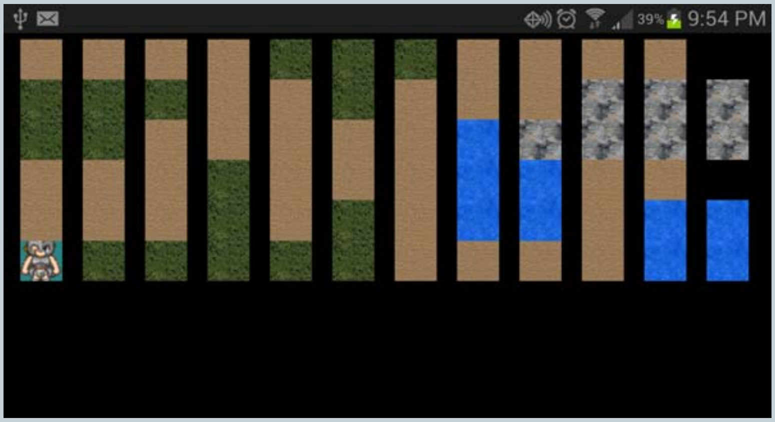
Initial Architecture

16

Initial Testing

17

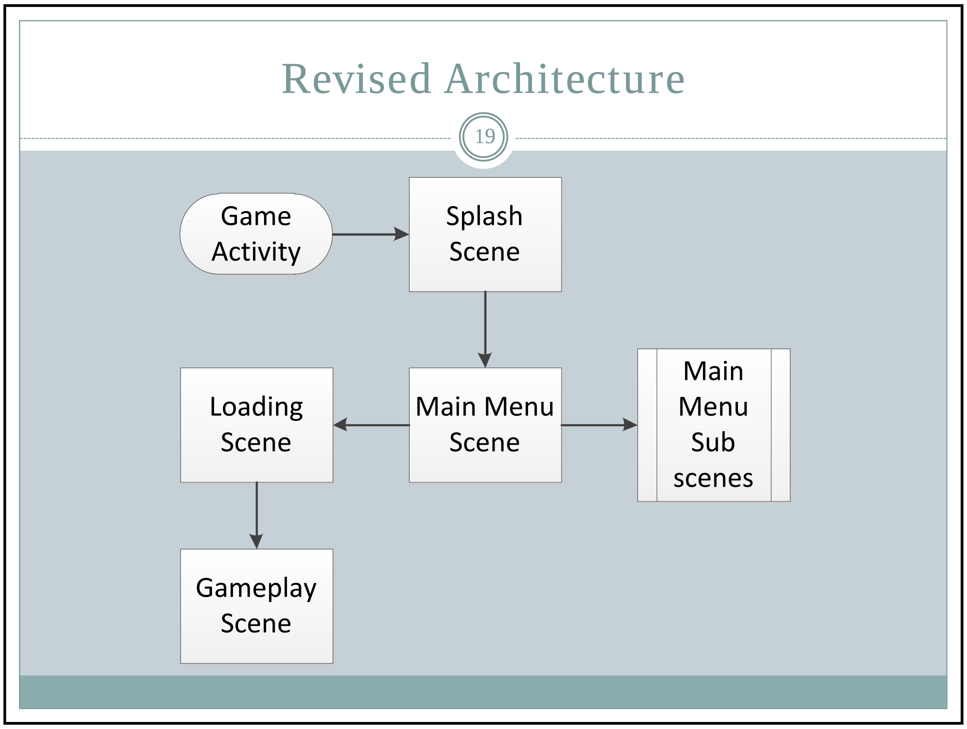
- Screen Resolution Issue



Solution

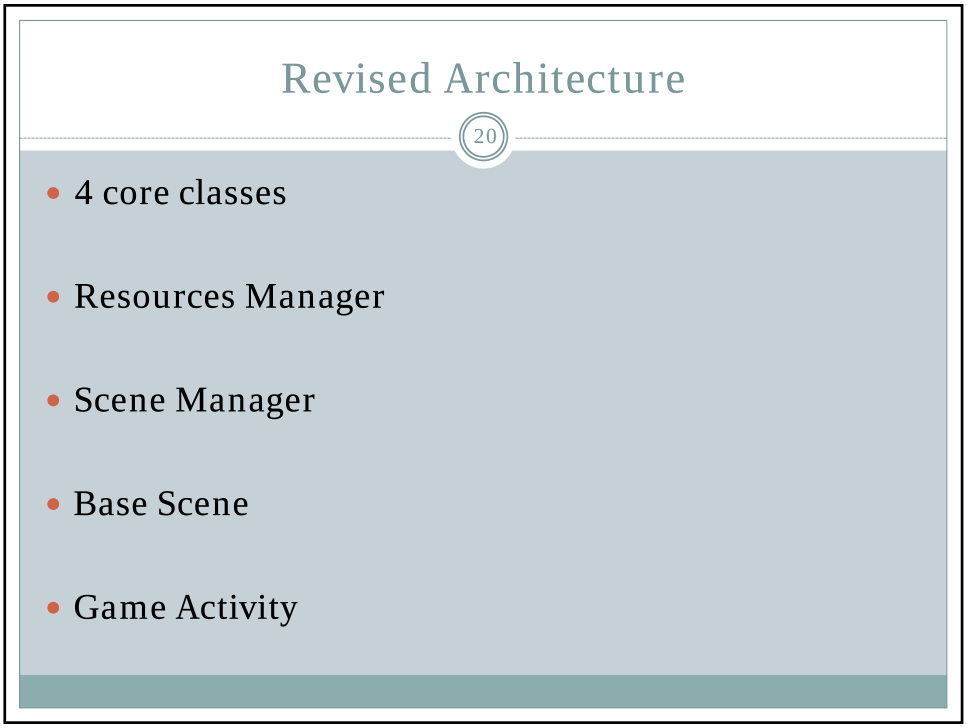
18

- AndEngine
- Built-in scaling screen resolution
- HUD Libraries
- Architecture Overhaul
- Behind Schedule



Revised Architecture

19



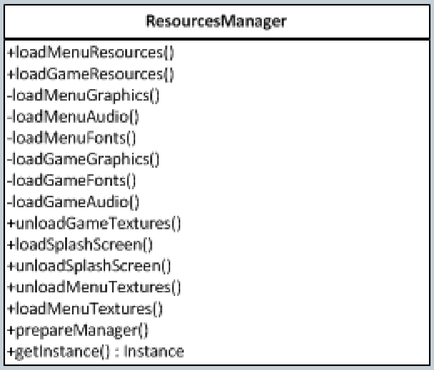
Revised Architecture

20

- 4 core classes
- Resources Manager
- Scene Manager
- Base Scene
- Game Activity

Resources Manager

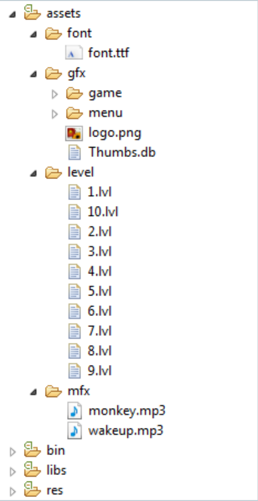
- Class responsibilities
- Class variables
- Class methods



```
ResourcesManager
+loadMenuResources()
+loadGameResources()
-loadMenuGraphics()
-loadMenuAudio()
-loadMenuFonts()
-loadGameGraphics()
-loadGameFonts()
-loadGameAudio()
+unloadGameTextures()
+loadSplashScreen()
+unloadSplashScreen()
+unloadMenuTextures()
+loadMenuTextures()
+prepareManager()
+getInstance() : Instance
```

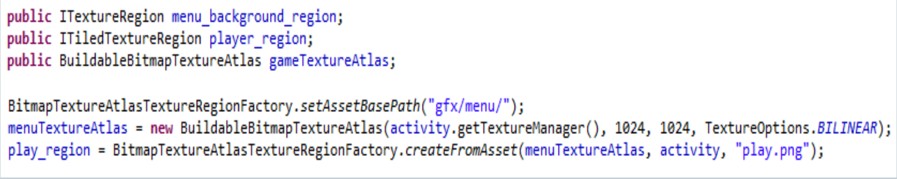
Assets

- Font
- Graphics
- Level
- Music



```
assets
  font
    font.ttf
  gfx
    game
    menu
    logo.png
    Thumbs.db
  level
    1.lvl
    10.lvl
    2.lvl
    3.lvl
    4.lvl
    5.lvl
    6.lvl
    7.lvl
    8.lvl
    9.lvl
  mfx
    monkey.mp3
    wakeup.mp3
bin
libs
res
```

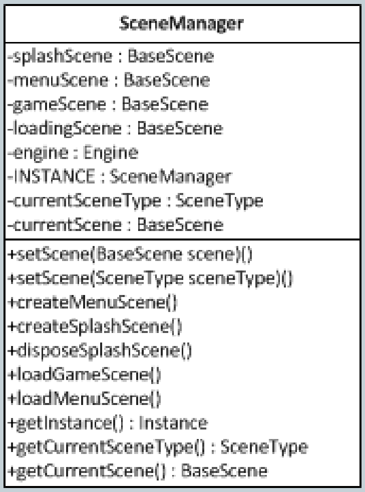
Resources Manager Sample Code



```
public ITextureRegion menu_background_region;
public ITiledTextureRegion player_region;
public BuildableBitmapTextureAtlas gameTextureAtlas;

BitmapTextureAtlasTextureRegionFactory.setAssetBasePath("gfx/menu/");
menuTextureAtlas = new BuildableBitmapTextureAtlas(activity.getTextureManager(), 1024, 1024, TextureOptions.BILINEAR);
play_region = BitmapTextureAtlasTextureRegionFactory.createFromAsset(menuTextureAtlas, activity, "play.png");
```

Scene Manager

- Responsibilities
- 4 total scenes
- Loading Scenes
- Disposing of Scenes



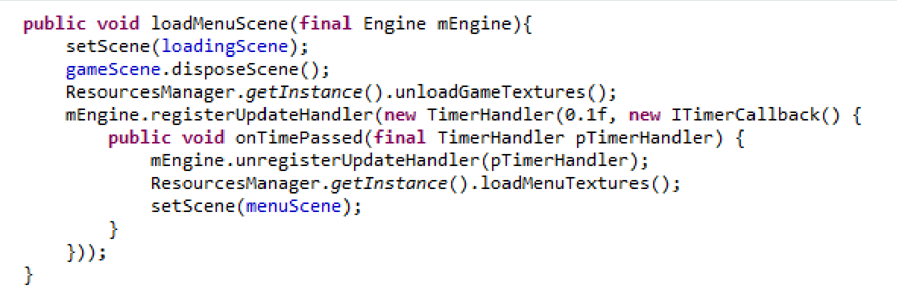
SceneManager

-splashScene : BaseScene
-menuScene : BaseScene
-gameScene : BaseScene
-loadingScene : BaseScene
-engine : Engine
-INSTANCE : SceneManager
-currentSceneType : SceneType
-currentScene : BaseScene

+setScene(BaseScene scene)()
+setScene(SceneType sceneType)()
+createMenuScene()
+createSplashScene()
+disposeSplashScene()
+loadGameScene()
+loadMenuScene()
+getInstance() : Instance
+getCurrentSceneType() : SceneType
+getCurrentScene() : BaseScene

Scene Manager Sample Code

25

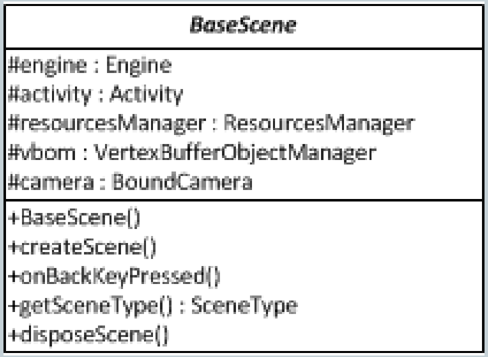


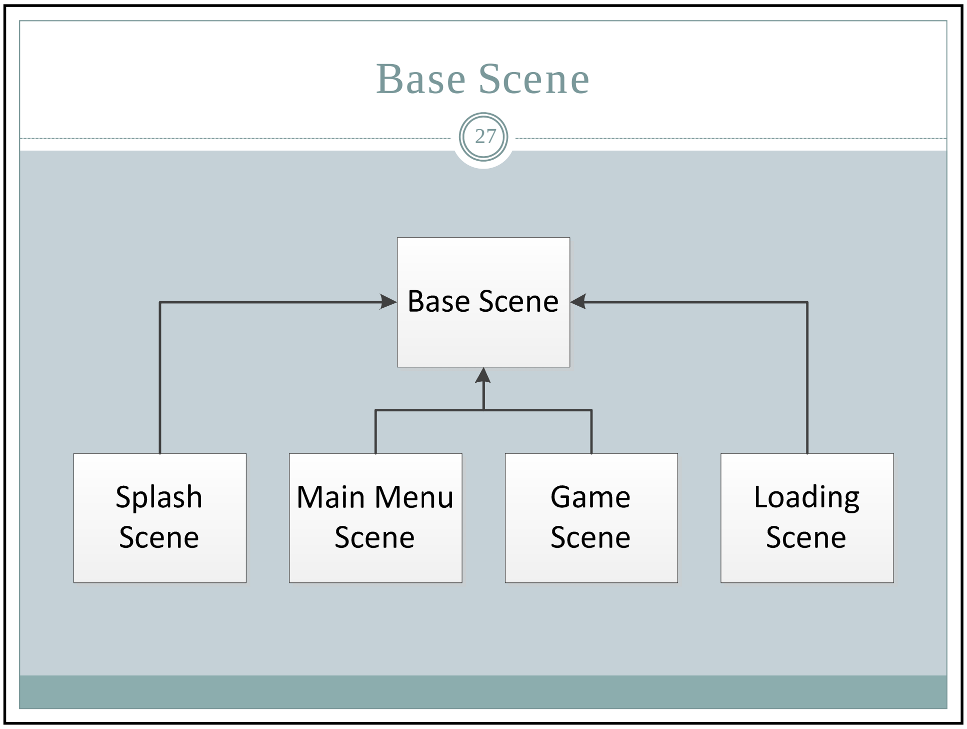
```
public void loadMenuScene(final Engine mEngine){
    setScene(loadingScene);
    gameScene.disposeScene();
    ResourcesManager.getInstance().unloadGameTextures();
    mEngine.registerUpdateHandler(new TimerHandler(0.1f, new ITimerCallback() {
        public void onTimePassed(final TimerHandler pTimerHandler) {
            mEngine.unregisterUpdateHandler(pTimerHandler);
            ResourcesManager.getInstance().loadMenuTextures();
            setScene(menuScene);
        }
    }));
}
```

Base Scene

26

- Template for all scenes
- Extended by all scenes
- Abstract Methods





Base Scene

27

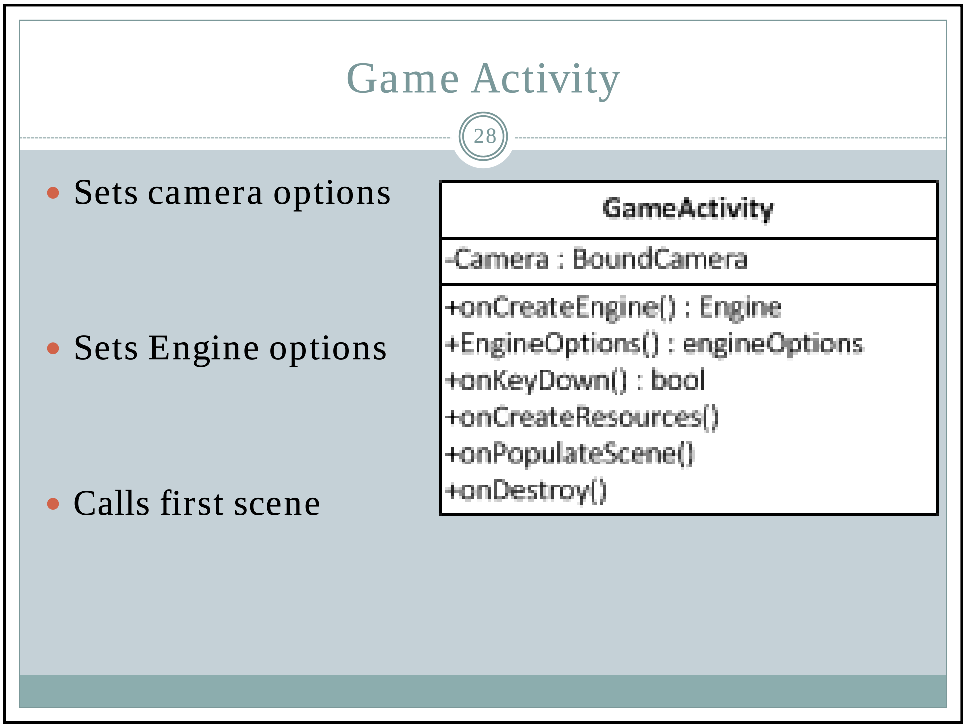
Base Scene

Splash Scene

Main Menu Scene

Game Scene

Loading Scene



Game Activity

28

- Sets camera options
- Sets Engine options
- Calls first scene

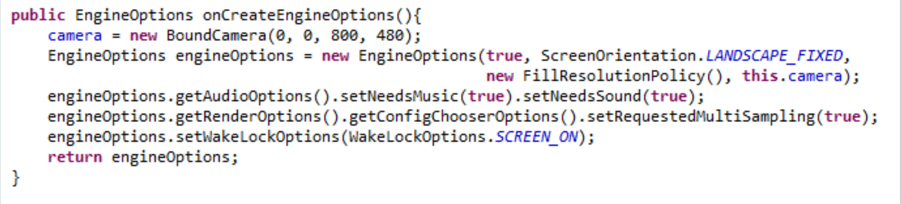
GameActivity

-Camera : BoundCamera

+onCreateEngine() : Engine
+EngineOptions() : engineOptions
+onKeyDown() : bool
+onCreateResources()
+onPopulateScene()
+onDestroy()

Game Activity Sample Code

29

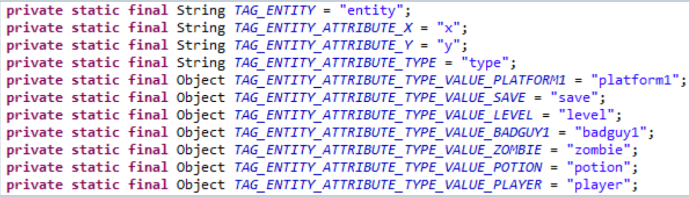


```
public EngineOptions onCreateEngineOptions(){
    camera = new BoundCamera(0, 0, 800, 480);
    EngineOptions engineOptions = new EngineOptions(true, ScreenOrientation.LANDSCAPE_FIXED,
                                                new FillResolutionPolicy(), this.camera);
    engineOptions.getAudioOptions().setNeedsMusic(true).setNeedsSound(true);
    engineOptions.getRenderOptions().getConfigChooserOptions().setRequestedMultiSampling(true);
    engineOptions.setWakeLockOptions(WakeLockOptions.SCREEN_ON);
    return engineOptions;
}
```

Levels

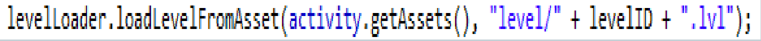
30

- Created using XML files
- Level Item Creation



```
private static final String TAG_ENTITY = "entity";
private static final String TAG_ENTITY_ATTRIBUTE_X = "x";
private static final String TAG_ENTITY_ATTRIBUTE_Y = "y";
private static final String TAG_ENTITY_ATTRIBUTE_TYPE = "type";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_PLATFORM1 = "platform1";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_SAVE = "save";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_LEVEL = "level";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_BADGUY1 = "badguy1";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_ZOMBIE = "zombie";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_POTION = "potion";
private static final Object TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_PLAYER = "player";
```

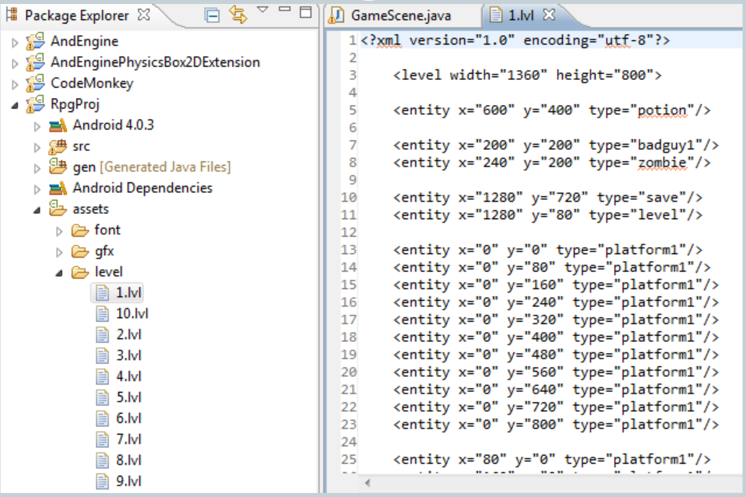
- Level is called using:



```
levelLoader.loadLevelFromAsset(activity.getAssets(), "level/" + levelID + ".lvl");
```

Levels

31



```
<?xml version="1.0" encoding="utf-8"?>

    <level width="1360" height="800">

    <entity x="600" y="400" type="potion"/>

    <entity x="200" y="200" type="badguy1"/>
    <entity x="240" y="200" type="zombie"/>

    <entity x="1280" y="720" type="save"/>
    <entity x="1280" y="80" type="level"/>

    <entity x="0" y="0" type="platform1"/>
    <entity x="0" y="80" type="platform1"/>
    <entity x="0" y="160" type="platform1"/>
    <entity x="0" y="240" type="platform1"/>
    <entity x="0" y="320" type="platform1"/>
    <entity x="0" y="400" type="platform1"/>
    <entity x="0" y="480" type="platform1"/>
    <entity x="0" y="560" type="platform1"/>
    <entity x="0" y="640" type="platform1"/>
    <entity x="0" y="720" type="platform1"/>
    <entity x="0" y="800" type="platform1"/>

    <entity x="80" y="0" type="platform1"/>
```

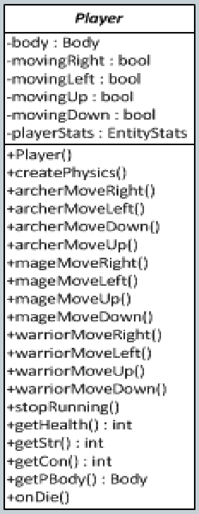
Player Class

32

- Movement Direction

- Movement Speed

- Attributes



Player

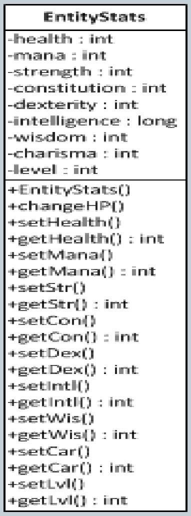
-body : Body
-movingRight : bool
-movingLeft : bool
-movingUp : bool
-movingDown : bool
-playerStats : EntityStats

+Player()
+createPhysics()
+archerMoveRight()
+archerMoveLeft()
+archerMoveDown()
+archerMoveUp()
+mageMoveRight()
+mageMoveLeft()
+mageMoveUp()
+mageMoveDown()
+warriorMoveRight()
+warriorMoveLeft()
+warriorMoveUp()
+warriorMoveDown()
+stopRunning()
+getHealth() : int
+getStr() : int
+getCon() : int
+getPBody() : Body
+onDie()

Attributes Class

33

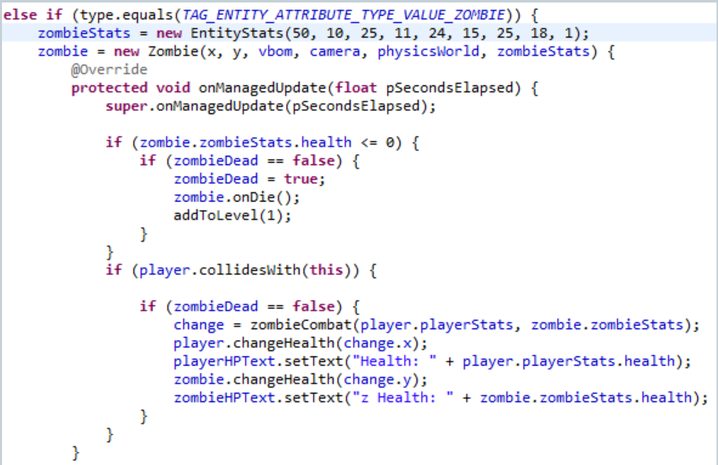
- Attribute Creation
- Attribute Storage
- Changing Health



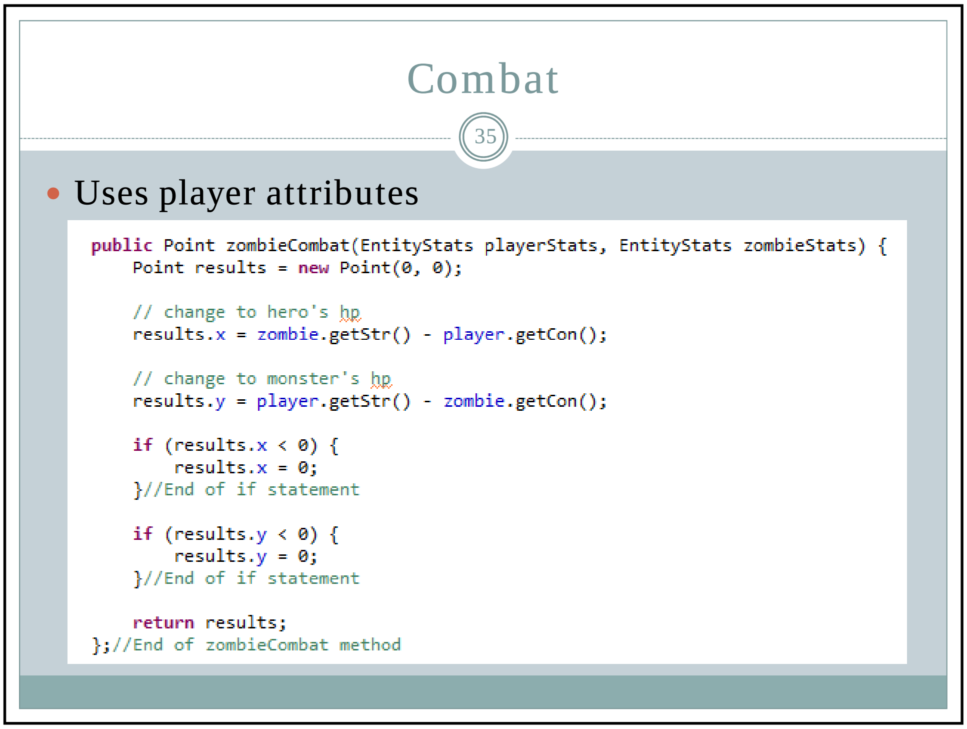
```
EntityStats
-health : int
-mana : int
-strength : int
-constitution : int
-dexterity : int
-intelligence : long
-wisdom : int
-charisma : int
-level : int
+EntityStats()
+changeHP()
+setHealth()
+getHealth() : int
+setMana()
+getMana() : int
+setStr()
+getStr() : int
+setCon()
+getCon() : int
+setDex()
+getDex() : int
+setIntl()
+getIntl() : int
+setWis()
+getWis() : int
+setCar()
+getCar() : int
+setLvl()
+getLvl() : int
```

Collisions

34



```
else if (type.equals(TAG_ENTITY_ATTRIBUTE_TYPE_VALUE_ZOMBIE)) {
    zombieStats = new EntityStats(50, 10, 25, 11, 24, 15, 25, 18, 1);
    zombie = new Zombie(x, y, vbom, camera, physicsWorld, zombieStats) {
        @Override
        protected void onManagedUpdate(float pSecondsElapsed) {
            super.onManagedUpdate(pSecondsElapsed);

            if (zombie.zombieStats.health <= 0) {
                if (zombieDead == false) {
                    zombieDead = true;
                    zombie.onDie();
                    addToLevel(1);
                }
            }
            if (player.collidesWith(this)) {

                if (zombieDead == false) {
                    change = zombieCombat(player.playerStats, zombie.zombieStats);
                    player.changeHealth(change.x);
                    playerHPText.setText("Health: " + player.playerStats.health);
                    zombie.changeHealth(change.y);
                    zombieHPText.setText("z Health: " + zombie.zombieStats.health);
                }
            }
        }
```

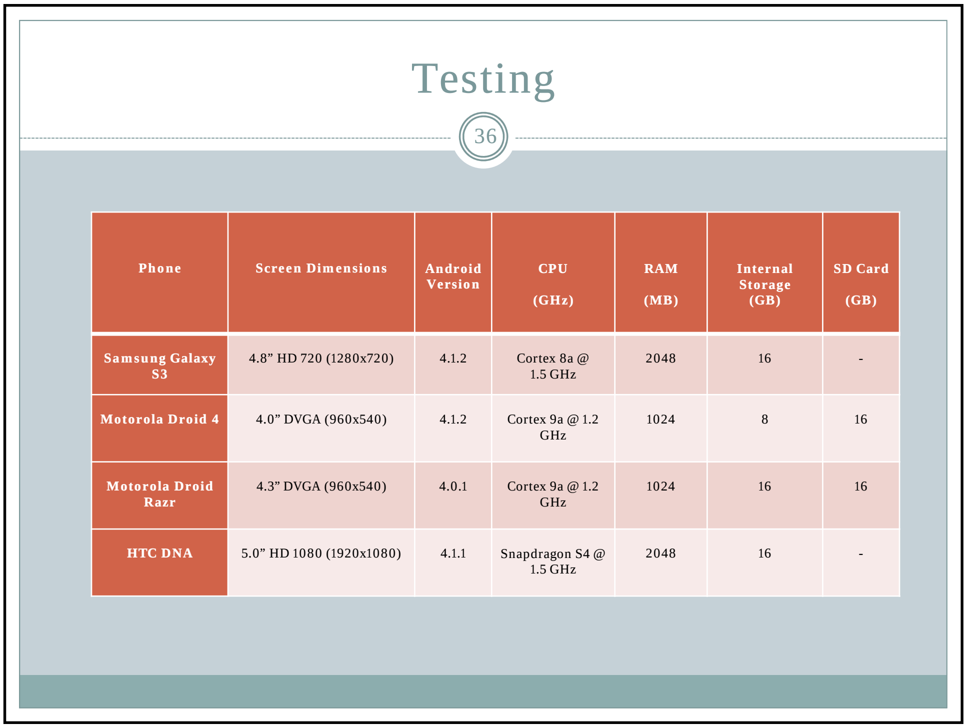


Combat

35

- Uses player attributes

```
public Point zombieCombat(EntityStats playerStats, EntityStats zombieStats) {
    Point results = new Point(0, 0);

    // change to hero's hp
    results.x = zombie.getStr() - player.getCon();

    // change to monster's hp
    results.y = player.getStr() - zombie.getCon();

    if (results.x < 0) {
        results.x = 0;
    }//End of if statement

    if (results.y < 0) {
        results.y = 0;
    }//End of if statement

    return results;
};//End of zombieCombat method
```



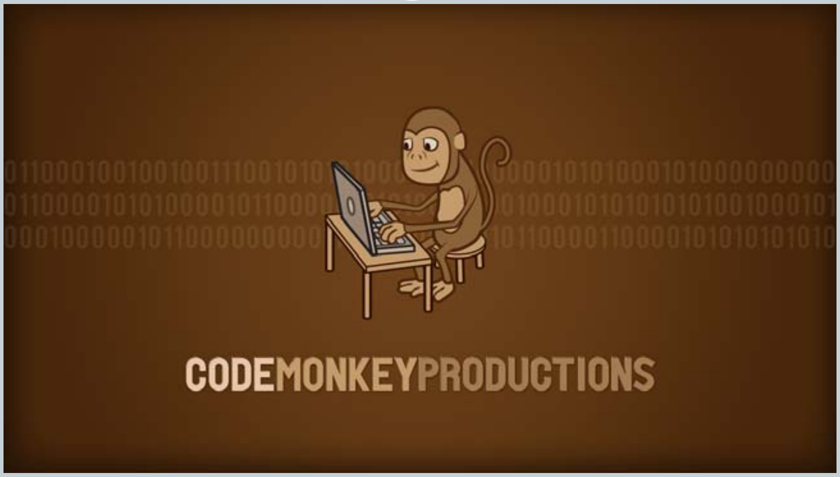
Testing

36

Phone	Screen Dimensions	Android Version	CPU (GHz)	RAM (MB)	Internal Storage (GB)	SD Card (GB)
Samsung Galaxy S3	4.8” HD 720 (1280x720)	4.1.2	Cortex 8a @ 1.5 GHz	2048	16	-
Motorola Droid 4	4.0” DVGA (960x540)	4.1.2	Cortex 9a @ 1.2 GHz	1024	8	16
Motorola Droid Razr	4.3” DVGA (960x540)	4.0.1	Cortex 9a @ 1.2 GHz	1024	16	16
HTC DNA	5.0” HD 1080 (1920x1080)	4.1.1	Snapdragon S4 @ 1.5 GHz	2048	16	-

Demo

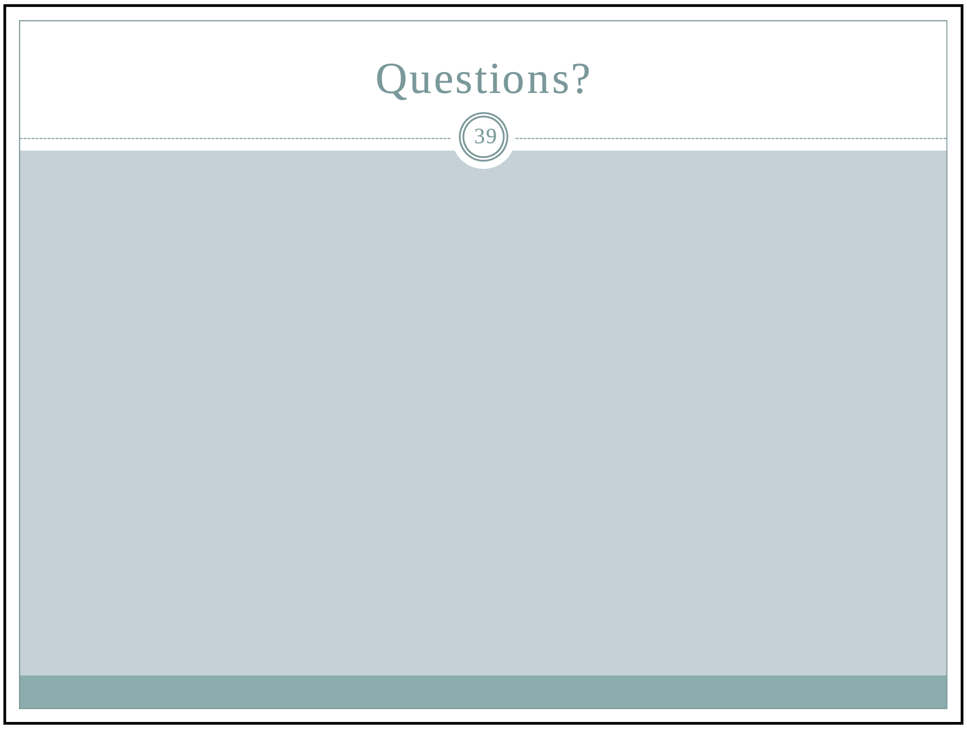
37



Conclusion

38

- Lessons Learned
- Time Management
- Documentation
- Perseverance



Questions?

39