Дипломная работа

на тему:

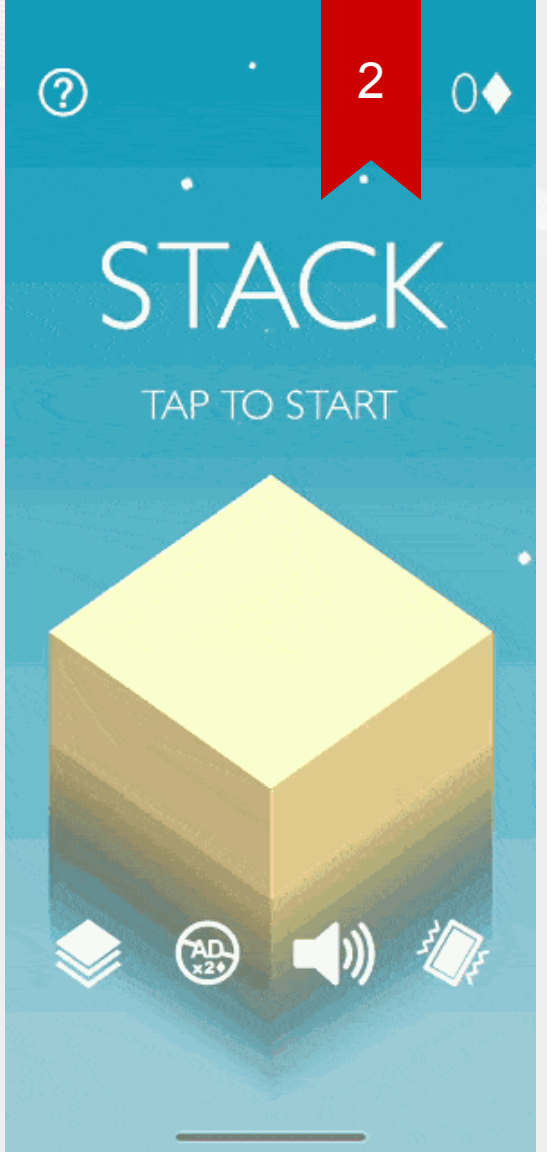
Разработка компьютерной игры в жанре Казуальные «HyperCasual»

Выполнил: студент группы 1304 Воробьёв К.А.

Научный руководитель: Киселёв Леонид Александрович

Введение

Данная работа посвящена разработке двухмерной компьютерной игры «HyperCasual» на движке Unity3D. Проект относится к жанру казуальных игр. Казуальная игра (от англ. casual game) — компьютерная игра, предназначенная для широкого круга пользователей. Казуальные игры отличаются простыми правилами и не требуют от пользователя



Компьютерная (видео-) игра - это

компьютерная программа, служащая для организации игрового процесса, связи с партнёрами по игре, или сама выступающая в качестве партнёра.

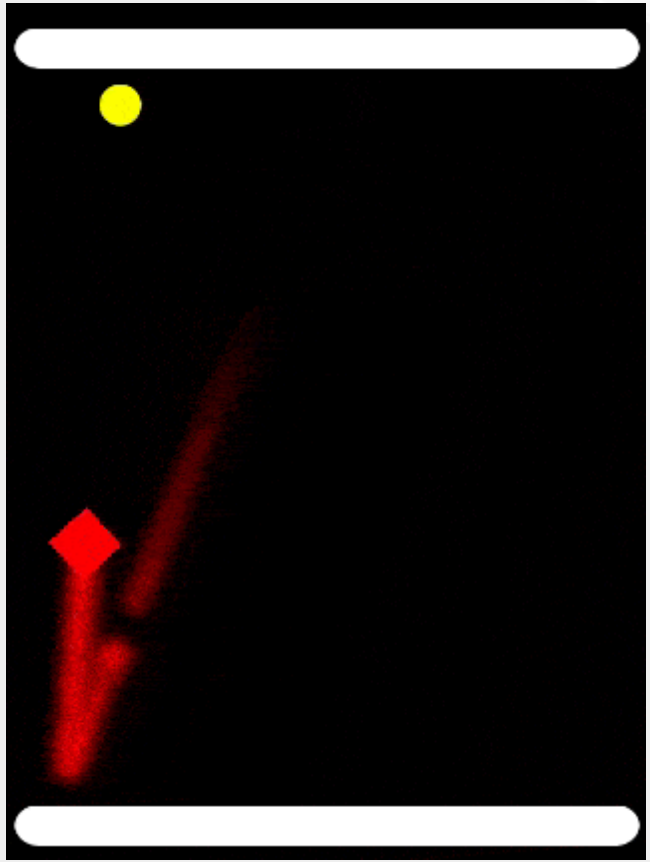
Основные составляющие видео игры - это Сеттинг - место действия, условия, время, среда, Геймплей - интерактивное взаимодействие

Выбор технологии

	GameMaker 2	Unity3D
Простота освоения	✔	✔
Лицензия для старта	7500 руб./12 мес.	✖
Набор возможностей	Меньше	Больше
Работа с измерениями	2D	2D, 3D
Используемый язык	GML	C#, UnityScript
Система сохранения	Встроенная	Реализуемая

Корневая механика

Суть игры: по игровому полю самостоятельно перемещается игрок, курсируя между двумя точками, расположенными на разных концах поля. При достижении одной цели пользователь получает одно очко и игрок переключается на цель с другой стороны поля. Сам же пользователь может лишь менять направление движения между двумя точками, но добравшись до края поля игрок развернётся самостоятельно.



Основной скрипт игры: x2_PlayerController: переменные

[SerializeField] DataBase **inGameSet**;

[SerializeField] Profile **profile**;

public GameObject **gameStats**;

public GameObject **goal1**;

public GameObject **goal2**;

public GameObject **Explosion**;

public GameObject **ScoreText**;

public GameObject **ComboText**;

public GameObject **DeathText**;

public GameObject **scrapController**;

public int **score** = 0;

public int **combo** = 0;

public int **death** = 0;

public float **body_speed** = 2f;

public float **body_rotation** = 1f;

public bool **check** = false;

public int **hp**;

public GameObject[] **hpArr** = new GameObject[10];

public GameObject **hpXY**;

public GameObject **hpBody**;

public float **Wmax** = 5f;

public float **Wobj** = 0.6f;

private Vector3 **g1pos**;

private Vector3 **g2pos**;

x2_PlayerController: Start()

hp = inGameSet.hpStart;

g1pos = new Vector3 (Random.Range(-2.7f,2.7f), goal1.transform.position.y, 1);

g2pos = new Vector3 (Random.Range(-2.7f,2.7f), goal2.transform.position.y, 1);

goal1.transform.position = g1pos;

goal1.SetActive (true);

goal2.transform.position = g2pos;

goal2.SetActive (false);

x2_PlayerController: Start()

```
var start_g2pos = goal2.transform.position;

transform.position = new Vector3(start_g2pos.x, start_g2pos.y, 2f);


hpControler("Start");

scrapController.GetComponent<ScrapSpawner>().ScrapSpawn();


gameStats.GetComponent<subMenuController>().pos =

gameStats.GetComponent<DeadMenucontroller>().startPos.transform.position;
```

x2_PlayerController: Update()

```
if(transform.position == goal1.transform.position || transform.position ==
goal2.transform.position) check = !check;
if(!check){
   transform.position = Vector3.MoveTowards(transform.position, goal1.transform.position,
   body_speed * Time.deltaTime);
} else {
   transform.position = Vector3.MoveTowards(transform.position, goal2.transform.position,
   body_speed * Time.deltaTime);
}
transform.rotation *= Quaternion.Euler(0f,0f,body_rotation);
```

x2_PlayerController: OnCollisionEnter2D(Collisio D other)

```
if(other.gameObject.tag == "goal1"){
    goal1.SetActive (false);
    goal2.SetActive (true);
    g2pos = new Vector3(Random.Range(-2.7f,2.7f), goal2.transform.position.y, 1);
    goal2.transform.position = g2pos;
}else if(other.gameObject.tag == "goal2"){
    goal1.SetActive (true);
    goal2.SetActive (false);
```

x2_PlayerController: OnCollisionEnter2D(Collision2D other)

```
score++;

ScoreText.GetComponent<Text>().text = "" + score;

gameStats.GetComponent<DeadMenucontroller>().score = score;

profile.ScoreController(inGameSet.state);


combo++;

ComboText.GetComponent<Text>().text = "x" + combo;

gameStats.GetComponent<DeadMenucontroller>().ComboController(combo);

profile.ComboController(inGameSet.state, combo);
```

x2_PlayerController: OnCollisionEnter2D(Collisio D other)

```
double ost = score % inGameSet.healScore;

if(ost == 0 && hp != inGameSet.maxHp){

   hp++;

}

scrapController.GetComponent<ScrapSpawner>().ScrapSpawn();
```

x2_PlayerController: OnTriggerEnter2D(Collider2D other)

```
if(other.gameObject.tag == "Scrap"){
    body_rotation *= -1;
    hp--;
    combo = 0;
    ComboText.GetComponent<Text>().text = "x" + combo;
    death++; DeathText.GetComponent<Text>().text = "" + death;
    gameStats.GetComponent<DeadMenucontroller>().death = death;
    profile.DeathController(inGameSet.state);
```

x2_PlayerController: OnTriggerEnter2D(Collider2D other)

```
if(hp == 0){
    Instantiate(Explosion, transform.position, Quaternion.identity);
    goal1.SetActive(false);
    goal2.SetActive(false);
    gameObject.SetActive(false);
    gameStats.SetActive(true); gameStats.GetComponent<DeadMenucontroller>
    ().Death();
    gameStats.GetComponent<subMenuController>().pos =
```

Диаграмма классов: общий вид

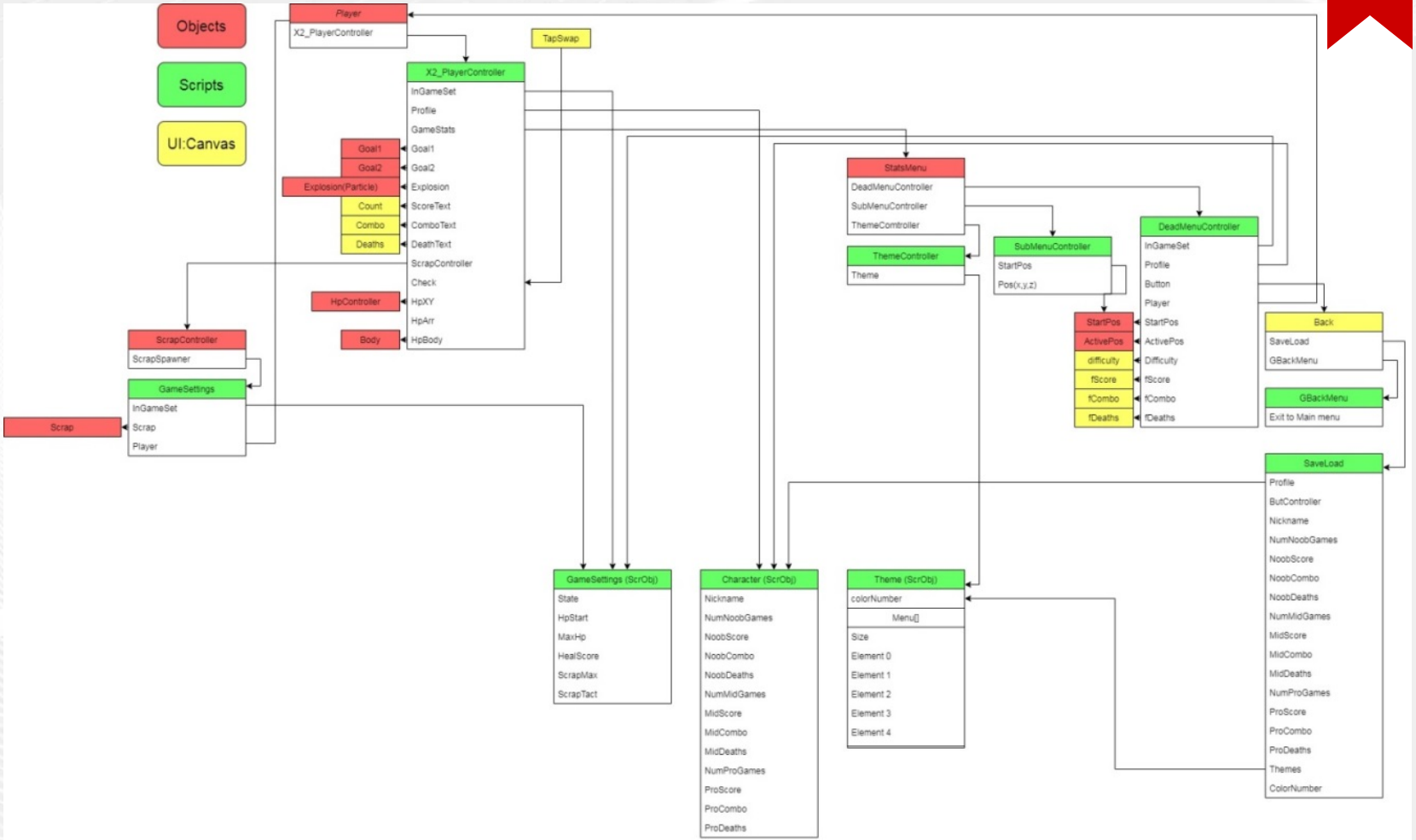


Диаграмма классов: первая часть

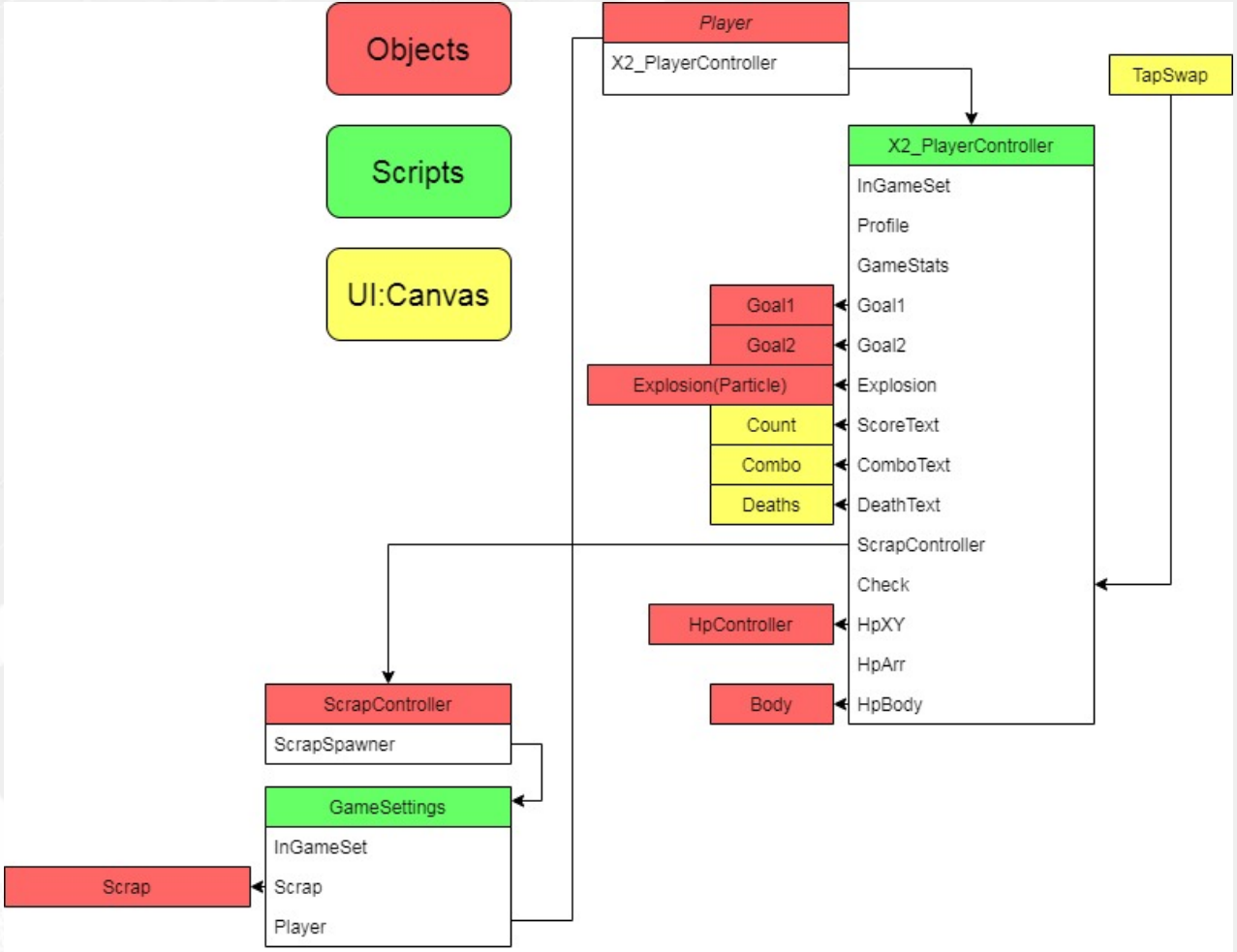
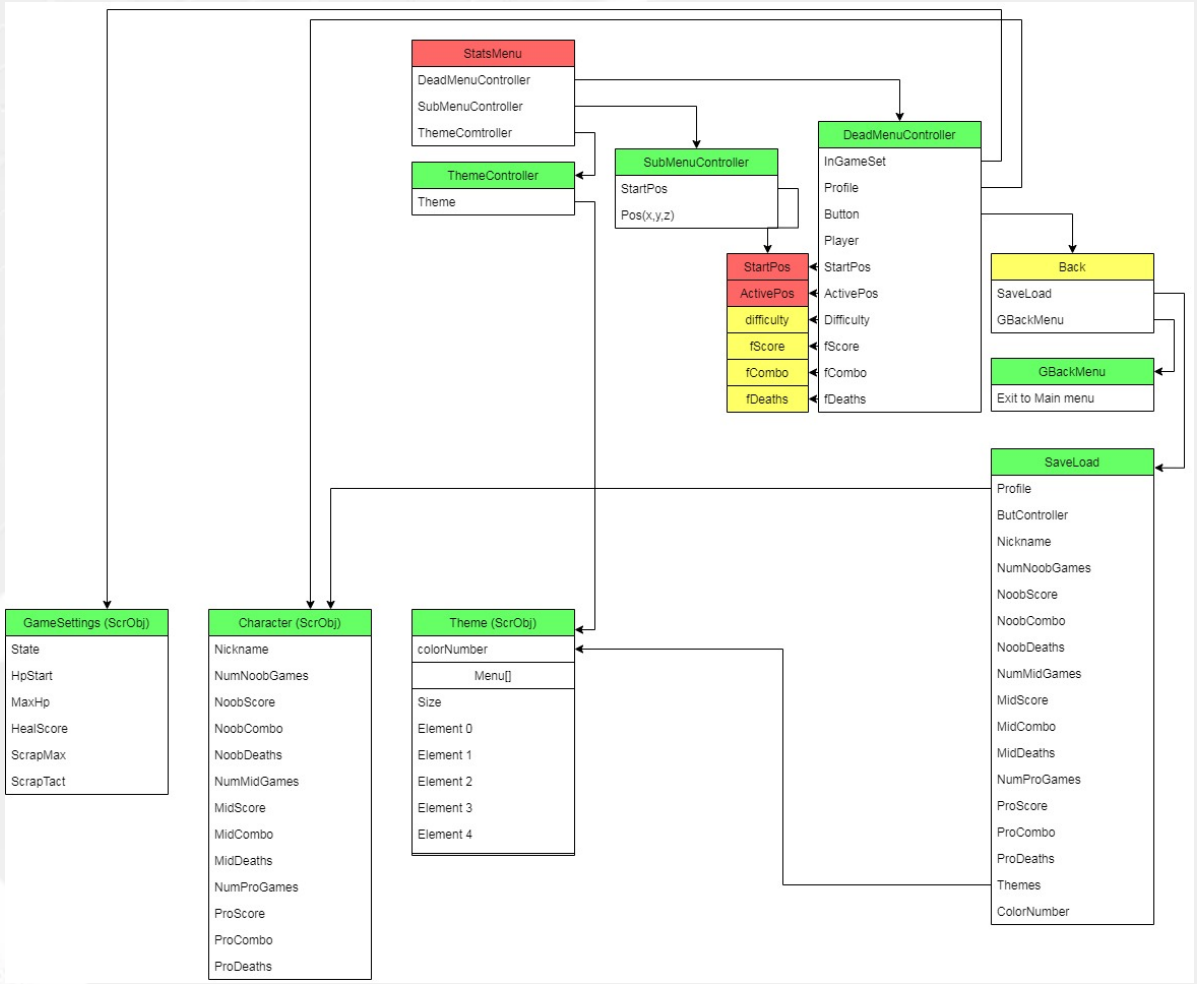


Диаграмма классов: вторая часть



Заключение

Результатом дипломной работы является готовый программный продукт «Техническая демонстрационная версия двухмерной компьютерной игры “HyperCasual”». Реализованы подсистемы графического интерфейса пользователя, управления реестром операционной системы и взаимодействия внутриигровых объектов. Все подсистемы отлажены, оптимизированы, протестированы и интегрированы в единую программную систему.